Non-linear computational geometry for planar algebraic curves

Luis Mariano Peñaranda

Nancy-Université
Université Nancy 2

UFR mathématiques et informatique – École doctorale IAEM Lorraine

Département de formation doctorale en informatique

Géométrie algorithmique non linéaire et courbes algébriques planaires

Thèse

presentée et soutenue publiquement le 3 décembre 2010

pour l'obtention du

Doctorat de l'université Nancy 2

(spécialité informatique)

par

Luis Peñaranda

Rapporteurs: Ioannis Emiris (National Kapodistrian University of Athens)
Stefan Schirra (Otto-von-Guericke University Magdeburg)

Examinateurs: Sylvain Lazard (INRIA Nancy Grand Est – Loria)
Marc Pouget (INRIA Nancy Grand Est – Loria)
Fabrice Rouillier (INRIA Paris – Rocquencourt)
Michael Sagraloff (Max-Planck-Institut für Informatik Saarbrücken)
Salvatore-Antoine Tabbone (Université Nancy 2)
Monique Teillaud (INRIA Sophia Antipolis – Méditerranée)

Laboratoire Lorrain de Recherche en Informatique et ses Appplications – UMR 7503

Loria

Mis en page avec XƎTEX.

Contents

Chapter 1

Introduction

La línea consta de un número infinito de puntos; el plano, de un número infinito de líneas; el volumen, de un número infinito de planos; el hipervolumen, de un número infinito de volúmenes... No, decididamente no es éste, more geométrico, el mejor modo de iniciar mi relato. Afirmar que es verídico es ahora una convención de todo relato fantástico; el mío, sin embargo, es verídico.[1]

Jorge Luis Borges [24]

1.1 Historical review

Geometry, as we know it today, can be defined as a field in mathematics dealing with sizes, shapes, spaces and positions. This idea is the result of an evolution of thousands of years. In the past, different cultures developed geometry independently, even before settling the foundations of mathematics.

[1]Lines consist of an infinite number of points; planes an infinite number of lines; volumes an infinite number of planes, hypervolumes an infinite number of volumes... No, this, this *more geométrico*, is definitely not the best way to begin my tale. Affirming a fantastic tale's truth is now a story-telling convention; mine, though, is true.

In the bronze age, cavemen formed a collection of empirical concepts dealing with angles, lengths and areas. Egyptians and Babylonians knew approximate values of π and formulas to compute some areas and volumes.

In the ancient Greek culture, geometry was one of the most important forms of mathematics. They contributed to the beauty of the geometric problems by introducing the use of logical deduction instead of the trial-and-error methodology, as well as new geometric entities as curves and surfaces. Thales (Θαλῆς), in the sixth century BC, was the first man known to use deduction in mathematics. His disciple Pythagoras (Πυθαγόρας) traveled to Egypt, Arabia, Phoenicia, Judea, Babylon and India with the purpose of gathering all available knowledge, forming a group of students that founded the basic geometry we learn today. Plato (Πλάτων), one of the most important Greek philosophers, wrote in the fourth century BC "Let none ignorant of geometry enter here" in the entrance to his school, showing the importance of geometry to the philosophy given by Greeks. In the summit of Hellenistic geometry, in the third century BC, Euclid of Alexandria (Εὐκλείδης) became the "father of the geometry" by writing his Elements (Στοιχεῖα), book which presented geometry in axiomatic form, establishing what we know today as "Euclidean geometry". Archimedes (Ἀρχιμήδης), finally, developed methods similar to coordinate systems, but he did not have at hand an algebraic language to formulate his ideas three hundred years BC.

Written almost three thousand years ago, the Hindu text śatapatha brāhmaṇa, (शतपथ ब्राह्मण, "Brahmana of one-hundred paths") one of the prose texts describing the Vedic ritual, shows some geometric knowledge spread along mythology. Appeared a few hundreds years more recently, the Śulba Sūtras (शुल्ब सूत्र, "Aphorisms of the Chords") contain geometry related to fire-altar construction. These are presumably the first references to the Pythagorean theorem, some Pythagorean triples, geometric constructions, such as squares and rectangles and approximate area-preserving transformations. The classical period of the Indian geometry came with the Bakshali manuscript, which used a decimal place value system with zero. In this period, the Āryabhatīya astronomical treatise by Āryabhaṭa described operations on cube roots, ratios, plane figures and elemental objects in space, and the Brāhma Sphuṭa Siddhānta by Brahmagupta contains results on cyclic quadrilaterals and rational triangles.

In China, the oldest known reference to geometry is the Mo Jing, written four hundred years before current era, containing advanced geometrical concepts that suggest that this culture had a mathematical knowledge prior to that book. It described objects such as points, lines, circumferences and planes and properties as lengths and parallelism. Two hundred years later, the mathematical treatise Suàn shù shū (算數書, "Book on Numbers and Computation") introduced geometrical solutions to mathematical problems and carefully attempted to give estimations on the value of π. The Jiǔzhāng Suànshù (九章算術, "Nine Chapters on the Mathematical Art") summarizes Chinese knowledge in geometry in the first two hundreds of years of our era. It shows solutions to problems where geometry was applied and trying to find general methods to solve problems, in contrast to the Greek axiomatic deduction. It also lists areas and volume formulas for a wide range of figures.

Historically, starting in the seventh century before our era, Islamic mathematics was more related to algebra and number theory than to geometry. Nevertheless, Muslims developed algebraic geometry. Abū ʿAbdallāh Muḥammad ibn Mūsā al-Khwārizmī (أبو عبد الله محمد بن موسى الخوارزمي), in the eighth century, invented the notion of algorithm, concept which naturally exists independently of computer science even though it is mainly used in that context now days. The connections between algebra, geometry, arithmetics and other fields explored by the followers of al-Khwārizmī completed the elements which today conform the foundations of computational geometry.

Until 16th century, all the knowledge of the different cultures was formalized and unified. The 17th century saw the dawning of the modern geometry: René Descartes and Pierre de Fermat created the analytic geometry, and Girard Desargues introduced the projective geometry. The more recent history of geometry has seen the separation in many almost independent research fields, such as Euclidean and non-Euclidean geometry, algebraic geometry, finite geometry and topology.

1.2 From geometry to computational geometry

Although mechanical calculators where invented in the 17th century, modern computer science was born in the middle of the 20th century, with Alan Turing as principal contributor. The algorithm analysis stated new challenges, and computational geometry arisen in the mid-seventies as the discipline that studies algorithms stated in terms of geometry [136]. This distinction sounds today very rough, since on each of the numerous subfields of computer science, many problems can be stated in terms of geometry. It turns out then that this thesis, as many other research work, cannot be categorized in a single discipline.

The trend in the flourishing computer science consisting in using geometrical language to express problems gained many followers, for it turned out to be a very elegant and convenient way to state and solve problems. Lee and Preparata [113] described the state-of-the-art of computational geometry in the early eighties by distinguishing five classes of problems.

The problem of convex hull, which consists of finding the smallest convex set that contains a given set of points, gave rise to the first class of problems to be considered. This problem was thoroughly studied, giving solutions such as the Graham's scan [87], the Jarvis' march [99] and many others. The many ramifications of this problem, each one having numerous applications, are fundamental bricks in the construction of more complex algorithms.

Another fundamental problem is the intersection detection, consisting in finding all the intersections between a number of given objects. Among a myriad of different variations of this problem and solutions to them, the seminal Bentley-Ottman sweep-line algorithm was presented [16]. This algorithm proved to be easily adaptable to many types and shapes of objects in the plane or in the space, which turned it into a fundamental algorithm in computational geometry.

Geometric searching problems are another pillar of the discipline. A searching problem is simply a query to a certain data base. The importance, from the geometric point of view, is that the geometric data is inherently complex, carrying the need of optimal data structures. Research in this topic lead to the development of many data structures during the seventies [15], but research in the field continues today.

The concept of proximity between objects is related to many practical problems. The Voronoi diagram [161] and the related Delaunay triangulation [54] served in the design of solutions to many problems of this class, from finding the closest pair among a set of points, passing through neighboring problems and minimum spanning trees, to construction of triangulations.

Many problems in operations research, such as linear programming, can also be expressed in terms of geometry. These, along with purely geometric optimization problems, form a vast research topic. Finding the smallest circle enclosing a set of points or finding the largest empty rectangle not containing points from a set belong to this last class.

All the classes of problems described above form the core of computational geometry research. They motivated in the seventies and eighties the creation of methods for solving different variations and generalizations of these problems. These techniques are general enough to attack many problems, but some of them are not sufficiently general and others need some adaptation effort in order to handle complex objects. For instance, the sweep-line algorithm can be used to construct arrangements in many contexts, but it has to be provided with primitives such as intersection of the involved objects. If the objects are complex, like algebraic curves or surfaces, computing intersections is not trivial. In the last years, one of the emerging trends in computational geometry is the development of algorithms that deal with complex objects. This was motivated by the maturity of the basic geometric algorithms, as well as the algorithmic progress in other areas such as computer algebra.

1.3 The burst of computer algebra

The development of computers and computer science created new approaches for solving geometric problems. One of these is computer algebra, area where mathematical tools and computer software are employed to obtain formal solutions to problems [81]. The present thesis addresses the application of computer algebra techniques to computational geometry problems. In the light of the above historical brief review, this thesis is far from being the first attempt to marry geometry with algebra. Much work was done in the past, starting more than two thousand years ago. Modern research in this direction validates the idea that state-of-the-art algebraic techniques extends the application range of many geometric algorithms.

Perhaps the first area related to computational geometry that benefited from computer algebra was motion planning. For instance, in the early eighties, Schwartz and Sharir made one of the first studies of this kind of problems [146]. Canny studied few years later such problems, making contributions both to the field of robotics and the analysis of algorithms [37]. A survey on the advances in this field during those years is given in [110].

One of the more recent and relevant examples of the application of computer algebra to computational geometry was given by Everett et al. [72]. In this paper, they used computer algebra programs to prove a conjecture on the Voronoi diagrams of three lines in general position in three-dimensional space. This fact sketches one of the main motivations of the present work: computer algebra tools permit geometric algorithms to deal with complex objects. The cells of the Voronoi diagram of lines in three dimensions are bounded by quadrics, and it is very hard to deal with the resulting equations of degree two in three variables by using basic geometry.

1.4 Exact geometric computing

Geometric algorithms are often conceived under theoretical assumptions that are known to be unrealistic. The numbers that algorithms handle are often real numbers, and computers on which algorithms are implemented are not capable of representing all real numbers. This gap between theory and practice would not necessarily be a big issue if the implementations were to provide close approximations to the results. Unfortunately, this kind of errors is extremely problematic in computational geometry because it often conducts to incoherences in the decisions performed in the algorithms, leading to totally wrong results, crashes, infinite loops or pretty much any bad scenario that one can think of. Kettner et al. [105] give many such examples in which algorithms fail in different ways.

Algorithms in computational geometry must be aware of these possible inconsistencies. The parts of the algorithm that can lead to failures because of arithmetic errors are usually encapsulated in *predicates*. A predicate is defined, in logic, as an operator which returns either true or false. In computational geometry, a predicate refers to a basic geometric question, based on which the algorithms make decisions. Predicates are stated in terms of geometric objects. For example, a predicate may be: what is the planar orientation of points p, q and r? [105]. They either lie on a common line or form a left or right turn. For instance, the Jarvis' march algorithm [99] relies on the orientation predicate to calculate the convex hull of a set of points. The correctness of the algorithm's output is guaranteed by the correctness of the answers to the orientation predicate.

Answering predicates usually involves evaluating the sign of some expression with real numbers. In simple settings, these expressions are polynomials in the input parameters. For instance, the orientation of three points in the plane can be determined by evaluating the sign of their determinant (in homogeneous coordinates). Furthermore, calculating the determinant using naively computers arithmetic sometimes leads to wrong results. Indeed, when the determinant is zero or close to zero, a small numeric error may induce a wrong sign of the determinant which leads to a wrong orientation. Some arithmetic techniques must be employed in order to guarantee the correctness of geometric results, that is, correctness of the answers to predicates. This paradigm, in which geometric results are always guaranteed to be correct, is referred to as *exact geometric computing paradigm*. A small numeric error in geometric constructions is usually accepted in many applications, but not wrong results to the geometric predicates because, as mentioned before, they lead to major errors. This fact makes this paradigm the standard in geometric computations.

One approach usually taken in order to guarantee geometric results is to perform *exact computation* using *exact arithmetic*. This is possible when the predicates can be answered by evaluating the sign of a polynomial expression. The exact evaluation of polynomial expressions is achieved by representing integer or rational numbers using a memory array, and define arithmetic operations on these numbers. This is called *multiple-precision arithmetic*. A major drawback of such approach is that such exact computations tend to be extremely slow. The usual solution to this issue is to *filter* computations, by using *interval arithmetic*. The idea behind interval arithmetic is to work with intervals, which are guaranteed to contain the exact result (see Section 2.2 for details on interval arithmetic). This way, the sign of an expression can be determined when all the numbers contained in the interval which contains the result of the computation have the same sign. This is not always the case, some results will be known (and their correctness guaranteed) and others will be still unknown. In the latter case, computations are performed again with the help of exact arithmetic. Besides interval arithmetic, there exists other kind of filters, such as static filters.

Static filters assume that the input parameters are smaller than some constant (in absolute value) and an upper bound on the maximum error that can occur when evaluating the expression appearing in the predicate is precomputed (by the scientist). During the processing of the algorithm, as long as the evaluation of the expression is farther away from zero from the computed bound, the sign is ensured (see Sylvain Pion's thesis [133]).

Another approach to guarantee geometric predicates consists in using *multiple-precision interval arithmetic* combined with *separation bounds* [35]. Instead of using machines arithmetic to perform interval computations, intervals bounds are arbitrary-size fixed-precision floating-point numbers. When an interval computation fails to ensure the sign of an expression (*i.e.*, when the interval contains zero), the operation is performed again using finer precision. This process ends when the result of the operation is an interval which either does not contain zero, or whose width is smaller than a so-called separation bound. Such bound is a lower bound on the smallest (in absolute value) non-zero value that the expression can take. One advantage of such bound is that the implementation is rather transparent for the programmer (once appropriate libraries such as CORE [45, 102] are used), and that the expressions in the predicate do not need to be polynomials. One big drawback is that when the expression to be evaluated is actually equal to zero, this approach tends to be very slow.

Dubé and Yap [169] give a comprehensive study of the arithmetic aspects related to the exact computing paradigm. Yap [167] considers specifically geometric issues in the light of the exact computing paradigm. Mehlhorn and Yap [122] also consider algebraic techniques to handle curves and surfaces. Schirra [145] describes geometric robustness issues and shows techniques to overcome them. The book edited more recently by Boissonnat and Teillaud [22] discusses the exact approach to many classical problems in computational geometry.

Determining which approach to use in order to certify results depends on many factors. Nonetheless, the current standard in computational geometry can be said to be the approach combining exact arithmetic and interval arithmetic filtering. It should also be stressed that obtaining efficient implementations in the exact computing paradigm is far from trivial, even in simple cases such as the orientation predicate. The problem turns even more complex when considering curved objects. Since curved objects are defined by polynomials or polynomial systems, they involve algebraic operations such as solving systems of polynomials. Certifying results in this case is thus more challenging. The algorithms in this thesis contain predicates involving curved objects, and the challenges arising in their implementation are discussed.

1.5 Contributions and organization of the thesis

This thesis addresses one of the most basic problems of non-linear computational geometry, that is the determination of the topology of algebraic plane curves. The main originality of this work is a new approach based on computer algebra techniques that permit, in particular, to develop an algorithm whose main novelty is the elimination of the requirement of input curves to be in generic position. The method also avoids in all cases the computation of sub-resultant sequences and computations with algebraic numbers. This algorithm is presented in Chapter 3.

We implemented our algorithm using MAPLE [118]. This implementation was then validated by comparing to other state-of-the-art methods. Implementation and experiments are presented in Chapter 4. The MAPLE programming language is a natural choice to implement the algorithms using algebraic machinery and, in particular, the algorithm we presented in Chapter 3. On the other hand, this computer algebra system is not extensively used in the computational geometry community (and has no reason to be so). The C++ Computational Geometry Algorithms Library, CGAL [38], provides a rich set of geometric algorithms, as well as multiple-precision arithmetics. This library is *the* reference for the development of computational geometry algorithms. This library is thus also a perfect candidate to implement our algorithms. However, algebraic curved objects are defined with polynomials and polynomial systems, and any algorithm handling such objects must be able to work with polynomials. Before implementing algorithms handling curved objects such as the one mentioned above, the library must thus be equipped with some computer algebra machinery.

In the last years, there have been many discussions between European researchers, about the specifications of which algebraic tools were needed in CGAL. In that context, we implemented a so-called CGAL univariate algebraic kernel, a part of the library that contains functions to isolate roots of univariate polynomials, as well as functions to handle and compare polynomial roots. This kernel is related to the development of the RS library [139] by Fabrice Rouillier, and this work was done in close connexion with him. Chapter 5 introduces our CGAL univariate algebraic kernel, and shows experiments that demonstrate the efficiency of our kernel and validate the algebraic kernel approach.

Replacing computer native floating-point arithmetic by multiple-precision arithmetic exposes an issue in the light of the analysis of algorithms. In computational geometry, algorithms are often quite involved and complexity is usually non-trivial to calculate. This is the main reason why algorithms in computational geometry are often analysed in the real-RAM model which assume that computation with reals can be performed in constant time; see [136] for details. We adopt here the bit-complexity model of computation. In it, each operation is implemented using many single precision operations, which turns the cost of performing an operation into a function of the size of the operands. This has of course an impact on the complexity. We recall this notion, introducing the different models of computation and present in Chapter 6 the analysis of the complexity of the algorithm presented in Chapter 3, for the computation of the topology of algebraic planar curves and of the sweep-line algorithm, used in Chapter 5, for computing an arrangement of arcs of algebraic curves defined by univariate polynomials.

Before presenting these contributions, we start in Chapter 2 by recalling some arithmetic and algebraic techniques such as interval arithmetic, root isolation, rational univariate representation and Gröbner bases. Chapter 2 also discusses models of computation, justifying the choice of the bit-complexity model and describes some techniques used in exact geometric computing.

Chapter 2

Preliminaries

> Existe una opinión generalizada según la cual la matemática es la ciencia más difícil cuando en realidad es la más simple de todas. La causa de esta paradoja reside en el hecho de que, precisamente por su simplicidad, los razonamientos matemáticos equivocados quedan a la vista. En una compleja cuestión de política o arte, hay tantos factores en juego y tantos desconocidos e inaparentes, que es muy difícil distinguir lo verdadero de lo falso. El resultado es que cualquier tonto se cree en condiciones de discutir sobre política y arte -y en verdad lo hace- mientras que mira la matemática desde una respetuosa distancia.[2]
>
> Ernesto Sabato [142]

This chapter introduces the basic tools used in the rest in the thesis. It is far from being self-contained, but it provides a brief exposition of the concepts needed to understand the other chapters. Before entering in mathematical details, we begin with an exposition of different models of computation often used in the analysis of geometric

[2]There exists a widespread view which states that mathematics is the most difficult science when it is actually the simplest of all. The cause of this paradox lies in the fact that, precisely for its simplicity, wrong mathematical reasonings are visible. In a complex matter of politics or art there are so many involved factors and so many unknowns and inapparents, that it is very difficult to distinguish the true from the false. The result is that any fool would believe himself in a position to discuss politics and art, and indeed he does, while watching mathematics from a respectful distance.

algorithms. Later, we discuss arithmetic issues relevant to exact geometric computing and we finish by explaining the polynomial system solving strategies used in the thesis.

2.1 Models of computation

When an algorithm is developed, it is targeted to be executed by a certain type of machine. It can be, for instance, read and interpreted by a robot, by a personal computer or even by a human being [46]. The instructions given to each entity differ in the operations that each one can perform. Even when different machines can interpret equal sets of instructions, each one can take a different time to perform each task. A personal computer is able to calculate quite complex arithmetic operations quickly, while a human being may need hours to compute it. Moreover, each machine is able to work with a certain type of data, such as integers, floating-point numbers, or even real numbers. This data is stored in what is called *registers*, and each machine has a certain number of registers.

The set of instructions accepted by the machine in which the algorithm is executed, along with the time spent on each operation is known, in algorithmic, as the *model of computation*. Each algorithm is designed for a specific model of computation. Thus, the complexity of the algorithm is studied in the light of the adopted model of computation.

A widely used model of computation is named *random access machine* (RAM) model. A random access machine consists in a fixed number of registers, each one holding integers of arbitrary size. See [132, § 2.6], for a formal definition of a RAM, along with the set of operations it executes. When analyzing algorithms under the RAM model, it is commonly assumed that each instruction (arithmetic or access to memory) that the RAM executes takes a constant time. This assumption is quite unrealistic in the real world. Today, most of the algorithms are executed in personal computers. They are able to handle integers of a fixed size, commonly 16, 32 or 64-bit. In this case, the assumption that RAM registers can hold any integer is wrong. Arithmetic operations between integers that do not fit in registers can be seen as multiple applications of the RAM arithmetic operations. In this case, each operation is not executed anymore in constant time. The cost of operations turns out to be a function on the size of the operands, since the size of the operands determines the amount of RAM operations needed to complete the operation between big integers.

The idea of cost of operations as a function on the size of the operands is captured by the *logarithmic cost measure* [121, § 1.1]. With this measure, the cost of each operation is a function on the size of the operands. Since, in real computers, numbers are stored in binary form, the size of an operand is its base-two logarithm. This logarithmic cost measure is closer to reality than the unit cost measure, but it fails to represent other aspects. For instance, multiplying integers takes more time than adding them.

In computational geometry, the principal model of computation used is the real-RAM model [136]. Registers of a real-RAM machine are capable of holding real numbers, not only integers. As in a RAM, each operation is performed in a constant amount of time. Storing and operating with real numbers gives the possibility, for instance, of computing coordinates or lengths. While this aspect is very important in computational geometry, this model is farther to reality than the RAM model. The reason is that real numbers cannot, in general, be represented in the registers of real computers. Moreover, the cost of operating with them can be very hard to compute.

In algorithms that handle small size numbers, complexity analyses based on RAM or even real-RAM models give reasonable bounds on the real execution time on a computer. Algorithms presented in this thesis are based on algebraic techniques. Some algorithms in computer algebra present a growth in the size of intermediate results. For instance, calculating the greatest common divisor of polynomials introduces a phenomenon called *intermediate expression swell* [81, § 6.1]. It means that, even calculating the gcd of polynomials of small coefficients giving as a result another polynomial of small coefficients, the computation may introduce huge numbers. The cost of operating with these numbers is far from being constant. Analyzing such an algorithm under the RAM or real-RAM model will correctly estimate the number of arithmetic operations, but the real execution time of the algorithm will be far from the theoretical combinatorial complexity.

The discrepancy between development, theoretical analysis and execution time of an algorithm motivates the introduction of computing models that capture the behavior of the algorithms being executed in a computer or in any real machine. This means, a model capable of representing numbers that a computer can represent, providing instructions that a computer can perform and associating them the time that a computer spends on them. But, since we are willing to work with big numbers that do not fit in computers registers, it will be necessary to define algorithms that perform them. Each number of arbitrary size must be stored in computers memory. Arithmetic operations between big operands in memory must be performed by algorithms that use computers registers and native arithmetic. This approach to machine arithmetic is known as *multiple-precision arithmetic* and it will be explained in the next section. Each multiple-precision arithmetic operation algorithm has a cost, which is a function on the size in memory of its operands. Real numbers cannot be directly represented in this model, but it is possible to represent arbitrary size integer, rational and floating-point numbers. This model of computation is known as *bit-complexity model*. Its name reflects the fact that the cost of operations on this model is a function on the size in bits of the operands, referred to as *bitsize*. Note that the bit-complexity model is close to the notion of logarithmic cost RAM. The cost of operations is, in both models, a function on the bitsize of the operands. Nevertheless, the logarithmic cost RAM defines the cost of all operations using the same function (base-two logarithm of the operands), whereas the bit-complexity model defines the cost of each operation as a different function on each case. For instance, under the bit-complexity model, the cost of an addition is linear in the bitsizes of the operands, while the cost of a multiplication is the bitsize of the operands multiplied by their base-two logarithm. On the other hand, the cost of these two operations (and of all arithmetic operations) under the logarithmic cost RAM model is calculated by applying the same function to the operands.

Real computers have *random-access* memory. This means that accessing any memory cell (a datum that fits in a machine register) is done in constant time. Thus, accessing a number represented in binary in the memory is a linear function on its bitsize, since the number must be copied in registers in order to perform operations on it. Note that this random-access scheme differs from the memory access of Turing machines [157]. In a Turing machine, numbers are represented on a tape which moves one position at a time, thus accessing a stored number is a function on its size, its position on the tape and current tape position.

Algorithms presented in this thesis were analyzed following the bit-complexity model of computation. Using the RAM or real-RAM models would incur in a completely inaccurate analysis. However, it should be stressed that the RAM and real-RAM models provide in many cases an accurate measure of the combinatorial complexity of algorithms. Since combinatorics are the most important part of many algorithms in computational geometry, these computation models are extensively used in the field.

2.2 Computer arithmetic and exact geometric computing

In the last section, we motivated the fact of adopting the bit-complexity model from the point of view of the algorithm analysis. In this section, we explain the need of multiple-precision arithmetic in exact geometric computing. As mentioned in Section 1.4, basic arithmetic operations in a computer are not able to guarantee in all cases the correctness of geometric results. [105] contains examples in geometry where a minor arithmetic error conducts to wrong geometric results.

Before detailing the arithmetic techniques used in exact geometric computing, we will describe the arithmetic that modern computers provide. Arithmetic operations use as operands numbers stored in registers, and computers have a finite number of registers. The result of operations is also stored in a register. Registers have a limited size, normally 32 or 64-bits. Thus, operands and results have a limited size.

Numbers are stored in binary representation. Some different representations can be used to represent numbers in registers. Usually, the representation of a number is called the *type* of the number. The most basic type is the *unsigned integer*, where all the bits of the register are used to represent a number in binary form. Thus, the unsigned integer type can represent integer numbers between zero and $2^b - 1$, where b is the size in bits of computer registers. Negative integers can be also represented as numbers of the type *signed integers*, where the used representation is usually two's complement [108]. Different variants of signed and unsigned integer types are available in computers, depending on the amount of bits used to represent them. For instance, the computer may choose to use half a register to represent a *short* number. Some modern computers can also represent an integer number using two registers. In any case, the size of the numbers is still limited.

Integers are not enough for many applications. Computers also use to represent numbers as *floating-point*. Modern computers follow the ANSI/IEEE 754 [130] standard to implement floating-point types. These types are able to represent numbers in the form $s \times 2^f$, where s is an integer known as *significand* and f is an integer referred to as *exponent*. Due to the limited size of registers, s and f must fit in the same register. The most commonly used floating-point types are the *single-precision* or *float* type, which uses 32-bits to represent both s and f, and the *double-precision* or *double* for short, which uses 64-bits to represent significand and exponent. Floating-point number types also represent infinite values and results of illegal operations such as division by zero. Nevertheless, this kind of arithmetic sometimes fail to provide correct results. Despite the limited size of registers, rationals cannot be represented. For instance, the decimal number 0.1 is not representable, since $1/10$ presents a periodic binary decimal expansion. Operations are neither exact: it may be the case that the exact result of an arithmetic operation does not fit in the space destined to hold the result. The result will be approximated, following certain criteria. This is known as *rounding*. When an operation fails to provide an exact result and returns a rounded approximation, there exists a mechanism to tell the user what happened. In modern computers, users can also control how to round numbers (for instance, they may choose to round numbers towards infinity or towards zero, or just forget the part of the significand that does not fit in the register).

It is common to use double-precision numbers to implement geometric algorithms. One of the popular beliefs is that floating-point types represent any real number within a certain range. If this were true, many implementations of geometric algorithms would be exact. Unfortunately, this is not the case. Let us consider again the orientation predicate from [105] presented in Section 1.4. Implementing this predicate using double-precision floating point arithmetic can lead to incorrect geometric results. The computation of the sign of the determinant will be correct when the result of the determinant is far from zero. In this case, a minor error in floating-point arithmetic is hidden by the fact that the sign of the result is correct. But this is not always the case when the determinant is very close to zero. The first challenge is thus to know when the sign of the determinant is correct. In this case, controlling how the computer rounds the floating-point numbers is the key. It permits to perform operations twice, once rounding the result towards minus infinity and once rounding the result towards plus infinity. This method is called *interval arithmetic*. It is possible that the two results are different, but it is guaranteed that the *exact* result of the operation is contained between the two rounded results. For instance, to add two numbers a and b, contained in intervals $\left[\underline{a}, \overline{a}\right]$ and $\left[\underline{b}, \overline{b}\right]$ respectively, interval arithmetic guarantees that the exact result of $a + b$ is contained in the interval $\left[\underline{\underline{a} + \underline{b}}, \overline{\overline{a} + \overline{b}}\right]$. Here, $\underline{\underline{a} + \underline{b}}$ represents the result of $\underline{a} + \underline{b}$ rounded towards negative infinity, and $\overline{\overline{a} + \overline{b}}$ represents the result of $\overline{a} + \overline{b}$ rounded towards positive infinity. In the same way, the exact result of $a - b$ is contained in the interval $\left[\underline{\underline{a} - \overline{b}}, \overline{\overline{a} - \underline{b}}\right]$. Details on interval arithmetic can be found in [4]. Replacing plain floating-point arithmetic in determinant computation by interval arithmetic gives an interval as result, instead of a floating-point value. If both endpoints of the interval have the same sign, the sign of the determinant is certified, despite the inexact floating-point operations.

Nevertheless, this technique does not suffice to certify the results of all operations. What to do when endpoints of the interval have different signs? The answer is not easy, since this approach reveals that computer's native floating-point arithmetic is not enough to find the correct result but does not solve the issue. What is needed is an *exact* computation of the determinant. But computers native arithmetic proved to be insufficient for this task. In this case, *multiple-precision* arithmetic provides a central tool for exact computing. This approach to computers arithmetic hides the limited size of registers. Each number is stored in computers memory, and its size is not limited to one or two registers. Operations between multiple-precision numbers stored in memory are performed in terms of operations between registers, using native arithmetic. The size of numbers represented in memory is only bounded by the amount of memory the computer has. Multiple-precision arithmetic is usually implemented by libraries on which programs rely, like GMP [83].

Like computers native arithmetic, multiple-precision arithmetic provides different number types, referred to as *multiple-precision number types*. The simplest multiple-precision number type is the *multiple-precision integer* type, also know as *big integer* type. The principal advantage of this number type is that the range of representable integer is virtually unbounded. The direct consequence of this is that operations between big integers never overflow.

The *multiple-precision rational* type, or *big rational* type, is built on top of multiple-precision integer type. A big rational number is represented as two big integer numbers. Operations between rationals are performed as operations on the big integers that form the operands. Rational arithmetic without overflows provides an exact arithmetic as needed in the orientation example. When interval arithmetic fails to provide the sign of the determinant, computing the determinant with exact rational arithmetic provides the exact result. This is permitted by the fact that floating-point numbers are rational numbers. Of course, exact rational arithmetic could be used instead of interval arithmetic, but it is much slower. Switching between different kinds of arithmetic is a very commonly used technique in exact geometric computing. Operations are performed using some kind of fast arithmetic, and most expensive exact arithmetic is used only when necessary.

Multiple-precision techniques are also used to implement floating-point arithmetic. Commonly, a number is represented as a multiple-precision significand a single-precision exponent. Arithmetic operations between this kind of operands can be implemented in two ways. One approach consists in determining the size of the resulting significand when performing the arithmetic operation. The second approach consists in fixing the size of the result of the operation, rounding it when the exact result does not fit in the assigned space. The latter approach, known as *fixed-size* floating-point arithmetic, is widely used in exact geometric computing. In the scenario where interval arithmetic with double-precision floating-point numbers fails to provide a certified result, the operations can be done again using interval arithmetic, using as bounds fixed-precision floating-point numbers of a size bigger than hardware double-precision numbers. The size in memory of the significand of a floating-point number is called its *precision.* This procedure is thus known as *increasing the precision* of computations.

In the example of orientation in Section 1.4, increasing the precision of the determinant computation does not necessarily solve the problem when the input points have rational coordinates, since floating-point numbers cannot exactly represent rational numbers whose denominator is a power of two. Increasing the precision can help to find the sign of some cases, but not in general, unless the precision is increased until a certain threshold does certify the results. This threshold is called *separation bound*, and it refers to a lower bound on the absolute value of the smallest non-zero value that the expression can take [34]. If the width of an interval that contains zero is smaller than a separation bound, it is ensured that the evaluation of that expression is zero.

2.3 Polynomial system solving

As in most geometric algorithms that deal with curved objects, the algorithms presented in this thesis need to find the solutions of systems of polynomials. Solving them is far from trivial, and has been a wide subject of research of in the last fifty years. Resolution strategies for systems of polynomials include numeric, geometric and symbolic methods.

Numeric methods are classified in iterative and homotopy (semi-algebraic) methods [117]. Iterative methods, such as Newton's method, usually work well when they are given good initial guesses, but this is not always possible. Homotopy methods are very demanding when the results need to be certified.

Geometric methods use the particular geometric information of each problem. Examples of these techniques are subdivision methods [163] and ray-tracing [6]. These algorithms are developed in view of specific application domains, thus they are not suitable to be applied in a more general framework.

The most widely used symbolic approaches to polynomial system resolution are resultant methods [152], triangular sets [9] and Gröbner bases [32, 31]. A complete study of symbolic polynomial system solving can be found in [162].

In this thesis, we chose to use the Gröbner approach which, combined with a strategy known as rational univariate representation [140], permits to express the solutions of a zero-dimensional polynomial system (that is, a polynomial system having a finite number of complex roots) as functions on the roots of univariate polynomials. It can be argued that different approaches to polynomial system solving are just different tool sets employed in geometric algorithms. While this is true from a theoretical point of view, we show in Chapter 4 the benefit of our choice when computing with non-generic curves.

Before proceeding to describe the steps of system solving, it is necessary to introduce some geometric and algebraic notions.

Definition 1 ([48] Chapter 1, § 2, Definition 1). *Let $\mathbb{K}$ be a field, and let $f_1, \ldots, f_s$ be polynomials in the polynomial ring $\mathbb{K}[x_1, \ldots, x_n]$. Then we set*

$$\mathrm{V}(f_1, \ldots, f_s) = \left\{(a_1, \ldots, a_n) \in \mathbb{K}^n : f_i(a_1, \ldots, a_n) = 0 \text{ for all } 1 \leq i \leq s\right\}.$$

We call $\mathrm{V}(f_1, \ldots, f_s)$ the affine variety *defined by $f_1, \ldots, f_s$.*

Definition 2 ([48] Chapter 1, § 4, Definition 1). *A subset $\mathrm{I} \subset \mathbb{K}[x_1, \ldots, x_n]$ is an* ideal *if it satisfies:*

(i) $0 \in \mathrm{I}$,

(ii) if $f, g \in \mathrm{I}$, then $f + g \in \mathrm{I}$, and

(iii) if $f \in \mathrm{I}$ and $h \in \mathbb{K}[x_1, \ldots, x_n]$, then $hf \in \mathrm{I}$.

Lemma 3 ([48] Chapter 1, § 4, Definition 2 and Lemma 3). *Let $f_1, \ldots, f_s$ be polynomials in $\mathbb{K}[x_1, \ldots, x_n]$. Then, the ideal generated by $f_1, \ldots, f_s$*

$$\langle f_1, \ldots, f_s \rangle = \left\{ \sum_{i=1}^{s} h_i f_i : h_1, \ldots, h_s \in \mathbb{K}[x_1, \ldots, x_n] \right\}$$

is an ideal of $\mathbb{K}[x_1, \ldots, x_n]$.

Geometrically, a variety is a curve or surface defined by polynomial equations. Ideals are needed to algebraically manipulate these equations. The relation between varieties and ideals becomes then evident. Given a system of equations in the form $\{f_i = 0\}_{i=1,\ldots,s}$ where $f_i \in \mathbb{K}[x_1, \ldots, x_n]$, one can derive other equations using linear algebra. For example, being $h_i \in \mathbb{K}[x_1, \ldots, x_n]$, $h_1 f_1 + \ldots + h_s f_s = 0$ is a consequence of the original system. This illustrates the fact that and ideal of $\langle f_1, \ldots, f_s \rangle$ consists of all the equations one can derive from the system of equations $\{f_i = 0\}_{i=1,\ldots,s}$. The relation between varieties and ideals is a strong connection between geometry and algebra, which must be exploited to solve systems associated to geometrical objects.

2.3.1 Gröbner bases

Gröbner bases were introduced by Buchberger in his 1965 doctoral dissertation [30]. One of their many applications is multivariate polynomial system solving: they permit to reduce the study of polynomial ideals to the study of monomial ideals, allowing to apply in higher dimensions techniques analogous to Gaussian elimination.

Throughout this section, we will note $\mathbb{Z}_{\geq 0}$ the set of non-negative integers, and thus $\mathbb{Z}^n_{\geq 0}$ the n-ary cartesian power of $\mathbb{Z}_{\geq 0}$. For $x = (x_1, \ldots, x_n) \in \mathbb{K}[x_1, \ldots, x_n]$ and $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{Z}^n_{\geq 0}$, we will refer to $(x_1^{\alpha_1}, \ldots, x_n^{\alpha_n})$ as the *monomial* x^α.

Definition 4 ([48] Chapter 2, § 2, Definition 1). *A* monomial ordering *on $\mathbb{K}[x_1, \ldots, x_n]$ is any relation $>$ on $\mathbb{Z}^n_{\geq 0}$, or equivalently, any relation on the set of monomials x^α, $\alpha \in \mathbb{Z}^n_{\geq 0}$, satisfying:*

(i) $>$ is a total (or linear) ordering on $\mathbb{Z}^n_{\geq 0}$,

(ii) *if* $\alpha > \beta$ *and* $\gamma \in \mathbb{Z}^n_{\geq 0}$, *then* $\alpha + \gamma > \beta + \gamma$, *and*

(iii) $>$ *is a well ordering on* $\mathbb{Z}^n_{\geq 0}$, *meaning that every nonempty subset of* $\mathbb{Z}^n_{\geq 0}$ *has a smallest element under* $>$.

Important and commonly used monomial orderings are the lexicographic order, graded lexicographic order and graded reverse lexicographic order. What follows is the formalization of some common ideas.

Definition 5 ([48] Chapter 2, § 2, Definition 7). *Let* $f = \sum_\alpha a_\alpha x^\alpha$ *be a nonzero polynomial in* $\mathbb{K}[x_1, \ldots, x_n]$ *and let* $>$ *be a monomial ordering.*

(i) *The* multidegree *of* f *is* $\text{multideg}(f) = \max(\alpha \in \mathbb{Z}^n_{\geq 0} : a_\alpha \neq 0)$, *the maximum being taken with respect to* $>$,

(ii) *the* leading coefficient *of* f *is* $\text{LC}(f) = a_{\text{multideg}(f)} \in \mathbb{K}$,

(iii) *the* leading monomial *of* f *is* $\text{LM}(f) = x^{\text{multideg}(f)}$ *with coefficient 1, and*

(iv) *the* leading term *of* f *is* $\text{LT}(f) = \text{LC}(f) \times \text{LM}(f)$.

Definition 6 ([48] Chapter 2, § 5, Definition 1). *Let* $\text{I} \in \mathbb{K}[x_1, \ldots, x_n]$ *be an ideal other than* $\{0\}$.

(i) $\text{LT}(\text{I})$ *denotes the set of leading terms of* I, *thus*

$$\text{LT}(\text{I}) = \left\{cx^\alpha : \textit{there exists} f \in \text{I} \textit{ with } \text{LT}(f) = cx^\alpha\right\}, \textit{ and}$$

(ii) $\langle\text{LT}(\text{I})\rangle$ *denotes the ideal generated by the elements of* $\text{LT}(\text{I})$.

The previous definitions permit to formally define what a Gröbner basis is.

Definition 7 ([48] Chapter 2, § 5, Definition 5). *Given a monomial ordering, a finite subset* $G = \{g_1, \ldots, g_n\}$ *of an ideal* I *is said to be a* Gröbner basis *if*

$$\langle\text{LT}(g_1), \ldots, \text{LT}(g_n)\rangle = \langle\text{LT}(\text{I})\rangle.$$

2.3.2 Rational univariate representation

In this thesis, Gröbner bases are used in the light of the rational univariate representation, *RUR* for short [140]. This modern solving technique permits to express the zeroes of a zero-dimensional system of multivariate polynomials as a function of rational functions of univariate polynomials. Briefly, the RUR of the zero-dimensional ideal $\mathrm{I} \in \mathbb{K}[x_1, \ldots, x_n]$ is a set set of (n+2) polynomials $\{f, g, g_0, \ldots, g_n\} \subseteq \mathbb{K}[\mathrm{T}]$ with the property that the roots of the system are exactly $\left\{ \left(\frac{g_1(t)}{g(t)}, \ldots, \frac{g_n(t)}{g(t)} \right) \middle| f(t) = 0 \right\}$. A more formal definition is given in [140, § 3].

All these univariate polynomials, and thus the RUR, are uniquely defined with respect to a given polynomial $\gamma \in \mathbb{K}[x_1, \ldots, x_n]$ which is injective on $V(\mathrm{I})$; γ is the *separating polynomial* of the RUR. In general, separating polynomials are defined as follows.

Definition 8 ([140] § 2). *A polynomial $t \in \mathbb{K}[x_1, \ldots, x_n]$ separates a variety* V*, if*

$$\forall \alpha, \beta \in \mathrm{V}, \alpha \neq \beta \Rightarrow t(\alpha) \neq t(\beta).$$

A random degree-one polynomial in $x_1, \ldots, x_n$ is a separating polynomial of the RUR with probability one. For details, including how to calculate a separating polynomial (also called *separating element*), see [140].

The RUR defines a bijection between the (complex and real) roots of the ideal I and those of f. Furthermore, this bijection preserves the multiplicities and the real roots. Computing a box for a solution of the system is done by isolating the corresponding root of the univariate polynomial f (see Section 2.3.3) and evaluating the coordinate functions with interval arithmetic. To refine a box, one just needs to refine the corresponding root of f and evaluate the coordinates again.

There exists several ways of computing a RUR. The strategy adopted throughout thesis is the one from [140, § 5], consisting in computing a Gröbner basis of I and then to perform linear algebra operations in order to compute a full expression of the RUR. This method is very efficient in practice, specially when working with polynomials with integer coefficients. Other strategies consist, for example, in replacing the Gröbner basis computation by the generalized normal form from [127], or more or less certified alternatives such as the geometrical resolution [82] (this method is probabilistic, since the separating element is randomly chosen and its validity is not checked, one also loses the multiplicities of the roots) or resultant-based strategies such as in [107].

2.3.3 Univariate polynomial root isolation

Gröbner bases and rational univariate representations permit, once they have been computed, to reduce the problem of solving a multivariate polynomial system to the problem of finding roots of univariate polynomials. This section discusses the methods that solve this problem.

The problem of solving univariate polynomial equations started to be studied by ancient cultures more than four thousand years ago. [131] contains a brief historical and technical review on polynomial solving. During history, research on this topic motivated the development of fundamental concepts of mathematics, such as irrational and complex numbers. Many attempts were done, starting in the sixteenth century, to find solutions to general polynomial equations of low degree. However, all attempts failed to provide a general solution of the equation of degree bigger than four.

In 1824, Abel showed that, in general, the quintic equation cannot be solved in terms of radicals [2]. This means that, in general, the algebraic numbers (roots of polynomials) of degree at least five can only be expressed in terms of polynomials of which they are roots. The main implication of this result is that operating with algebraic numbers implies operating with the polynomials that define them. For instance, to add two algebraic numbers α and β, roots of polynomials p and q respectively, one needs to calculate the polynomials of which $\alpha + \beta$ is root, that is, the resultant with respect to y of $p(x - y)$ and $q(y)$ [116]. Even though, arithmetic operations can be done in terms of polynomials. But comparison of algebraic numbers, a central problem in exact geometric computing, cannot be done easily without approximating them. Operating only with the polynomials defining algebraic numbers is in general not sufficient to compare algebraic numbers. Only in cases where approximation do not suffice, it is necessary to fall back to polynomial operations.

Algebraic numbers can be approximated in various ways. In exact computing, approximation is accomplished by *isolation*. This technique consists, in enclosing every root of a polynomial inside a real interval, in order to certify computations. Each interval contains one and only one root of the polynomial. Once isolated, each algebraic number can be approximated as much as desired, using one of the many *refinement* techniques. Today, the algorithms used for real root isolation can be classified in two groups: those based on the Sturm sequences and those based on the Descartes' rule of signs.

Descartes formulated, in the seventeenth century, a fundamental result in polynomial solving [55]. He established an upper bound on the number of positive roots of a polynomial. He observed that the difference between the number of sign changes in the coefficients and the number of positive real roots of a polynomial is even and nonnegative. This rule is formally stated as follows.

Definition 9. *The sign of an element* $a \in \mathbb{R}$ *is defined as* $\mathrm{sg}(a) = 0$ *if* $a = 0$ *and* $\mathrm{sg}(a) = \dfrac{a}{|a|}$ *otherwise.*

Definition 10. *The number of sign changes* $V(L)$ *in the list* $L = (l_0, \ldots, l_n)$ *of elements of* $\mathbb{R} \setminus \{0\}$ *is defined by induction as*

$$V(l_0) = 0$$

$$V(l_0, \ldots, l_k) = \begin{cases} V(l_0, \ldots, l_{k-1}) + 1 & \text{if } \mathrm{sg}(l_{k-1} l_k) = -1 \\ V(l_0, \ldots, l_{k-1}) & \text{otherwise.} \end{cases}$$

The number of sign changes of a list M *of elements of* $\mathbb{R}$*, is defined as* $V(M) = V(L)$*, where* L *is the list obtained by removing all the zeroes from* M*.*

Theorem 11 (Descartes' rule of signs). *Let a polynomial* $p(x) = \sum_{i=0}^{n} a_i x^i \in \mathbb{R}[x]$, $V(p)$ *the number of sign changes in the list* $(a_0, \ldots, a_i)$ *and* $pos(p)$ *the number of positive roots of* p*, counted with multiplicities. Then,* $pos(p) \leq V(p)$ *and* $V(p) - pos(p)$ *is even.*

Though Descartes' rule of signs only gives the exact number of roots when the difference is zero or one, it is of vital importance for root isolation.

Fourier's theorem [77, 78] introduced the Fourier sequence and, in 1829, Sturm replaced them with the Sturm series and used them to conceive the first root isolation method [123]. This algorithm computes a signed polynomial remainder sequence, evaluating it over the endpoints of the intervals. Based on Budan's theorem [33], Vincent presented in 1836 the continued fractions method [160]. This method computes the continued fraction expansion for each real root of the polynomial. Uspensky [158] extended Vincent's method in 1948. Based on Descartes' rule and in Vincent's theorem, Collins and Akritas [43] presented a faster version of Uspenky's algorithm. Many variations of this method were presented, but Rouillier and Zimmermann proved in 2004 that all of these were in fact instances of a generic algorithm [141]. They described the existing methods using a unified framework, and presented an algorithm which is optimal in memory usage, and does not perform more computations than other algorithms based on Descartes' rule of signs. Due to its memory optimality and its practical performance, this algorithm is the one used in this thesis to perform isolation of univariate polynomial roots (Sturm sequences have a lower complexity bound in the worst case, but in practice Descartes' algorithm tends to perform better).

A generic Descartes' algorithm can be seen as a recursive subdivision in intervals of the real axis. Each interval is checked for the presence of roots. If there are no roots in the interval, it is thrown away. If there is exactly one root, the interval is output since it is one isolating interval for a root. If there is an uncertain number of roots in the interval, this procedure of subdivision is applied recursively. This process ends when there are no more intervals in the real axis that contain an uncertain number of roots. The output of the algorithm is a list of isolating intervals, each one containing exactly one root of the input polynomial.

Note that Descartes' rule of signs in Theorem 11 counts the roots with multiplicities. In the case of an interval containing a multiple root, the rule of signs will fail to assert that there is only one (multiple) root. Then, before applying a Descartes-based algorithm, it will be necessary to compute the *square-free* part of the polynomial, that is, to eliminate the multiple roots of the polynomial. The square-free part of a polynomial P is computed as $\mathrm{sfp}(P) = \frac{P}{\gcd(P,P')}$. Thus we do not lose generality by requiring that the input polynomial is square-free.

To check the presence of roots of a square-free polynomial f in a given interval using the Descartes' rule of signs, f is transformed into another polynomial g having as many roots in the positive real axis as f in the given interval. Thus to find *all* the roots of f, it would be first necessary to find an interval containing all the roots of the polynomial, that is, lower and upper bounds to polynomial roots. There are many methods to compute such bounds on the values of roots. For instance, being m the maximum absolute value of polynomial's coefficients, the interval $[-m, m]$ contains all the roots of the polynomial. Finer bounds also exist, see [138]. Once found an interval bounding all the roots, the polynomial transformations that are applied are the following.

Definition 12. *Let $P \in \mathbb{R}(x)$, $\deg(P)$ the degree of P and $c \in \mathbb{R}$. The transformations R, H_c and T_c are defined as follows:*

$$
\begin{aligned}
R(P(x)) &= x^{\deg(P)} P\left(\frac{1}{x}\right) \\
H_c(P(x)) &= P(cx) \\
T_c(P(x)) &= P(x+c).
\end{aligned}
$$

The transformation $H_{b-a}T_a$ maps the interval $]0,1[$ to the interval $]a,b[$, that is, the polynomial $g(x) = H_{b-a}T_a(f(x))$ has as many roots in $]0,1[$ as f has in $]a,b[$. Analogously, $T_1 R H_{b-a} T_a$ maps the positive real axis to the interval $]0,1[$. The objective of algorithms based on Descartes' rule of signs is to divide the real axis in intervals, such that each interval has exactly zero or one root. The interval division of the real axis spans a tree, where each node is an interval and its siblings are its subintervals. Different methods construct this tree using different strategies. The reader is referred to [141] for details on the algorithms.

Our final consideration about univariate polynomial solving is related to the recovering of roots of the original polynomial system. The output of the root isolation algorithms is a list of isolating intervals. According to the strategy adopted in this thesis to solve polynomial systems, the roots are plugged in the RUR, in order to obtain the solutions of the original system. Plugging intervals in the RUR means that the evaluation of the RUR is done with interval arithmetic. Thus, each component of the solution of the original system will thus be, in general, a box containing the solution. It may happen that two boxes containing solutions of the same system overlap. Each one is certified to contain a different solution, but they are usually expanded by interval arithmetic. Thus they need to be refined, by refining the univariate isolating intervals which were plugged in the equations of the RUR, and evaluating the RUR again.

Chapter 3

Computation of the Topology of Planar Algebraic Curves

> A single curve, drawn in the manner of the curve of prices of cotton, describes all that the ear can possibly hear as the result of the most complicated musical performance.... That to my mind is a wonderful proof of the potency of mathematics.
>
> Sir William Thompson (Lord Kelvin) [154]

This chapter presents an algorithm aimed to determine the topology of algebraic curves in the plane. Moreover, it gives geometrical information on critical points, crucial in applications such as arrangement computation. A challenge is to compute efficiently this information for the given coordinate system even if the curve is not in generic position.

Previous methods based on the cylindrical algebraic decomposition [3, 7, 8, 28, 56, 61, 86] use sub-resultant sequences and computations with polynomials with algebraic coefficients. A novelty of the proposed approach is to replace these tools by Gröbner basis computations and isolation with rational univariate representations. This has the advantage of avoiding computations with polynomials with algebraic coefficients, even in non-generic positions. The algorithm isolates critical points in boxes and computes a decomposition of the plane by rectangular boxes. This decomposition also induces a new approach for computing an arrangement of polylines isotopic to the input curve.

The algorithm introduced in this chapter is the result of a long research project that started in 2007 and involved several people, including two postdoctoral fellows in the early stages of the project. These results were presented in 2009, during the annual *Symposium on Computational Geometry* [42], and has been accepted for publication in the journal *Mathematics for Computer Science* [40].

We also discuss in this chapter (Section 3.4) the problem of the embedding of the vertices of the output graph on a grid. Using a result from Milenkovic and Nackman, we show that this problem is in NP.

3.1 Introduction

Let $\mathscr{C}$ be a real algebraic plane curve defined in a Cartesian coordinate system by a bivariate polynomial f with rational coefficients, *i.e.*, $\mathscr{C} = \{(x, y) \in \mathbb{R}^2 : f(x, y) = 0\}$ with $f \in \mathbb{Q}[x, y]$. We consider the problem of computing the topology of $\mathscr{C}$ with additional geometric information associated to the given coordinate system. By computing the topology of $\mathscr{C}$, we mean to compute an arrangement of polylines $\mathscr{G}$, that is topologically equivalent to $\mathscr{C}$ (see Figure 3.1). Note that this arrangement of polylines $\mathscr{G}$ is often identified to a graph embedded in the plane, where the vertices can be placed at infinity and the edges are straight line segments. We first define formally what we mean by topologically equivalent, before discussing the additional geometric information we consider and their relevance.

Two curves $\mathscr{C}$ and $\mathscr{G}$ of the Euclidean plane are said to be (ambient) isotopic if there exists a continuous map $F : \mathbb{R}^2 \times [0, 1] \longrightarrow \mathbb{R}^2$, such that $F_t = F(\cdot, t)$ is a homeomorphism for any $t \in [0, 1]$, F_0 is the identity of $\mathbb{R}^2$ and $F_1(\mathscr{C}) = \mathscr{G}$. This notion formalizes the idea that one can deform one curve to the other by a deformation of the whole plane. Isotopy is stronger than homeomorphy, for instance, (i) two nested

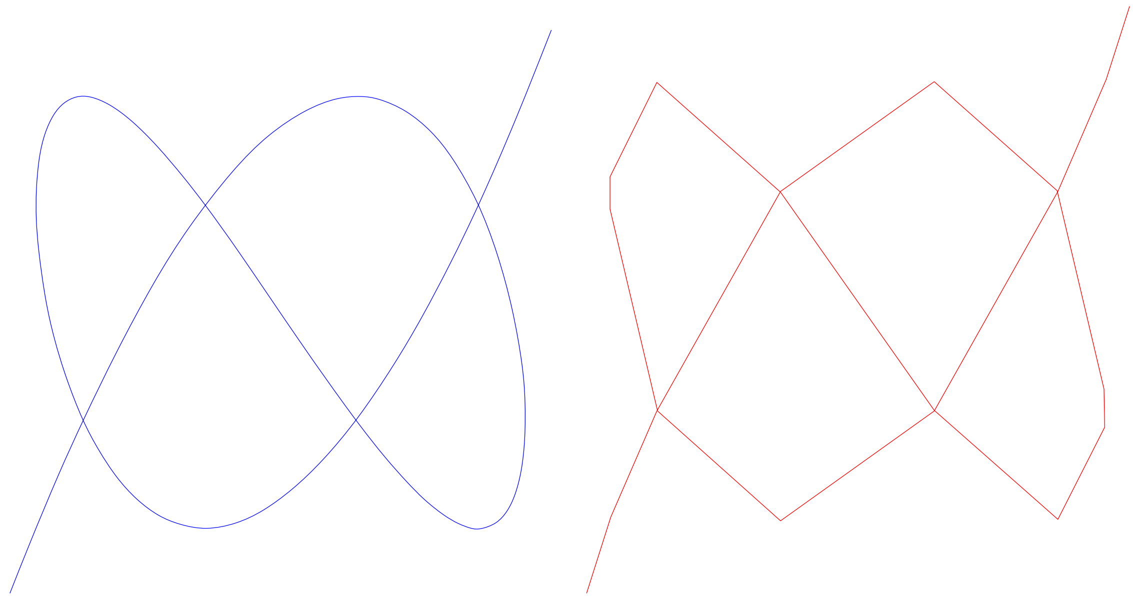


Figure 3.1: Curve of equation $16\,x^5 - 20\,x^3 + 5\,x - 4\,y^3 + 3\,y = 0$ plotted in MAPLE and its isotopic graph computed with our algorithm.

loops and (ii) two non-nested loops are not isotopic.[3]

We now discuss the relevance of adding geometric information to the graph $\mathscr{G}$. From the topological point of view, the graph $\mathscr{G}$ must contain vertices that correspond to self-intersections and isolated points of the curve. However, in order to avoid separating such relevant points from other singularities (*e.g.*, cusps), all *singular points* of $\mathscr{C}$, that is, points at which the tangent is not well defined, are chosen to be vertices of the graph.

While singular points are needed for computing the topology of a curve, the *extreme points* of a curve are also very important for representing its geometry. Precisely, the *extreme points* of $\mathscr{C}$ for a particular direction, say the direction of the x-axis, are the non-singular points of $\mathscr{C}$ at which the tangent line is vertical (*i.e.*, parallel to the y-axis); the extreme points in the direction of the x-axis are called *x-extreme*. These extreme points are crucial for various applications and, in particular, for computing arrangements of curves by a standard sweep-line approach [60]. Of course, one can theoretically compute an arrangement of algebraic curves by computing the topology of their

[3]Note that in two dimensions, the notion of ambient isotopy is equivalent to the notion of ambient homeomorphy, that is, to the existence of an orientation preserving homeomorphism of $\mathbb{R}^2$ that maps $\mathscr{C}$ onto $\mathscr{G}$ [21, Thm. 4.4 p.161].

product. However, this approach is obviously highly inefficient compared to computing the topology of each input curve and, only then, computing the arrangement of all the curves with a sweep-line algorithm. Note that the x-extreme and singular points of $\mathscr{C}$ form together the *x-critical points* of the curve (the x-coordinates of these points are exactly the positions of a vertical sweep line at which there may be a change in the number of intersection points with $\mathscr{C}$).

It is thus useful to require that all the x-critical points of $\mathscr{C}$ are vertices of the graph we want to compute. To our knowledge, almost all methods for computing the topology of a curve compute the *critical points* of the curve and associate corresponding vertices in the graph. (Refer to [3, 36] for recent subdivision methods that avoid the computation of non-singular critical points.) However, it should be stressed that almost all methods do not necessarily compute the critical points for the specified x-direction. Indeed, when the curve is not in *generic position*, that is, if two x-critical points have the same x-coordinate or if the curve admits a vertical asymptote, most algorithms shear the curve so that the resulting curve is in generic position. This is, however, an issue for several reasons. First, determining whether a curve is in generic position is not a trivial task and it is time consuming [84, 144]. Second, if one wants to compute arrangements of algebraic curves with a sweep-line approach, the extreme points of all the curves have to be computed for the same direction. Finally, if the coordinate system is sheared, the polynomial of the initial curve is transformed into a dense polynomial, which slows down, in practice, the computation of the critical points.

In this paper, given a curve $\mathscr{C}$ which is not necessarily in generic position, we aim at computing efficiently its topology, including all the critical points for the specified x-direction. In other words, we want to compute an arrangement of polylines that is isotopic to $\mathscr{C}$ and whose vertices include the x-critical points of $\mathscr{C}$. In terms of efficiency, our primary goal is the practical efficiency rather than worst-case complexity. In particular, we want to avoid computations with non-rational algebraic numbers or, equivalently in this context, algebraic computations such as Sturm sequences, Sturm-Habicht sequences (which are a generalization of Sturm sequences, with better specialization properties [85]), and sub-resultant sequences.

After a brief overview of our algorithm, we review below previous work on the problem and then present our contributions.

3.1.1 Previous Work

There have been many papers addressing the problem of computing the topology of algebraic plane curves (or closely related problems) defined by a bivariate polynomial with rational coefficients [3, 8, 47, 13, 61, 75, 84, 86, 97, 126, 143, 28, 7, 120, 151, 56]. Most of the algorithms assume generic position for the input curve. As mentioned above, this is without loss of generality since we can always shear a curve into generic position [144, 84] but this has a substantial negative impact on the time computation. All these algorithms perform the following phases, depicted in Figure 3.1.1.

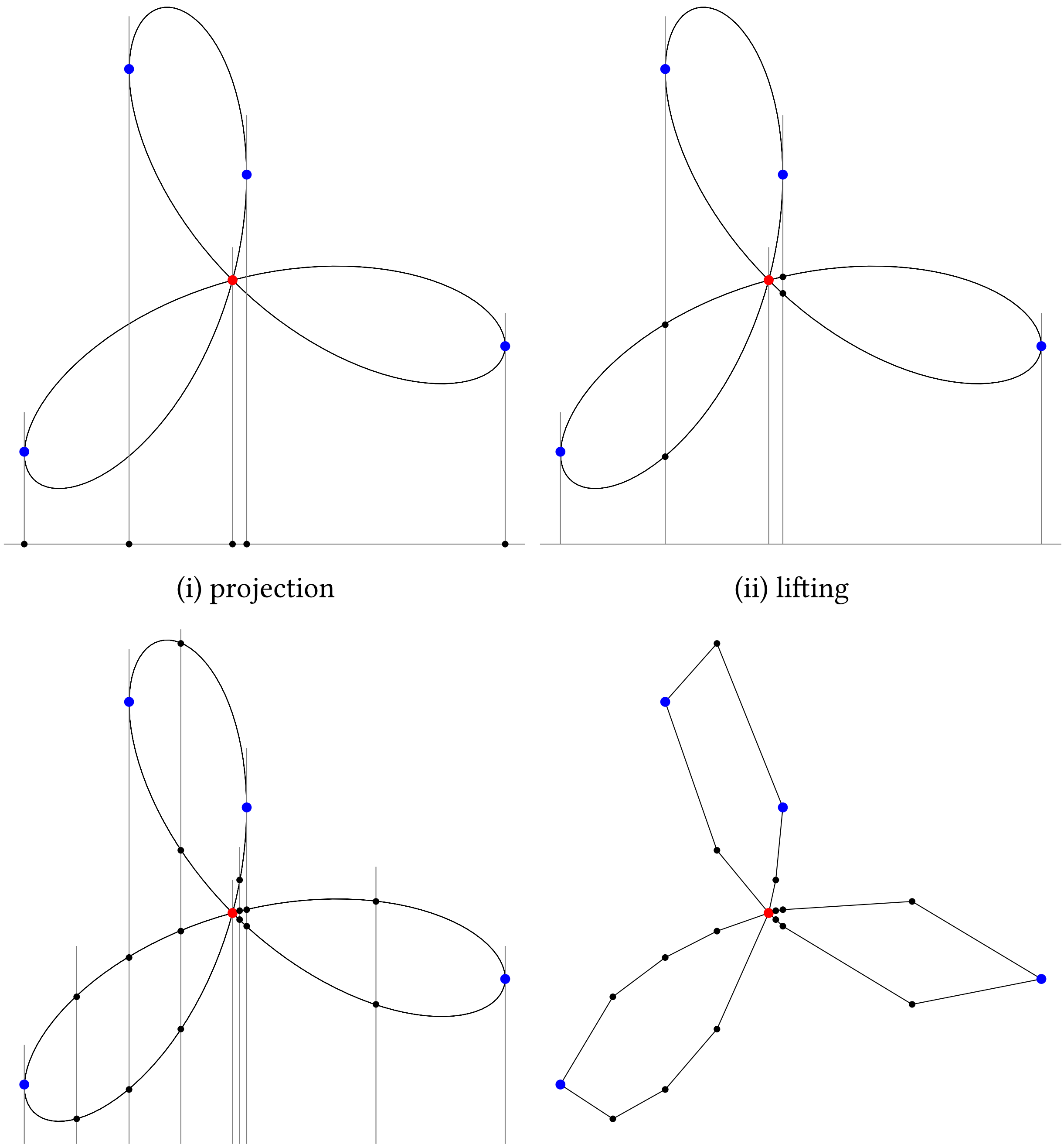
(i) Project the x-critical points of the curve on the x-axis using sub-resultants sequences, and isolate the real roots of the resulting univariate polynomial in x. This gives the x-coordinates of all the x-critical points.

(ii) For each such value x_i, compute the intersection points between the curve $\mathscr{C}$ and the vertical line $x = x_i$.

(iii) Through each of these points, determine the number of branches of $\mathscr{C}$ coming from the left and going to the right.

(iv) Connect all these points appropriately.

The main difficulty in all these algorithms is to compute efficiently all the critical points in phase ii because the x-critical values in phase i are, a priori, non-rational. Thus computing the corresponding y-coordinates in phase ii amounts, in general, to solving a univariate polynomial with non-rational coefficients and at least a multiple root (corresponding to the critical point). To this end, most algorithms [8, 47, 13, 75, 84, 86, 97, 126, 143, 7, 151] rely heavily on computations with real algebraic numbers, Sturm sequences or sub-resultant sequences. Most implementations of these algorithms, despite



(i) projection

(ii) lifting

(iii) intermediate fibers

(iv) adjacencies

Figure 3.2: The four usual phases of Cylindrical Algebraic Decomposition methods.

good theoretical bounds, have an arguable performance. However, handling cleverly algebraic expressions may yield algorithms with very good performances, as in the case of CA, the algorithm presented in [61] (see also [60, 64] for related results).

An approach using a variant of sub-resultant sequences for computing the critical points in phase ii was proposed by Hong [97]. He computed this way (xy-parallel) boxes with rational endpoints and separating the critical points. Counting the branches in phase iii can then be done by intersecting the boundary of the boxes with the curve which only involves univariate polynomials with rational coefficients. This approach, see also [7, 120, 28] and the software package CAD2D [27, 96], does not assume that the curve is in generic position.

In a more recent paper, González-Vega and Necula [86] use Sturm-Habicht sequences which allow them to express the y-coordinate of each critical point as a rational function of its x-coordinate. They implemented their algorithm in MAPLE with symbolic methods modulo the fact that the algebraic coefficients of the polynomials in phase ii are approximated in fixed-precision arithmetic. The algorithm takes as a parameter the initial precision for the floating-point arithmetic and recursively increases the precision. This approach is however not certified in the case where the curve is not in generic position because the algorithm checks for the equality of pairs of polynomials whose coefficients are evaluated. This initial precision can be selected by the user, but no guarantee is given that it is leads to the correct result. Hence their implementation, TOP, is not completely certified, in particular in non-generic positions, and incorrect results have been reported in [147] when critical points are crowded and the floating-point precision is not large enough. Note that there exists one variant of González-Vega and Necula algorithm that handles, without shearing, curves that are not in generic position [126]. This approach, however, requires substantial additional time-consuming symbolic computations such as computing Sturm sequences.

More recently, Seidel and Wolpert [147] presented an alternate approach for computing the critical points avoiding most costly algebraic computations but to the expense of computing several projections of the critical points. They project, in phase i, the x-critical points on both x and y-axes and also on a third random axis. After isolating the roots on each axis, they can recover (xy-parallel) boxes with rational endpoints that contain each exactly one critical point. From there, all computations only involve rational numbers but they, however, still need to compute Sturm-Habicht sequences for refining the boxes containing the singular points until each box interests only the branches of the curve incident to the singular point. Their approach assumes that the curve is in generic position by a pre-processing phase in which the curve is sheared if needed. They also present a MAPLE implementation, INSULATE, which is an implementation of a certified algorithm for curves in arbitrary positions. Note that their implementation does not report x-extreme points in the original system when the curved is sheared.

Even more recently, Eigenwillig et al.[61] (see also [103]) presented a variant of González-Vega and Necula approach, in which the roots of the polynomials with non-rational coefficients are efficiently isolated using an implementation of a variant of an interval-based Descartes algorithm [62]. This variant, as [86], does not assume that the curve is in generic position but detects such configurations online. More precisely, if the bit-stream Descartes algorithm is, in a sense, unlucky then, rather than refining down to a separation bound (*e.g.* [35, 114]), the algorithm shears the input curve and starts again. Note that this approach still computes Sturm-Habicht sequences for determining the polynomials appearing in phase ii and the multiplicity of its multiple root. Also, if the curve is sheared to a (x', y') coordinate system, they compute extreme points both for the x'-direction and the direction corresponding to the x-axis. This approach has been implemented in C++ and is an implementation of a certified algorithm that handles curves that are not necessarily in generic positions and that reports x-extreme points for the original coordinate system.

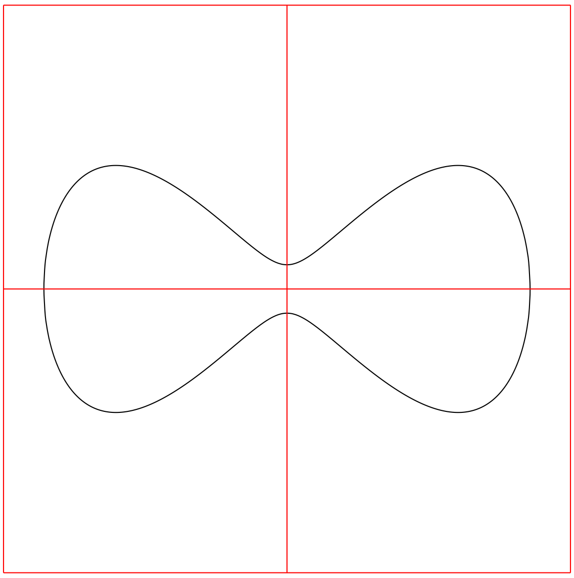
Note finally that another approach that avoids expensive algebraic computations is to compute the critical or singular points using subdivision methods. These methods consist of subdividing the plane in sections, and iteratively subdivide the interesting sections. Different methods have different criteria to choose the regions to be subdivided, and how to subdivide them. Figure 3.3 shows a typical subdivision of the plane, considering in this case a nonsingular curve used as example in [36]. The major drawback of these methods is that, in order to certify the results, the subdivision has, in general, to reach a separation bound (certification can also be achieved by solving algebraic systems by other means). It follows that, if no certification is required, these methods are very fast in practice, however, they can become very slow on difficult instances when certification is required. To our knowledge, no implementation of such a certified algorithm is available. Subdivision methods are the plane equivalent to the *marching cubes* methods, introduced by Lorensen and Cline in 1987. Later, Snyder [149] and Plantinga and Vegter [135] formulated some subdivision algorithms. See [3] and [36] for detailed description of some modern subdivision methods.

3.1.2 Novelty of the algorithm

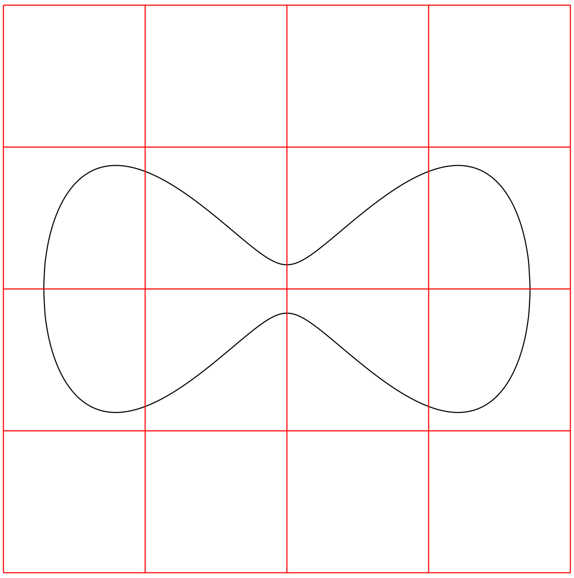
We present an algorithm for computing the topology of an algebraic plane curve which is possibly in non-generic position. The algorithm handles curves in non-generic positions in the Cartesian coordinate system in which they are defined. In particular, the algorithm never shears the coordinate system, which avoids the associated extra costs discussed above.

Another specificity of our approach is that we succeed to avoid, in all cases, the computation of sub-resultant sequences and computations with algebraic numbers. Instead, we compute, in particular, Gröbner bases and rational univariate representations. We show in the experiments in Chapter 4 the benefit of our choice when computing with non-generic curves. Furthermore, the philosophy of our approach is to avoid, as much as possible, computations that are time consuming in practice. This leads to various algorithmic choices such as avoiding the computation of y-critical points and allowing the curve to intersect the top and bottom sides of boxes isolating x-extreme points (which avoids substantial subdivisions since the tangent at an extreme point is vertical).

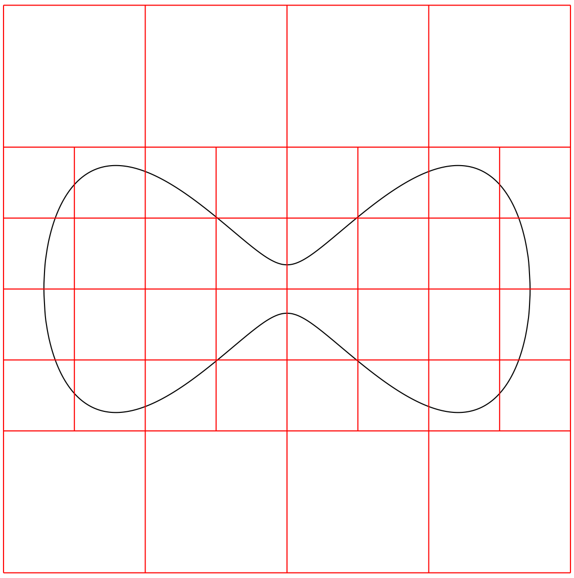
The novelty of our algorithm mainly relies upon the use of three new ingredients for this problem. First, we use some of the state-of-the-art techniques to isolate the roots of bivariate systems, *i.e.,* we use (i) Gröbner basis computations [74], (ii) Rational Univariate Representations (RUR) [140] which represent the roots of the system as rational



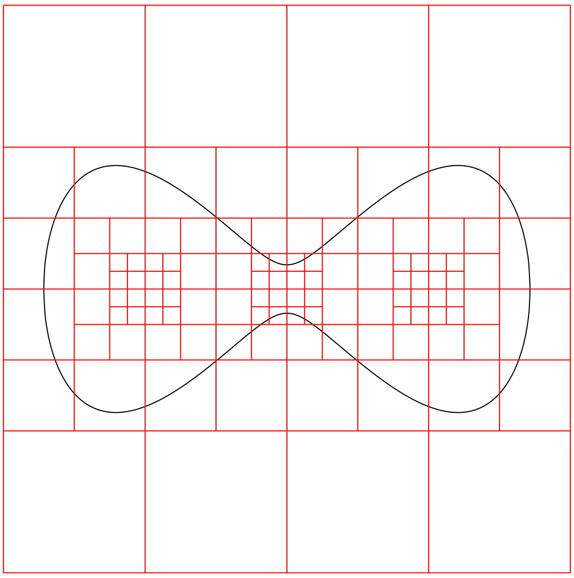
(i) first subdivision of the plane



(ii) second subdivision



(iii) subdivision of the interesting sections of the plane



(iv) final result

Figure 3.3: Typical behavior of subdivision methods.

functions of the roots of a univariate polynomial, and (iii) a subdivision technique based on Descartes' rule of signs (and filtered with interval arithmetic for efficiency) for isolating the roots of the univariate polynomials [141, 63, 70]. Even though this approach is well known for system solving, it was not used before for computing the topology of algebraic curves.

Second, we compute and use the multiplicities in fibers (see Definition 13) of critical points to compute the topology at singular points and to determine the connections at extreme points. For extreme points, we get these multiplicities by the RUR and a special case of a formula of Teissier [153]. For singular points, we solve the system of singular points with additional constraints. Note that the overall method to compute the topology at critical points is not new, it is described in full details in [147], see also for example [7, 97, 28, 120, 126, 3] for closely related approaches for curves in non-generic position. The novelty appears in the way we compute multiplicities in this context; once again we avoid computing sub-resultant sequences.

Third, we present a variant of the standard combinatorial part of the algorithm for computing the topology. We compute a decomposition of the plane (which is not a cylindrical algebraic decomposition) by rectangular boxes containing at most one critical point. Since we allow crossings on the top and bottom of extreme point boxes to avoid costly refinement of boxes, the connection is not always straightforward. To achieve the connection near such points, we use the multiplicities in fibers of extreme points. Note that one advantage of this variant is that, even when the curve is in generic position, the algorithm does not require to refine these boxes until they do not overlap in x.

With these tools, our algorithm for computing an arrangement of polylines $\mathcal{G}$ isotopic to a curve $\mathcal{C}$ can be summarized as follows (see Section 3.3 for details).

(i) Isolate the x-critical points in two dimensional rectangles, called critical boxes, using algebraic tools (Gröbner bases, RUR and Descartes' algorithm). Compute also the multiplicity of the critical points in their fibers. Refine the critical boxes until the restriction of $\mathcal{C}$ to each critical box is guaranteed to be a set of x-monotone non-crossing arcs connecting the critical point to a point on the boundary of the box.

(ii) Compute a rectangular decomposition of the plane by extending the vertical sides of the critical boxes either to infinity or to the first encountered critical box (see Figure 3.4). (Note that, for visualization purposes, a geometrically accurate picture of $\mathscr{C}$ can be easily obtained by further subdividing vertically the non-critical rectangles of the decomposition.) For every edge of this decomposition, determine its intersection with $\mathscr{C}$, that is, determine separating intervals each containing exactly one intersection point.

(iii) The vertices of $\mathscr{G}$ consist of the x-critical points of $\mathscr{C}$ and intersections of $\mathscr{C}$ with the edges of the rectangular decomposition. For every critical box of $\mathscr{G}$, connect (with a straight line segment) the critical vertex to the vertices on the boundary of the box. For every other rectangle of the decomposition, connect the vertices on its boundary using, if needed, the multiplicity of the extreme points in the neighboring rectangles combined with a greedy approach.

The output of the algorithm is an arrangement of polylines represented by the embedded graph $\mathscr{G}$. The vertices of $\mathscr{G}$ hence represent points whose coordinates are, in general, non-rational. Associated to each vertex, the algorithm also computes a box containing the represented point (the critical box for a critical point or the separating interval determined in Step ii for an intersection between the curve and a wall of the rectangular decomposition). These boxes can be refined, and any choice of point (for instance, with rational coordinates) in these boxes gives a graph isotopic to the curve.

3.2 Notation and definitions

The main algebraic tools, related to polynomial system solving, were already introduced in Chapter 2. This section introduces some concepts which are needed to understand the algorithm, as well as notation used along this chapter.

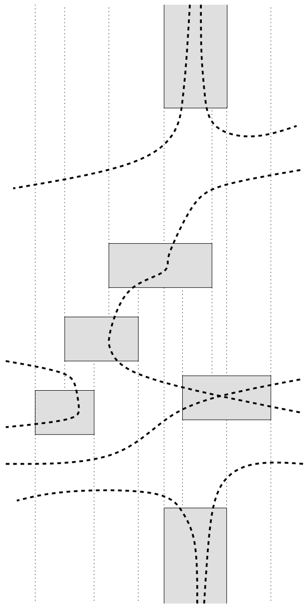


Figure 3.4: Example of rectangle decomposition of the plane induced by the isolating boxes (critical and asymptotes). There are boxes for the asymptote, the singular point, and the three extreme points, two of them with even multiplicities and one with odd multiplicity.

Let $\mathscr{C}$, also denoted $\mathscr{C}_f$, be a real algebraic plane curve defined by a bivariate polynomial f in $\mathbb{Q}[x, y]$. Since the geometry of the curve is not modified by taking the square-free part of f, we can assume without loss of generality that f is square-free. Note that $\mathscr{C}$ may consist of several algebraic components, that is, f is not necessarily irreducible in $\mathbb{R}[x, y]$. The algebraic components of the curve that are vertical lines (*i.e.*, lines parallel to the y-axis) can be easily computed since their abscissa correspond to the real roots of the polynomial in x obtained as the gcd of the coefficients of f seen as a polynomial in y.

Partial derivatives are denoted with subscripts: for instance, f_x denotes the derivative of f with respect to x and f_{y^k} (sometimes also f_k) denotes the k^{th} derivative with respect to y. A point $\mathrm{p} = (\alpha, \beta) \in \mathbb{C}^2$ is called x-critical if $f(\mathrm{p}) = f_y(\mathrm{p}) = 0$, singular if $f(\mathrm{p}) = f_x(\mathrm{p}) = f_y(\mathrm{p}) = 0$, and x-extreme if $f(\mathrm{p}) = f_y(\mathrm{p}) = 0$ and $f_x(\mathrm{p}) \neq 0$ (*i.e.*, it is x-critical and non-singular). Similarly are defined y-critical and y-extreme points. As x-critical and x-extreme points are more often used in the following, we often simply refer to them as critical and extreme points.

As mentioned in Chapter 2, the ideal generated by polynomials $P_1, \ldots, P_i$ is denoted $\langle P_1, \ldots P_i, \rangle$. In the following, we often identify the ideal and the system of equations $\{P_1 = 0, \ldots, P_i = 0\}$ (or any equivalent system induced by a set of generators of the ideal). We consider, in particular, the ideals $I_c = \langle f, f_y \rangle$ and $I_s = \langle f, f_x, f_y \rangle$; their roots are, respectively, the x-critical and singular points of $\mathscr{C}$.

We now recall the notion of multiplicity of the roots of an ideal, then we state two lemmas using this notion for studying the local topology at critical points. Geometrically, the notion of multiplicity of intersection of two regular curves is intuitive. If the intersection is transverse, the multiplicity is one; otherwise, it is greater than one and it measures the level of degeneracy of the tangential contact between the curves. Defining the multiplicity of the intersection of two curves at a point that is singular for one of them (or possibly both) is more involved and an abstract and general concept of multiplicity in an ideal is needed.

Definition 13 ([49] §4.2). *Let I be an ideal of $\mathbb{Q}[x, y]$ and denote $\overline{\mathbb{Q}}$ the algebraic closure of $\mathbb{Q}$. To each zero (α, β) of I corresponds a local ring $(\overline{\mathbb{Q}}[x, y]/I)_{(\alpha,\beta)}$ obtained by localizing the ring $\overline{\mathbb{Q}}[x, y]/I$ at the maximal ideal $\mathrm{I}(x - \alpha, y - \beta)$. When this local ring is finite dimensional as $\overline{\mathbb{Q}}$-vector space, we say that (α, β) is an isolated zero of I and this dimension is called the* multiplicity of (α, β) as a zero of I.

Let $f, g \in \mathbb{Q}[x, y]$ be such that the intersection of $\mathscr{C}_f$ and $\mathscr{C}_g$ in $\mathbb{C}^2$ contains a zero-dimensional component equal to point $\mathrm{p} = (\alpha, \beta)$. Then (α, β) is an isolated zero of $\langle f, g \rangle$ and its multiplicity, denoted by $\mathrm{Int}(f, g, \mathrm{p})$, is called the intersection multiplicity of the two curves at this point.

We call a fiber *a vertical line of equation $x = \alpha$. For a point $\mathrm{p} = (\alpha, \beta)$ on the curve $\mathscr{C}_f$, we call the multiplicity of β in the univariate polynomial $f(\alpha, y)$ the* multiplicity of p in its fiber *and denote it as* $\mathrm{mult}(f(\alpha, y), \beta)$.

The next lemma, due to Teissier [153], relates the multiplicity of a point in a fiber with the multiplicity in the critical ideal I_c. We will use it to deduce the multiplicity in the fiber knowing multiplicity in the ideal. More precisely, we will use the multiplicity in fibers of extreme points during the connection step of our algorithm.

Lemma 14 ([153][14, Lemma D.3.4 p.314]). *For an x-extreme point* $\mathrm{p} = (\alpha, \beta)$ *of* f*, one has*

$$\mathrm{mult}(f(\alpha, y), \beta) = \mathrm{Int}(f, f_y, \mathrm{p}) + 1. \tag{3.1}$$

To compute the local topology of the curve at a singular point, we aim at isolating the singular point in a box so that the intersection of its border and the curve determines the topology. Indeed for a small enough box, the topology is given by the connection of the singular point with all the intersections on the border. So the box shall avoid parts of the curve not connected to the singular point. Knowing the multiplicity of the singular point in the fiber enables to isolate the singular point from other crossings of the curve in this fiber. Requiring in addition that intersections with the curve only occur on the left or right sides of the box leads to the following.

Lemma 15 ([147]). *Let* $\mathrm{p} = (\alpha, \beta)$ *be a real singular point of the curve* $\mathscr{C}_f$ *of multiplicity* k *in its fiber. Let* B *be a box satisfying*

(i) B *contains* p *and no other* x*-critical point,*

(ii) the function f_{y^k} *does not vanish on* B*, and*

(iii) the curve $\mathscr{C}_f$ *crosses the border of* B *only on the left or the right sides.*

Then the topology of the curve in B *is given by connecting the singular point with all the intersections on the border.*

The proof of lemma 15 is based on a recursive application of the mean value theorem stating that the roots of the derivative of a polynomial P lie between those of P.

To conclude this section, it will be described how the tools introduced in Chapter 2 are used in the algorithm, to find roots of univariate and bivariate ideals.

In many places of the algorithm, it is necessary to count and/or isolate the real roots of univariate polynomials, possibly in a given interval. This is, in particular, needed for computing the intersections between $\mathscr{C}$ and the sides of the boxes isolating the critical points. Only polynomials with rational coefficients will be considered. The square-free part of the considered polynomials is first computed. The real roots are then isolated using recursive subdivision and the Descartes' rule of signs (see [63, 70, 141] for details).

It is needed to represent solutions of zero-dimensional ideals depending on two variables by boxes containing them. The rational univariate representation of the roots, introduced in Chapter 2, is used. This can be viewed as a univariate equivalent of the studied ideal. The key feature of this RUR is the ability to isolate solutions in easily refinable boxes and to compute multiplicities.

3.3 The algorithm

Neither the algorithm nor its implementation require assumptions about the existence of vertical components of the curve, but the processing of vertical lines is rather technical and, for clarity, its description will be postponed to Section 3.3.2. In other words, we first present our algorithm with the assumption that the input curve has no vertical components. A proof of correctness of the algorithm is presented in Section 3.3.3 and the algorithm is illustrated on an example in Section 3.3.4.

3.3.1 Curve without vertical lines

The input of this algorithm is, as discussed in Section 3.2, a real curve $\mathscr{C}$ without vertical lines and defined by a square-free polynomial $f \in \mathbb{Q}[x, y]$. In a few words, the algorithm first focuses on critical points, their rational univariate representations enable to compute multiplicities and boxes isolating each point with known topology inside the box. Then a sweep method computes a rectangular decomposition of the plane induced by the boxes of critical points. Eventually the connection is processed in all rectangles with a greedy method using multiplicities in fibers for extreme points. The algorithm is detailed in six steps.

Step 1. Isolating boxes of the singular points and of the x-extreme points. As a general practical rule, the smaller the number of solutions of a system, the easier it is to work with. Hence we split the system of critical points into the system of singular points and the system of extreme points. The system of singular points is the one of critical with in addition the equation $f_x = 0$. The system of extreme points, denoted I_e, is computed by saturation. Indeed, the extreme points are critical for which $f_x \neq 0$, thus we add to the critical system the equation $1 - uf_x = 0$ with a new variable u that we eliminate afterwards. We then compute the RURs of these systems I_s and I_e and isolating boxes for the solutions (see Lemma 23 in Section 6.1.3 for details on interval analysis). We may need to refine the boxes of extreme and singular points to avoid overlaps.

Step 2. Multiplicities of critical points in fibers. For extreme points, we use the Teissier formula: the multiplicity of an extreme point in I_c is the same than in I_e because precisely f_x does not vanish at these points. The multiplicity in I_e is given by the RUR, and hence the multiplicity of an extreme point in its fiber is this number plus one according to the equation of Lemma 14.

For singular points, we use the definition of univariate multiplicity, namely the smallest integer k such that the k^{th} derivative no longer vanishes. Let $I_{s,k}$ be the system of singular points with in addition the equations $f_{y^i} = 0$ for i from 2 to k. Hence we solve, for k increasing from 2, the systems $I_{s,k}$ until it has no solutions. At each step, a singular point which was a solution of $I_{s,k-1}$ but is no longer solution of $I_{s,k}$ has its multiplicity in fiber equal to k.[4] Note that the data of the systems $I_{s,k}$ will not be used later, they are only useful for the multiplicity computation. Theoretically, the complexity of solving these systems is analyzed in Chapter 6. In practice, as k increases, the systems have less and less solutions and hence tend to be easier to solve. Note also that the number of systems to solve is the highest multiplicity of the singular points of the curve.

Step 3. Refinement of the isolating boxes of the x-extreme points. Consider each such

[4]We also refer the interested reader to a more elegant way to compute the multiplicity in fibers with the Teissier formula [41]. Experimentally, it appears that this alternative was less efficient because even if it usually needs to work with less systems, these systems are larger (i.e. with more solutions).

box, B, in turn. For each vertical or horizontal side of B, isolate its intersections with $\mathscr{C}$ and refine the box until there are two intersection points. We further refine until there is at most one crossing on the top (resp. bottom) side of B. Note that, unlike comparable algorithms, we do not require that $\mathscr{C}$ intersects the boundary of B on its vertical sides. This is important in practice because, since the curve has a vertical tangent at an x-extreme point, refining until the curve intersects the vertical side is time consuming.

Step 4. **Refinement of the isolating boxes of the singular points**. We refine these boxes exactly as in [147] (see Lemma 15) except for the way the multiplicity k of each singular point in its fiber is computed. In [147], k is computed using Sturm-Habicht sequences under the assumption of generic position while we deduce k as explained in Step 2. Then, as in [147], every box is refined until the evaluation of f_{y^k} with interval arithmetic does not contain 0 (see Chapter 6 for details on interval analysis). Further refine the x-coordinates of the box until $\mathscr{C}$ only intersects the vertical boundary of the box.

Step 5. **Vertical asymptotes**. To determine the topology of a curve, it is required to know how many branches are going to infinity. However, it is not required, in general, to know which branch is related to which asymptote. Nevertheless, for our next connection step, we need to determine for each vertical asymptote which branches are related to it and if they are on its left or its right.

The x-coordinates of vertical asymptotes are the roots of the leading coefficient $V_a(x)$ of the polynomial $f(x, y)$ considered as a polynomial in y. To deal with an asymptote $x = \alpha$, the idea is, informally, to isolate the point (α, ∞) in a box $[a, b] \times [M, \ldots, \infty, \ldots, -M]$ whose vertical sides do not intersect the curve $\mathscr{C}$. Moreover, we want that every branch that intersects a horizontal side of the box is a branch going to $\pm\infty$ with this asymptote (see Figure 3.4). First, compute an upper bound M_y on the absolute value of the y-coordinates of the y-critical points (this is of course done without computing these critical points, but only the discriminant with respect to y and an upper bound of the roots of this univariate polynomial). Compute also a bound M_x on the absolute value of the y-coordinates of the x-critical points (for which we have already computed boxes). Isolate the roots of the polynomial V_a, hence each root α has an isolating interval $[a, b]$. Substitute $x = a$ (resp. $x = b$) in f and deduce an upper bound, M_1, on the absolute value of the y-coordinates of the intersection of $\mathscr{C}$ and $x = a$ (resp. $x = b$). Set $M = \max(M_1, M_x, M_y)$. Then, a branch crossing the segment $]a, b[\times M$ (resp. $]a, b[\times -M$) goes to $+\infty$ (resp. $-\infty$) with asymptote $x = \alpha$. Finally, we determine whether a given branch is to the left or to the right of the asymptote by comparing the x-coordinates of the asymptote and the crossing point.

Step 6. **Connections**. For simplicity, all the boxes computed above are called critical boxes and the points at infinity on vertical asymptotes are also called critical. First compute, with a sweep-line algorithm, the vertical rectangular decomposition obtained by extending the vertical sides of the critical boxes either to infinity or to the first encountered critical box (see Figure 3.4). On each of the edges of the decomposition, isolate the intersections with $\mathscr{C}$.[5] Create vertices in the graph corresponding to these intersection points and to the critical points. For describing the arcs connecting these vertices in the graph, we assimilate, for simplicity, the points and the graph vertices. Inside each critical box, the topology has been made as simple as possible: one has just to connect the critical point to the points on the boundary of the box.

There are several approaches to do the connections in the other rectangles of the decomposition. The usual and conceptually simplest is to refine boxes of extreme points so as to avoid top and bottom crossings; then, the number of left and right crossings in rectangles always match and the connection is one-to-one. Since we allow some top/bottom crossings for efficiency reasons (see

[5]For simplicity, we use refinements to ensure that the curve never intersects an endpoint of an edge, that is, a corner of a rectangle. Note also that the intersections are already isolated on the sides of the critical boxes; an isolating interval may, however, need to be refined if it contains a vertex of the rectangle decomposition.

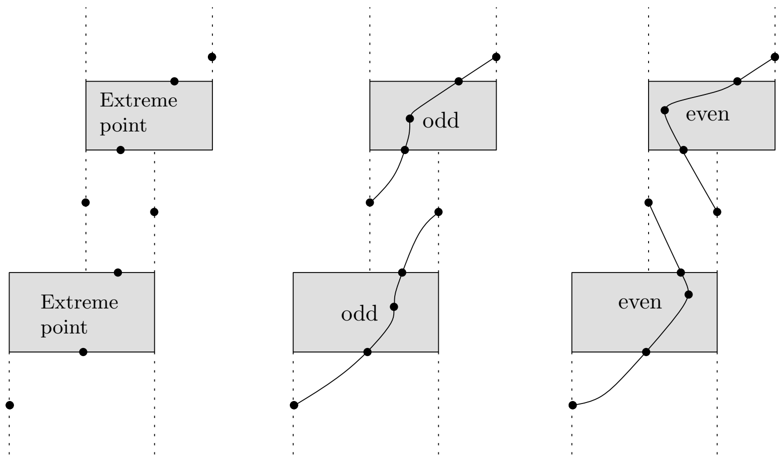


Figure 3.5: Possible connections involving extreme points depending on their multiplicities.

above), this straightforward method does not apply. Another approach (see [3, 143]) is to compute the sign of the slope of the tangent to the curve at the top/bottom crossings (this yields whether the top/bottom crossing should be connected to vertex to the right or to the left of its rectangle). We however want to avoid such additional computations.

For computing the connections in the non-critical rectangles of the decomposition, we use the multiplicities in fibers of the extreme points and a greedy algorithm. The geometric meaning of the parity of this multiplicity is the following: if it is even, the curve makes a U-turn at the extreme point, else it is odd and the curve is x-monotone in the neighborhood of the extreme point. Still, there are some difficulties for connecting the vertices, as illustrated on Figure 3.5: on the left is the information we may have on the crossings for two extreme points with x-overlapping boxes; the second and third drawings are two possible connections *in the middle rectangle* for different parities of the multiplicities. To distinguish between these two situations, we compute the connections in rectangles starting from the top such that the connections in a rectangle below a critical box are computed once the connections in all the rectangles above the box are done.

The connections can easily be computed as follows. First, if there are vertical asymptotes, we have already determined in Step 5 whether a point that lies on the boundary of an asymptote box belongs to a branch that lies to the left or to the right of its asymptote; recall that such a point lies on a horizontal side of the box. Consider a rectangle R of the decomposition that is adjacent to an asymptote box, say below it (the case where it is above is similar); note that the top wall of R is a subset of the bottom wall of the asymptote box. The vertices on the top wall of R split into k_l vertices that are left of the asymptote and k_r vertices that are right of it. Each of the k_l vertices necessarily connects to a vertex on the left or bottom side of R. Moreover, among these k_l vertices, the i-th vertex starting from the left, connects to the i-th vertex on the left-bottom sides of R starting from the top. Similarly, among these k_r vertices, the i-th vertex starting from the right, connects to the i-th vertex on the right-bottom sides of R starting from the top (see Figure 3.4).

Once these connections for asymptotes are done, due to requirements on extreme point boxes, there is at most one, not already connected, vertex on the *top or bottom* of any rectangle. The problem now is to determine if such a vertex should be connected to a vertex on the right or on the left side of the rectangle. The connections in the unbounded rectangles above critical boxes are straightforward: the ones between the vertices on the two vertical sides are in one-to-one correspondence, starting from infinity, and if a vertex remains on a vertical side, then there is a vertex on the horizontal side which it has to be connected with. Now, once all the connections have been computed in the rectangle(s) above the box of an extreme point, these connections and the multiplicity of the extreme point allow us to compute the connections in the rectangle(s) below, see Figure 3.5. Indeed, if there is a vertex on the bottom side of the critical box, then it lies on the top side of a rectangle. Inside this rectangle, the vertex is connected to the topmost vertex on the left or on the right side, depending on the multiplicity of the extreme point and on the side of the connection of the branch above the extreme point. The other connections in this rectangle and in the other rectangles below the critical box, if any, are performed similarly as for unbounded rectangles. Note that the two unbounded rectangles (the leftmost and the rightmost) that are vertical half-planes are treated separately: for each vertex on the vertical side we associate an arc that goes to infinity.

The output of the algorithm is a graph isotopic to the curve. In addition, x-extreme points, singular points and vertical asymptotes are identified and their position is approximated by boxes, hence refinable to any desired precision.

3.3.2 Curve with vertical lines

It was assumed in the previous section that the curve $\mathscr{C}$ has no vertical line. In order to generalize the algorithm, it is explained now how to calculate the topology of a curve $\mathscr{C}_F$ defined by a rational bivariate polynomial $F(x, y) = 0$, which has vertical lines. The idea is first to process the curve without its vertical lines, then study the intersections of this curve with the vertical lines. Technically, these two processings are intertwined.

Step V1. The vertical lines have as x-coordinates the roots of V_l, the gcd of the coefficients of F seen as a polynomial in y. V_l is an univariate polynomial in x, its roots are isolated (vertical lines can be thought of as vertical *stripes*). The curve $\mathscr{C}_f$ with $f = \dfrac{F}{V_l}$ which has no vertical line, is processed as explained before, until Step 4 included.

Step V2. Intersections between $\mathscr{C}_f$ and the vertical lines generically occur on non-critical points of $\mathscr{C}_f$. Nevertheless, this may not always be the case and we need to identify critical points of $\mathscr{C}_f$ that also are on vertical lines. Solving the singular and extreme ideals of $\mathscr{C}_f$ with the additional polynomial V_l enable to identify which critical point of $\mathscr{C}_f$ is on which vertical line. Note that these points are singular for $\mathscr{C}_F$.

Step V3. New relevant points of $\mathscr{C}_F$ must be enclosed in boxes, and all boxes (old and new) need to be refined again to meet the criteria of the connection step of the former algorithm. In more details, the x-intervals of critical points of $\mathscr{C}_f$ and vertical line stripes are refined until a vertical line stripe overlaps a critical box if and only if this critical point is on the line.

Create new boxes, called *vertical*, that contain every point that is an intersection between a vertical line and $\mathscr{C}_f$ and that is not a critical point of $\mathscr{C}_f$. Refine the extreme point boxes of $\mathscr{C}_f$ and vertical boxes until there is at most one crossing with $\mathscr{C}_F$ on top (and bottom).

Step V4. Then, Step 5 for vertical asymptotes of the previous algorithm is performed, with the following modifications:

(i) for the computation of the bound M_x, consider all vertical boxes in addition to critical boxes,

(ii) identify which vertical lines are also asymptotes, by computing the gcd of V_l and V_a,

(iii) refine then the stripes corresponding to vertical lines and asymptote boxes, so that a vertical stripe, whose line is not an asymptote, does not overlap any asymptote box, and

(iv) add crossing points on asymptote boxes whose asymptote is also a vertical line.

Step V5. Finally, Step 6 of the previous algorithm performs the connection. The output is a graph isotopic to the curve $\mathscr{C}_F$.

3.3.3 Correctness of the algorithm

All algebraic computations are certified, hence the only thing that has to be proved is that the output graph $\mathscr{G}$ is isotopic to the input curve $\mathscr{C}$. Our proof is constructive and elementary, we define the ambient isotopy F as follows. On the skeleton of the rectangle decomposition F_t is chosen to be the identity for any t. It remains to define F on each rectangle. In a rectangle that is not a critical box, $\mathscr{C}$ is a set of x-monotone non-crossing arcs and $\mathscr{G}$ is a set of straight line segments connecting the same pairs of points on the rectangle's boundary, as $\mathscr{C}$ does. F_1 is defined on each section $x = \alpha$ by mapping the points of $\mathscr{C}$ to that of $\mathscr{G}$ so that their ordering on $x = \alpha$ is preserved, and by linear interpolation on the other points. F_t is then defined by linear interpolation $F_t = t \operatorname{Id} + (1 - t)F_1$. To handle critical boxes, note that once cut vertically at the critical point, they behave like other rectangles (by Lemma 15).

3.3.4 Example

In this section, the algorithm is illustrated by an example. The chosen curve is $f(x, y) = y^4 - 6y^2x + x^2 - 4y^2x^2 + 24x^3$, as in [86]. The numeric results shown in this section were obtained with our implementation of this algorithm, which will be described in Chapter 4.

Before starting, one has to eliminate from the curve the vertical components. These vertical components are the gcd of the coefficients of f seen as a univariate polynomial in y. This gcd is 1: the curve has no vertical components and the algorithm will behave as in Section 3.3.1.

Step 1 of the algorithm calculates boxes for the critical points of the curve f. First, the system of singular points I_s is solved.

$$\begin{aligned} I_s &= \langle f, f_y, f_x \rangle \\ &= \langle y^4 - 6y^2x + x^2 - 4y^2x^2 + 24x^3,\ 4y^3 - 12xy - 8yx^2,\ -6y^2 + 2x - 8y^2x + 72x^2 \rangle \end{aligned}$$

The Gröbner basis of the system I_s with respect to the lexicographic ordering is calculated, giving $(3y^2 - x, xy, x^2)$. The RUR of this basis is the following; recall that

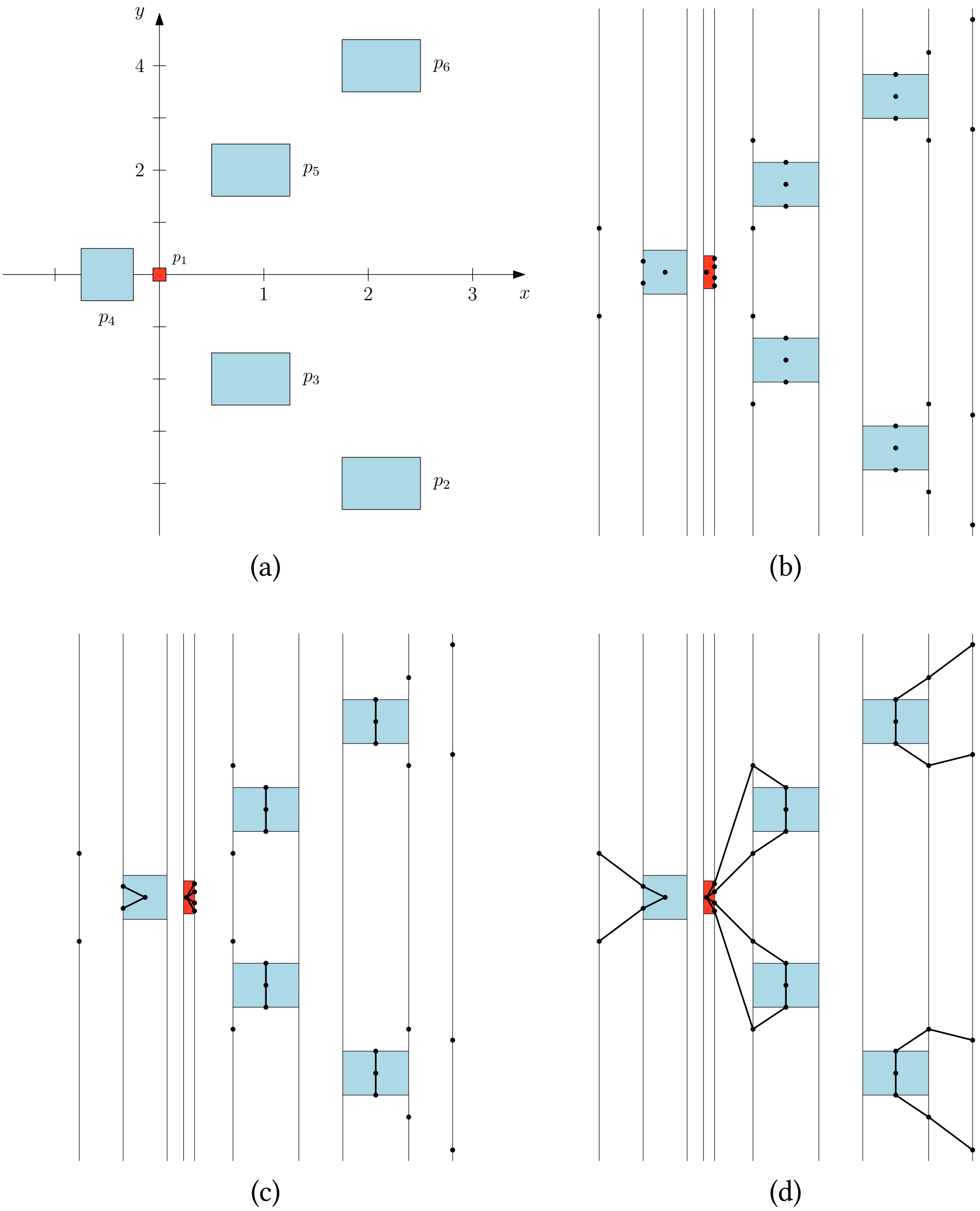


Figure 3.6: (a) Critical boxes of f. The small square at the origin contains the only singular point, p_1. Each other rectangles contains one of the x-extreme points $p_2, \ldots, p_6$. (b) Rectangular decomposition of the plane induced by the critical boxes. (c) Topology inside the critical boxes. (d) Graph isotopic to the curve.

the solutions of the system are $\left(x = \frac{G_1(t)}{G_0(t)}, y = \frac{G_2(t)}{G_0(t)}\right)$ for t solution of $F(t) = 0$.

$$F(t) = t^3, \quad G_0(t) = 3, \quad G_1(t) = 0, \quad G_2(t) = 3t.$$

As F has only one root, $t = 0$, the system I_s of singular points has only one solution $p_1 = (0, 0)$. Accordingly, our implementation reports that I_s has only one root in the box $[0, 0] \times [0, 0]$.

The system I_e of extreme points is given by $I_e = \langle f, f_y, 1 - u f_x \rangle$; here the new variable u is added, ensuring that $f_x \neq 0$ for any solution of the system (indeed if $f_x = 0$ then $1 - u f_x = 1$ for any u in $\mathbb{C}$). The system I_e is then

$$I_e = \langle y^4 - 6y^2x + x^2 - 4y^2x^2 + 24x^3, \quad 4y^3 - 12xy - 8yx^2, \quad 1 - u(-6y^2 + 2x - 8y^2x + 72x^2) \rangle.$$

To solve this system, we compute a Gröbner basis eliminating u and with lexicographical ordering on the other variables, giving $(72x^2 + 4 - 35y^2 + 99x, y^3 - 9xy + 4y, 3y^2x + 2 - 13y^2 + 48x)$. The associated RUR is

$$\begin{aligned} F(t) &= t^5 - 19t^3 + 70t, & G_0(t) &= 1680 + 120t^4 - 1368t^2, \\ G_1(t) &= -70 + 143t^4 - 1133t^2, & G_2(t) &= 912t^3 - 6720t. \end{aligned}$$

F has five real roots, and each one of them maps to a root of the system I_e. The software reports (small) boxes containing each one of these extreme points $p_2, \ldots, p_6$. For clarity of the exposition, we consider enlarged versions of these critical boxes, as shown in Figure 3.6 (a).

At this point, the isolating boxes of the extreme points are pairwise disjoint (and similarly for the singular boxes if there were more than one) but nothing ensures that the extreme boxes do not overlap with the singular boxes. The algorithm thus refine the boxes (by refining the isolating intervals of the roots of polynomials $F(t)$ in the RURs of I_e and I_s, and using interval arithmetic to obtain the refined 2D boxes) until all the boxes are pairwise disjoint. In this example, no refinement is needed because the boxes do not overlap.

The fact that the solver found exact coordinates for some points, here p_1, is actually a difficulty because, for instance, the number of intersection points of the curve with the boundary of a singular box does not yield the number of branches that are incident to the singular point. Point boxes are thus enlarged, initially to $\left[-\frac{1}{128}, \frac{1}{128}\right] \times \left[-\frac{1}{128}, \frac{1}{128}\right]$, and refined until they intersect no other critical boxes. This yields a box for p_1 which is $\left[-\frac{1}{4096}, \frac{1}{4096}\right] \times \left[-\frac{1}{4096}, \frac{1}{4096}\right]$.

In **Step 2**, the algorithm calculates the multiplicities of the critical points in fibers. For the singular point p_1, it calculates the smallest integer k such that the k^{th} derivative does not vanish. This is done by considering the systems $I_{s,k}$ obtained by adding f_{y^k} to the system $I_{s,k-1}$ with $I_{s,1} = I_s$ (for efficiency purpose, we actually add f_{y^k} to the Gröbner basis of $I_{s,k-1}$ which has already been computed when considering $I_{s,k}$).

Starting from $k = 2$, the solutions of $I_{s,k}$ are computed via a Gröbner and RUR calculations. Note that the solutions of $I_{s,k}$ are also solutions of $I_{s,k-1}$. The isolating boxes of the solutions of $I_{s,k-1}$ and $I_{s,k}$ are then refined until every box of each system intersects at most one box of the other system. This ensures that two intersecting boxes necessarily correspond to the same root of the two systems. We can thus easily decide whether a root of $I_{s,1}$ is also a root of $I_{s,2}, I_{s,3}, \ldots$ In our example the situation is quite simpler because $I_{s,1}$ has only one root, and thus any root of $I_{s,k}$ is necessarily that one. Still, computing the solutions of $I_{s,k}$ starting from $k = 2$, yields that the solution p_1 of $I_{s,1}$ is a solution of $I_{s,2}$ and $I_{s,3}$, but not of $I_{s,4}$. Hence, p_1 has multiplicity 4 in its fiber. Here, the systems $I_{s,k}$ and their Gröbner bases $Gb_{I_{s,k}}$ are:

$$
\begin{aligned}
&I_{s,1} = I_s && Gb_{I_{s,1}} = \{3y^2 - x,\ xy,\ x^2\} \\
&I_{s,2} = I(3y^2 - x,\ xy,\ x^2,\ 12y^2 - 12x - 8x^2) && Gb_{I_{s,2}} = \{x,\ y^2\} \\
&I_{s,3} = I(x,\ y^2,\ 24y) && Gb_{I_{s,3}} = \{y,\ x\} \\
&I_{s,4} = I(y,\ x,\ 24) && Gb_{I_{s,4}} = \{1\}
\end{aligned}
$$

On the other hand, the multiplicities of the five x-extreme points are computed using Teissier formula (see Lemma 14): the multiplicity of each point in its fiber is the multiplicity of the corresponding roots in the RUR of I_e plus one. In this case, $p_2, \ldots, p_6$ have multiplicity one in the RUR, which implies that they all have multiplicity 2 in their fibers.

Step 3 of the algorithm deals with the refinement of the boxes containing x-extreme points, that is, the boxes containing $p_2, \ldots, p_6$. The goal of this step is to obtain boxes such that the curve intersects every box's boundary at most twice, with at most one intersection on the top (resp. bottom) of the box. Each box is treated independently.

The intervals defining the box enclosing the point p_2, computed during the first step of the algorithm, are

$$x_{p_2} = \left[\frac{4611611823926328587}{2305843009213693952}, \frac{4611761039352188815}{2305843009213693952}\right]$$

and

$$y_{p_2} = \left[-\frac{8627721659600529273}{2305843009213693952}, -\frac{8627594605412894855}{2305843009213693952}\right].$$

The algorithm isolates the roots on the vertical walls, $f(x_{p_2,left}, y)$ and $f(x_{p_2,right}, y)$ in the interval y_{p_2}, and similarly for the horizontal walls. Two intersection points are found, one on the top wall, the other on the bottom wall. This means that the box containing p_2 does not need to be refined.

The situation is identical for the boxes containing p_3, p_5 and p_6. The box containing p_4 has also two intersections with the curve, here both on the left wall, which again does not require further refinement.

In **Step 4**, the algorithm refines the boxes enclosing singular points. The method uses the multiplicity k of each singular point (in this case, the only singular point is p_1) and the k^{th} derivative of the curve with respect to y (both were calculated in Step 2). Every singular point is treated independently; here with have only one.

We use interval arithmetic to ensure that f_{y^k} does not vanish in a box: computing with interval arithmetic, if the image of the box by f_{y^k} does not contain zero, then f_{y^k}

does not vanish in the box (the converse is not true). We thus refine the box until its image by f_{y^k} does not contain zero.

We then refine the box in x until the curve $\mathscr{C}$ intersects neither the top nor the bottom of the box. Actually, before doing this last refinement, for efficiency purposes, we actually enlarge the box in y as much as we can, under the two constraints that it should continue to avoid the curve f_{y^k} and all the other critical boxes. The final resulting box of p_1 is $\left[-\frac{1}{4096}, \frac{1}{4096}\right] \times \left[-\frac{1}{4}, \frac{1}{4}\right]$ and it was not necessary to refine the box in x.

Step 5 deals with vertical asymptotes. Their x-coordinates are the roots of the leading coefficient of $f(x, y)$ view as a polynomial in y. The leading term of f is y^4, what means that the leading coefficient is 1, that is, the curve f has no vertical asymptotes.

Step 6 is the last stage of the algorithm, in which the graph isotopic to the input curve $\mathscr{C}$ is computed. The boxes containing critical points induce a subdivision of the plane in rectangles, as shown in Figure 3.6 (b), and the intersection points between $\mathscr{C}$ and every wall of this subdivision are computed. As explained in Section 3.3.1, we connect vertices in a straightforward manner inside the critical boxes (see Figure 3.6 (c)). In this example, the connections in the other rectangles of the subdivision are also straightforward (see Figure 3.6 (d)) and we do not need to use the multiplicity of the extreme points nor to use a greedy approach.

3.4 Embedding the graph on a grid

The output of the algorithm in Section 3.3.2 is a graph isotopic to the input curve. The vertices of this graph represent points of the curve, whose coordinates are given as algebraic numbers. This information is given by disjoint boxes containing each vertex. Choosing any point inside each box will maintain the isotopy of the graph and the curve. Our graph can thus be seen as a graph where the vertices have algebraic coordinates and lie on the curve, or where the vertices are chosen anywhere in the isolating boxes (with rational coordinates). Note that algebraic coordinates can also be rational.

An important problem consists in computing approximate positions for the vertices with fixed precision while preserving the topology (actually, the isotopy) of the graph. This problem can be described as the embedding of the graph vertices on a *grid*. Given a grid, all vertices of the geometric graph are moved to distinct grid crossings. In practice, a grid can be determined by the pixels of a screen or by the numbers that fit in a double-precision floating-point number.

A similar problem was studied by Milenkovic and Nackman [124]. They considered the problem of finding fixed-size approximations of families of polygons. They showed that rounding the real numbers which define the family of polygons to a fixed precision while preserving the combinatorial structure of the family is in NP-complete.

We first recall Milenkovic and Nackman's result. It is stated in terms of (P, K)-approximations to numbers, that is, P-bit approximation to numbers in which the first $P - K$ digits are correct (identical to the original number). They consider approximations of sets of non-intersecting *simple* (non-self-intersecting) polygons. These polygons are defined by ordered sequences of the lines that contain their edges, and the approximation applies to the coefficients of these lines. We will call such polygons *line-based*.

Theorem 16 ([124]). *The language of sets of line-based simple polygons with at least one* (P, K)*-approximation is NP-complete.*

Milenkovic and Nackman prove that the problem is in NP-complete by proving that it is in NP and in NP-hard. Nevertheless, there is a small issue in the statement of the result, because the coordinates of the lines are given by real numbers, and they claim that a candidate solution can be verified in polynomial time. This is not quite true in the Turing model of computation (which is implicitly used by Milenkovic and Nackman), since verifying that the first $P - K$ bits of a rational number are equal to the first $P - K$ bits of a real number cannot be done in polynomial time. It was proven that some real numbers cannot even be computed [157] (see also [39] for an example of a real number that can be defined but cannot be computed). Nevertheless, their proof is correct if we state that lines are defined by algebraic coefficients instead of real coefficients. Algebraic numbers can be computed and approximated as much as desired in polynomial time.

The second part of the proof in [124] is very elegant. They reduce a graph coloring problem, known to be in NP-hard, to their problem. The elegance of the proof comes from the fact that the reduction consists in transforming the input graph into an electronic circuit. All the constructions of the proof are done with real numbers. However, all these constructions can be done with algebraic numbers. In conclusion, Theorem 16 is valid if lines are defined by algebraic coefficients and not by real coefficients.

Though very useful, this result does not apply directly to our problem, since polygons in [124] are defined by the lines that contain their edges. They consider approximations to the coefficients of those lines, and not approximations to the points as we want. Finding a (P, K)-approximation to the coordinates of a point given as algebraic numbers means to embed it in a particular grid. Figure 3.7 shows a geometrical interpretation of a (P, K)-approximation of a point. Given a point q with algebraic coordinates, the coordinates of the point q_a are the first $P - K$ bits of the coordinates of q. The thick grid represents all the points whose coordinates are the first $P - K$ bits of an algebraic number. Finding a (P, K)-approximation means to construct a P-bit number, by adding K bits to the numbers of $P - K$ bits. Those numbers are the x and y-coordinates of the thin grid. Finding (P, K)-approximations of a point means to embed it on a square grid as described. There are, on the grid of figure 3.7, 2^P choices for each coordinate, and thus 2^{2P} different choices for each point. Finding a (P, K)-approximation of a simple polygon given by the coordinates of its vertices thus means to find (P, K)-approximations of the coordinates of its vertices, while maintaining its topology. As each point has 2^{2P} possible (P, K)-approximations, a polygon with n vertices has 2^{2Pn} possible (P, K)-approximations.

Milenkovic and Nackman point out that this difference in representation is "of little consequence", since these are equivalent under a dual transform (for instance, the standard duality that maps a line of equation $ax + by + c = 0$ to a point (a, b)). See [59, § 15.2] for details on duality. Nevertheless, there is an issue because the simplicity of polygons is not necessarily preserved under this transformation.

Another approach to check whether approximating lines is equivalent to approximating point coordinates consists in observing the expression of point coordinates in terms of line coefficients. In particular, when a polygon is given by the equations of the

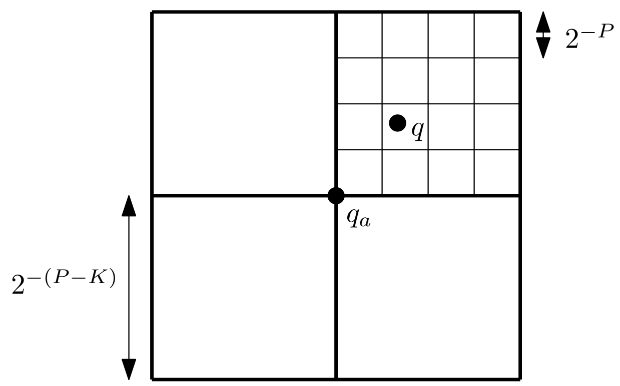


Figure 3.7: Geometrical interpretation of (P, K)-approximations to point coordinates. For simplicity, the point q_a has positive coordinates.

lines that contain its edges, each vertex is the intersection of two of these lines. In our case, in order to maintain the isotopy, each intersection point of approximated lines must be *close* to the intersection point of the original lines. Nevertheless, this is not always the case. For instance, two almost-parallel lines, when approximated, may give an intersection point which is far away from the original intersection point. Thus, this approach does not suffice to show that approximating line coefficients is equivalent to approximating coordinates of points.

Despite these considerations, we believe that these two problems are equivalent. We thus state it as the following conjecture. We call a polygon *point-based* if it is defined by the coordinates of their vertices.

Conjecture 17. *The language of sets of point-based simple polygons with at least one (P, K)-approximation is NP-complete.*

As pointed out before, vertices coordinates must be given as algebraic numbers for Conjecture 17 to be valid.

We return to the problem of embedding the geometric graph, output by the algorithm in Section 3.3.2, on a grid. Each vertex of this geometric graph represents a point of the curve, whose coordinates are algebraic numbers. However, the information associated to each vertex of the geometric graph consists in a box containing the point that this vertex represents and the associated polynomial system, so that the box can be refined as much as desired. In particular, they can be smaller than the given size of the grid in which the graph vertices must be embedded, and so boxes may contain no grid point.

Note, however, that any set of simple polygons is an instance of our geometric graph. We can thus formulate a new natural conjecture, and prove that it is valid if Conjecture 17 is valid. We will consider the bounding box (in which the algebraic curve was considered) as a part of the graph, and the isolated vertices as degenerate polygons.

Conjecture 18. *The language of point-based geometric graphs with at least one (P, K)-approximation is NP-complete.*

Proof. The proof consists of two parts. In the first part, we show that our problem is in NP-hard, while in the second we show that it is in NP.

Demonstrating that our problem is in NP-hard is straightforward, and is done by reducing in polynomial time a problem known to be in NP-hard to our problem. As said before, any set of simple polygons is an instance of our problem. The answer to our problem will be affirmative for this input if and only if the answer to the problem in Conjecture 17 is affirmative for this input. Thus, Milenkovic and Nackman's problem is reducible to our problem in constant time, and our problem is in NP-hard if Conjecture 17 is valid.

Our problem is in NP if and only if a candidate solution can be verified in polynomial time. This is done in two steps: verifying that the coordinates of each vertex are (P, K)-approximations of the algebraic numbers and checking that the combinatorial structure of the graph is maintained.

Verifying that the coordinates of each one of the n vertices are (P, K)-approximations to the algebraic coordinates is done in various steps. First, compute (in polynomial time) the first P bits of each coordinate. Then, compare these P bits with the approximated number, in linear time in P.

Checking the combinatorial structure of the graph is done in two steps. First, a standard sweep-line algorithm [16] is used to check in polynomial time whether there are no new intersections between the segments of the embedded graph (the only permitted intersections are those between endpoints of segments).

The final step consists in constructing the combinatorial structure of both graphs (the original and the embedded one) and check that they are the same. The combinatorial structure consists in a forest of trees, analogous to that of [124]. The nodes of the trees are simple polygons, with the property that if a polygon L shares an edge with a polygon M geometrically included in L, then L and M are the same polygon. A polygon K is a descendant of a polygon J in some tree if and only if J is geometrically included in K. This combinatorial structure is computed by using again the standard sweep-line algorithm, updating the forest of trees at each event point. Each update is done in logarithmic time, and comparisons are done in linear time. Finally, both forests of trees can be compared in linear time because nodes are labelled and thus the siblings of each node can be sorted when constructing and updating the trees. Since a candidate solution can be verified in polynomial time, the problem is in NP.

Since the problem is in NP-hard and in NP, we conclude that the problem is in NP-complete, as long as Conjecture 17 is valid. □

In other words, this conjecture states that it is not easy to round the geometric information of the output geometric straight-line graph of our algorithm and preserve at the same time the topology. For instance, it may be the case that critical points of the curve are very close to each other, thus having points of a big bitsize becomes unavoidable.

This has an impact on visualization. A direct implication of this result is that positions of points cannot be easily altered in order to make a plotted graph clearer. When facing a curve of which the topology is unknown, it would be nice to visualize a graph showing its topology, and all its critical points at the same time. But our conjecture claims that, roughly speaking, displaying the “farthest” critical points in their correct position (using a rounded value, for instance) and to move the “closest” critical points so as to clearly display their topology is a NP-complete problem.

3.5 Conclusion

We introduced in this chapter a new algorithm to compute the topology of plane algebraic curves. This algorithm makes use of algebraic techniques such as Gröbner bases, rational univariate representations, Descartes' univariate polynomial root isolations and interval arithmetic.

A natural question would be how these algebraic techniques can be applied to compute the topology of algebraic curves in three dimensions. The algebraic part of the algorithm in space, that is the computation of the critical points of the curves, can also be solved by means of Gröbner bases, rational univariate representations and Descartes' isolations. The application of these techniques for solving systems in three dimensions permits, in particular, to compute the critical points of the curves without computing all the critical points of some projection of the curves. It also permits to avoid special treatment of non-generic cases. Solutions of the systems would be enclosed in parallelepipeds instead of in rectangular boxes. However, connecting the critical points remains a challenge in three dimensions.

The complexity analysis of this algorithm is deferred to Chapter 6. The reason is that another algorithm is considered in the sequel, and all complexity analyses are grouped in the same chapter. The next chapter presents an implementation of the algorithm we have presented in this chapter and comparisons with other implementations based on different techniques. The most relevant aspect of these comparisons is that they validate our approach, proving the claim that our algorithm performs very well either in generic and non-generic cases.

Chapter 4

Maple Implementation and Experiments

> I conclude that there are two ways of constructing a software design: One way is to make it so simple that there are obviously no deficiencies and the other way is to make it so complicated that there are no obvious deficiencies.
> The first method is far more difficult. It demands the same skill, devotion, insight, and even inspiration as the discovery of the simple physical laws which underlie the complex phenomena of nature. It also requires a willingness to accept objectives which are limited by physical, logical, and technological constraints, and to accept a compromise when conflicting objectives cannot be met.
>
> C.A.R. Hoare [95]

This chapter presents an implementation of the algorithm to compute the topology of planar algebraic curves. Some considerations on efficiency are given. The behavior of the implementation is depicted with examples. A thorough comparison with implementations of other algorithms confirms that modern tools and libraries make this implementation very competitive.

The experiments presented in this chapter appeared in [42] and [40]. We also present here a description of the software.

The chapter is organized as follows. Section 4.1 presents the interface of ISOTOP, the MAPLE implementation of our algorithm. Section 4.2 discusses the software used for the development. Section 4.3 shows the execution of an example with ISOTOP. Section 4.4 describes thorough experiments we did to compare ISOTOP with other existing implementations. The results of these comparisons are then summarized, since the amount of data obtained is not easy to interpret. Nevertheless, a complete description of the tested curves and experimental results are presented in Appendix A for completeness.

4.1 ISOTOP interface

To introduce ISOTOP, we begin by describing its user interface. This section is based on the documentation shipped with ISOTOP. As this software was developed in MAPLE [118], we assume that the reader is familiar with this computer algebra system. If not, we refer to [119].

The only function available to the user is `topology_real_curve`. It should be called with a bivariate rational polynomial in x and y as mandatory parameter. The function call `topology_real_curve(`$\mathscr{C}$`)` returns a graph encoding the topology of the real curve defined in the plane by the equation $\mathscr{C} = 0$. It also displays a plot of this graph in a region of the plane such that all connected components and critical points are shown, or in a user-defined region. Singular points of the curve (points where the tangent is not defined) are displayed in red, and x-extreme points (points where the tangent is vertical, that is, parallel to the y-axis) are displayed in green. For convenience, the input curve is not assumed to be square-free and ISOTOP starts by computing its square-free part.

The function `topology_real_curve` can also be called with some of the following optional parameters:

- `verbosity`: integer in the range $0, \ldots, 3$ (default is `verbosity=0`, the minimum verbose level);

- precision: integer which specifies a lower bound on the number of digits of the significand of the decimal floating-point representation of the vertices coordinates correctly computed (default is precision=2);

- plot_graph: boolean; the output graph is visualized when plot_graph=true (which is the default setting);

- nb_splits: positive integer; the graph is computed from a rectangular decomposition of the plane induced by the critical points and for a better visualization we split vertically each rectangle nb_splits−1 times, resulting in nb_splits rectangles (default is nb_splits=10);

- view: a list containing two ranges, in the form [x_min..x_max,y_min..y_max] specifying the region of the plane where to visualize the plot (when not specified, these values are determined in such a way that all points of the output graph are shown in the plot).

When one is only interested in the topology of the curve, but not in its display, the options plot_graph=false and nb_splits=1 make the computation more efficient. On the other hand, if one wants to increase the quality of the approximation of the displayed graph, the value of the option nb_splits can be increased, but this slows down the computations.

The vertices of the returned MAPLE graph have information defining an embedding in the real plane. The point type attribute of each vertex is set to singular for vertices representing singular points, to extreme for vertices representing x-extreme points and regular for other vertices representing regular points of the curve. The coordinates (x, y) of each vertex, stored in the attribute coordinates, are approximations of the coordinates of the corresponding point of the curve; the number of correct digits is controlled, and bounded from below, by the optional value precision. The resulting embedding of the graph and the curve are isotopic in a bounding box witnessing all connected components, singularities, and x-extreme points.

4.2 Implementation

We used the MAPLE computer algebra system [118] to implement our algorithm, referred to as ISOTOP. This choice was motivated by the availability of the needed tools. MAPLE provides multiple-precision arithmetic, as well as heavy machinery to work with polynomials. In the last few years, it incorporated the libraries FGB and RS [73, 139] to perform multivariate and univariate roots isolation by means of Gröbner bases, rational univariate representations and Descartes' univariate root isolation, described in Chapter 2. Additionally, MAPLE offers a reliable visualization interface, which makes it an appealing choice as implementation platform.

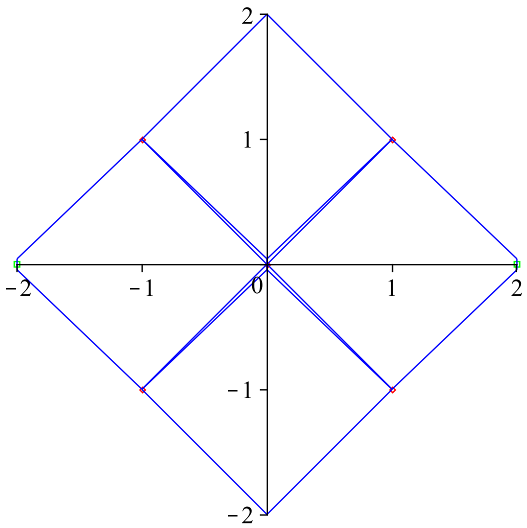
In more detail, the FGB library calculates Gröbner bases of a given multivariate polynomial system, using the state-of-the-art algorithm F_4 [74], and was developed in C by Jean-Charles Faugère. The RS library performs RUR calculations and Descartes' univariate real root isolation. It focuses on zero-dimensional systems (those with a finite number of complex solutions). The algorithms, described in [140] and [141], were developed in C by Fabrice Rouillier.

The development of FGB and RS libraries started more than fifteen years ago, and they are still under development. During their lifetime, they were thoroughly tested and debugged. Debugging such huge pieces of code is known to be a very difficult task. Moreover, another critical difficulty is that there is, to our knowledge, no method for verifying the result, that is verifying if the computed isolating boxes correspond *indeed* to the root of the considered system. During ISOTOP tests, we discovered that FGB/RS failed to solve correctly many bivariate systems. These errors became evident during the execution of the algorithm. Such inconsistencies exposed a bug in the system solving software, which bug resulted in the fact that some critical points were very slightly outside of the isolating boxes supposed to be in. This precise bug was caused by an error in the algorithm used to recover the solutions of the original system from the roots of the univariate polynomials of the RUR. This algorithm was later modified in RS and these modifications are now part of the version 14 of MAPLE. This shows how critical the consistency of intermediate results is in geometric algorithms. In general, geometric problems produce valuable challenges and test cases for solvers.

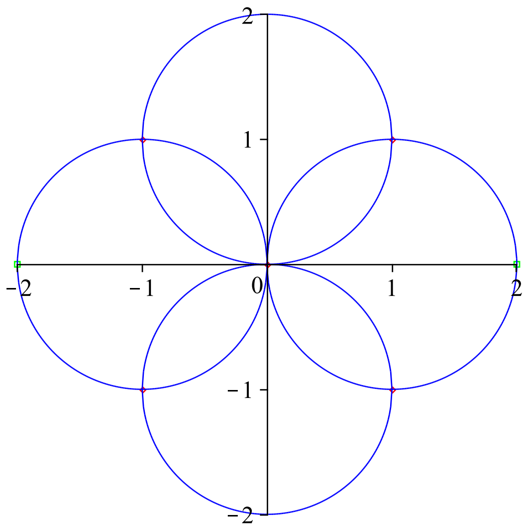
The implementation of ISOTOP follows the sequential steps of the algorithm given in Section 3.3.2. Of course, we also addressed many considerations related to efficiency issues and details not in the algorithm had to be addressed. For instance, as mentioned before, we experimentally found good values for precisions of isolations and refinements. Due to the bug discovered in RS, explained in the above paragraph, we also implemented temporarily our own version of the recovering of system solutions, based on interval arithmetic. We used these functions until the bug was corrected in RS.

Debugging had to be kept in mind during development, since ISOTOP is a reasonably large program (8000 lines of code). As MAPLE's programming language is untyped and we created data structures to deal with different entities of our algorithm (for instance, RUR of each system of equations, rectangles in the vertical planar decomposition of the plane or event points during the sweep), we used MAPLE's type-checking strategies to help in the debugging. We mined our code with assertions and warnings. We also wrote a set of debugging functions, aimed to print the content of data structures, import and export polynomial systems, and to plot portions of the rectangular plane subdivision.

As pointed out in Section 3.3, the output of the algorithm is a geometric graph. The implementation permits to consult and traverse this graph, for example with functions that enumerate or count connected components. But, as one of the direct applications of the algorithm is curve plotting and MAPLE provides a robust plot interface, we developed plotting functions. They convert the data structure representing the output geometric graph into a MAPLE plot structure, which permits to display the curve using MAPLE functions. The advantage of this is that we do not have to deal with graphic primitives such as displaying or zooming. In graphical mode, ISOTOP outputs, besides the geometrical graph, a drawing. By default, the graph is drawn, by plotting all its vertices and edges. But, in order to obtain a better visualization of the curve, ISOTOP offers the possibility of vertically subdividing the empty rectangles (those that do not contain critical points), calculating regular points on fibers in their interior. Letting the user choose in how many parts to split the empty rectangles, the plot of the curve is a very good approximation to it. Figure 4.1 shows the graph of a curve, plotted without



(a)



(b)

Figure 4.1: Graphs isotopic to the curve $x^8 + 4x^6y^2 + 6y^4x^4 + 4y^6x^2 + y^8 - 4x^6 - 12y^2x^4 - 12y^4x^2 - 4y^6 + 16y^2x^2 = 0$, plotted (a) without and (b) with vertical subdivision of empty rectangles (`nb_splits=1` and `nb_splits=100` respectively).

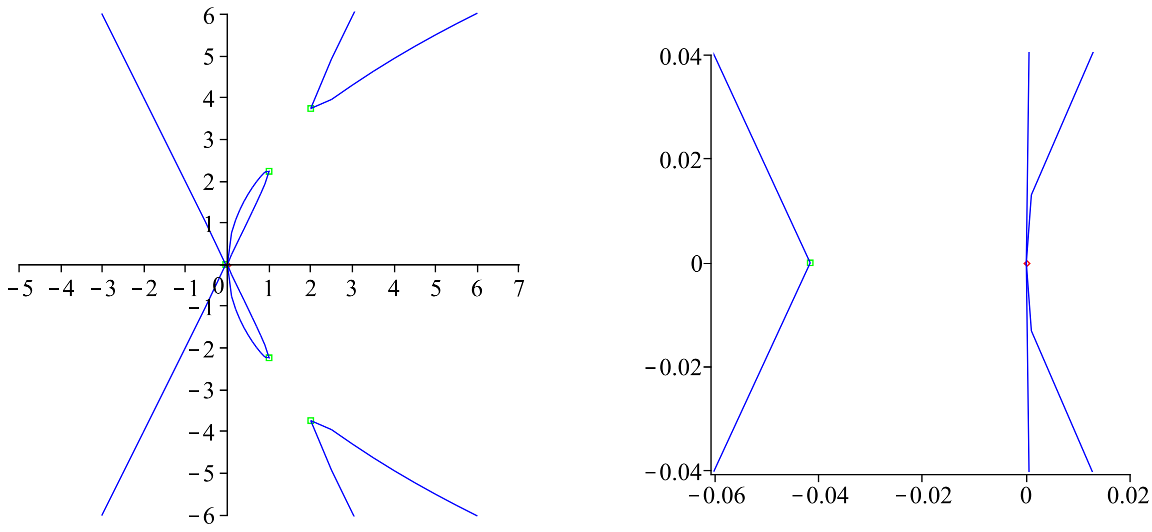
and with vertical splits. Note that the graph in Figure 4.1 (a) is correct, that is, isotopic to the input curve, although Figure 4.1 (b) gives a better approximation of the curve.

4.3 Program output

The curve used as example in Section 3.3.4 is introduced as input of the MAPLE worksheet. The example is ran on a MacBook Pro, Intel Core Duo, 2 GHz with 2Gb RAM. The printout is slightly modified for readability.

```
# The input curve.
f := y^4 - 6*y^2*x + x^2 - 4*y^2*x^2 + 24*x^3:


# Optional parameters fix the verbosity of the output, as well as the accuracy
# of the output graph: the precision for root isolations (10^{-precision}) and
# the number of additional vertical subdivisions of non-critical rectangles.
ISOTOP:-topology_real_curve(f,
```



(a) (b)

Figure 4.2: (a) Graph drawn with ISOTOP (with refinements) of the curve $y^4 - 6\,y^2\,x + x^2 - 4\,y^2\,x^2 + 24\,x^3$. (b) Detail of the graph near the origin.

```
verbosity=2,
precision=10,
plot_graph=true,
nb_splits=10);
```

The program produces the following output with the graph shown in Figure 4.2 (a). Figure 4.2 (b) shows the details of the graph near the origin (this zooming was obtained with MAPLE plot interface options). The singular point is marked by a (red) diamond, and the extreme points are marked by (green) squares.

```
1. Compute boxes:
Vertical asymptotes (0 found) and vertical lines (0 found) computed in
  0.000000 seconds
Extreme Grobner basis obtained in 0.018000 seconds
RUR calculated in 0.002000 seconds
Univariate isolation done in 0.008000 seconds
Singular Grobner basis obtained in 0.014000 seconds
```

```
RUR calculated in 0.003000 seconds
Univariate isolation done in 0.002000 seconds
Computed 1 singular and 5 x-extreme points in 0.048000 seconds
Boxes of critical points refined to avoid overlap in 0.001000 seconds
Compute singular points of multiplicity k in the fiber for k=2
Compute singular points of multiplicity k in the fiber for k=3
Compute singular points of multiplicity k in the fiber for k=4
Multiplicities of singular points in fibers computed in 0.048000 seconds
Multiplicities of singular points in fibers are: [4]
Boxes of extreme points refined for topology in 0.041000 seconds
Boxes of singular points refined for topology in 0.008000 seconds
Total time for computing the boxes of critical points (1 singular and
  5 extreme) = 0.149000 seconds (including 0.096000 seconds for Gb/RS)

2. Sweep:
Partitionned the plane into 49 rectangles in 0.185000 seconds
Elapsed total computation time: 0.334000 seconds

3. Construct graph: done in 0.002000 seconds
Total computation time: 0.336000 seconds
```

4.4 Experiments

We believe that comparing MAPLE and C/C++ implementations is fair for our problem when the running time is not too small because then, most of the time is usually spent on algebraic computations which are coded in C/C++ (possibly in the kernel of MAPLE). When the running time is too small, the MAPLE part of the code and, in our case, the interface to GB/RS is not negligible and comparing MAPLE and C/C++ implementations becomes meaningless. This is why we focused our tests on examples for which the running time exceeds 1 second. We measure the running time for computing the isotopic graph, but not the drawing. All the experiments were performed using 2.6 GHz single-core Pentium 4 with 1.5Gb of RAM and 512kb of cache, running 32-bit Debian GNU/Linux.

We compared our code, ISOTOP, with two C++ implementations, CA [61] (formerly known as ALCIX) and CAD2D [29, 27] and two MAPLE implementations, TOP [86] and INSULATE [147]. Another promising software is AXEL [3, 11], but no implementation of the certified subdivision algorithm is currently available.

CA is a C++ code, part of the CGAL library [38], since version 3.7. Following the recommendation of Michael Kerber, the CA author, we ran two versions of the code with the flag `CGAL_ACK_RESULTANT_FIRST_STRATEGY` set to 1 and 0. One being optimized for generic cases, while the other is optimized for singular curves. We always compare to the better running time. CAD2D is a stand-alone C++ code which can also be compiled in combination with the SINGULAR library [88] (used for polynomial factorization). In our tests, CAD2D appears to be much more efficient when ran with SINGULAR [53] (and we report these tests). Finally, recall that TOP requires an initial precision, which we set to 50.

As discussed in Section 3.1, the various implementations do not compute exactly the same thing and comparisons should thus be taken with care. Recall that when the curve is not in generic position, TOP and INSULATE shear the curve to perform calculations, but they do not shear back the results, thus not computing the critical points (and, in particular, the x-extreme points) in the original coordinate system. ISOTOP, CA and CAD2D always output the critical points in the original coordinate system.

We ran large scale benchmarks on over a thousand of curves during several weeks. In particular, we considered curves suggested in [109, 29, 86] and several classes of non-generic curves. We considered about 1300 curves from [109], which are classified in 18 challenges covering a large variety of interesting cases such as isolated points, high multiplicity of tangency at singularities, large number of branches at singularities or many singularities. This set contains curves of degree up to 90 that are both in generic and non-generic position. As suggested in [29], typical curves in generic position can be generated (i) as a random bivariate polynomial (which usually do not have singular points) or (ii) as resultants of two random trivariate polynomials (which usually have singular points, including isolated points). In both cases, we considered random polynomials with 50% non-zero coefficients of bitsize 32 in Case (i) and initial bitsize 8 in Case (ii). We generated such curves with degrees up to 25. We also generated classes of curves in non-generic position in two different ways. First, we considered products of a curve with one or several of its vertical translates. Second, we considered curves of the type $g = f^2(x, y) + f^2(x, -y)$; such curves are usually irreducible and consist of isolated points which are the intersections of the curve $\mathcal{C}_f$ with its symmetric with respect to the x-axis. We generated such curves with degrees up to 24.

We set in our experiments a limit of 30 minutes for the computation of the topology of one curve. We report as *time out* instances that exceed this running time. Also, CAD2D which uses Singular for modular arithmetic, often reports on difficult instances that the table of primes has been exhausted, which results in an interruption of computation; this is reported in the tables as *aborted*. This happens because it was designed to work in 16-bit environments. There are roughly six thousand prime numbers that fit in a 16-bit register, it is reasonable to think that all of them can be hardcoded in order to avoid implementing efficient primality testing functions. Thus, when intermediate results need more prime numbers to perform modular computations, the program stops. Thus not the subject of this discussion, some considerations on efficient primality testing are given in Section 5.4.4.

In summary, we ran our benchmarks on a total of 1500 curves. As mentioned above, it is not significant to compare C++ and MAPLE implementations when the running time is too small. We thus only report experiments on 650 curves whose running times exceeded 1 second for ISOTOP. The distribution of degrees and number of critical points of these 650 curves is shown in Figure 4.3.

Figure 4.4 shows the ratio of running times between each of the competing implementations and ISOTOP over our set of 650 curves. It appears difficult to analyze the benchmarks globally because there are always particular examples that are processed faster by a given implementation. We note, however, that INSULATE is almost always

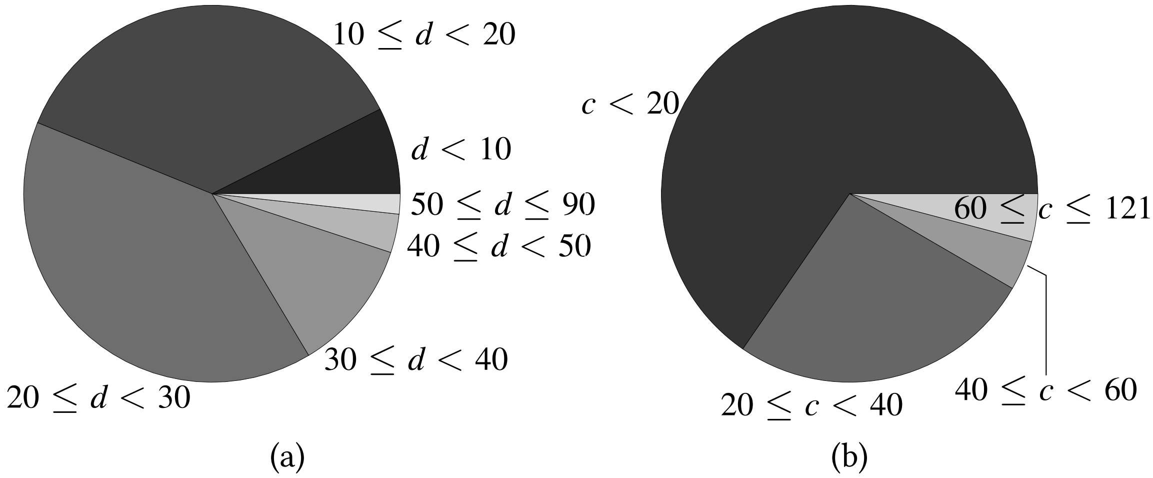


Figure 4.3: Distributions of (a) degrees and (b) number of critical points in our 650 examples.

slower than ISOTOP, except for random curves with no singular points. In addition, INSULATE and TOP reached the time limit on more than half of the examples and, in particular, on difficult examples. We can, nevertheless, comment on the general behavior of the different approaches depending on the classes of examples.

To illustrate the behavior on curves in generic positions, we report the running times for random curves in Table 4.1 and for resultants of surfaces in Table 4.2. Random curves have no singular points and few extreme points. In this case, we observe that ISOTOP is the least efficient implementation. This can be explained by the fact that ISOTOP computes the Gröbner basis of a large system without multiplicities, which is the worst case in practice. On the other hand, the other implementations benefit from interval arithmetic filters in the lifting phase, which speed up computations by avoiding expensive symbolic computations, see for example [29]. Generic curves generated as resultants have many singularities and extreme points. ISOTOP benefits from splitting the critical system in two smaller (singular and extreme) systems and hence it performs relatively better than in the completely random case. We observe that ISOTOP is typically a bit slower than CA but faster than TOP, and that CAD2D aborts.

To illustrate the behavior on curves in non-generic position, we consider different classes of curves. The first class of non-generic curves are constructed with one curve

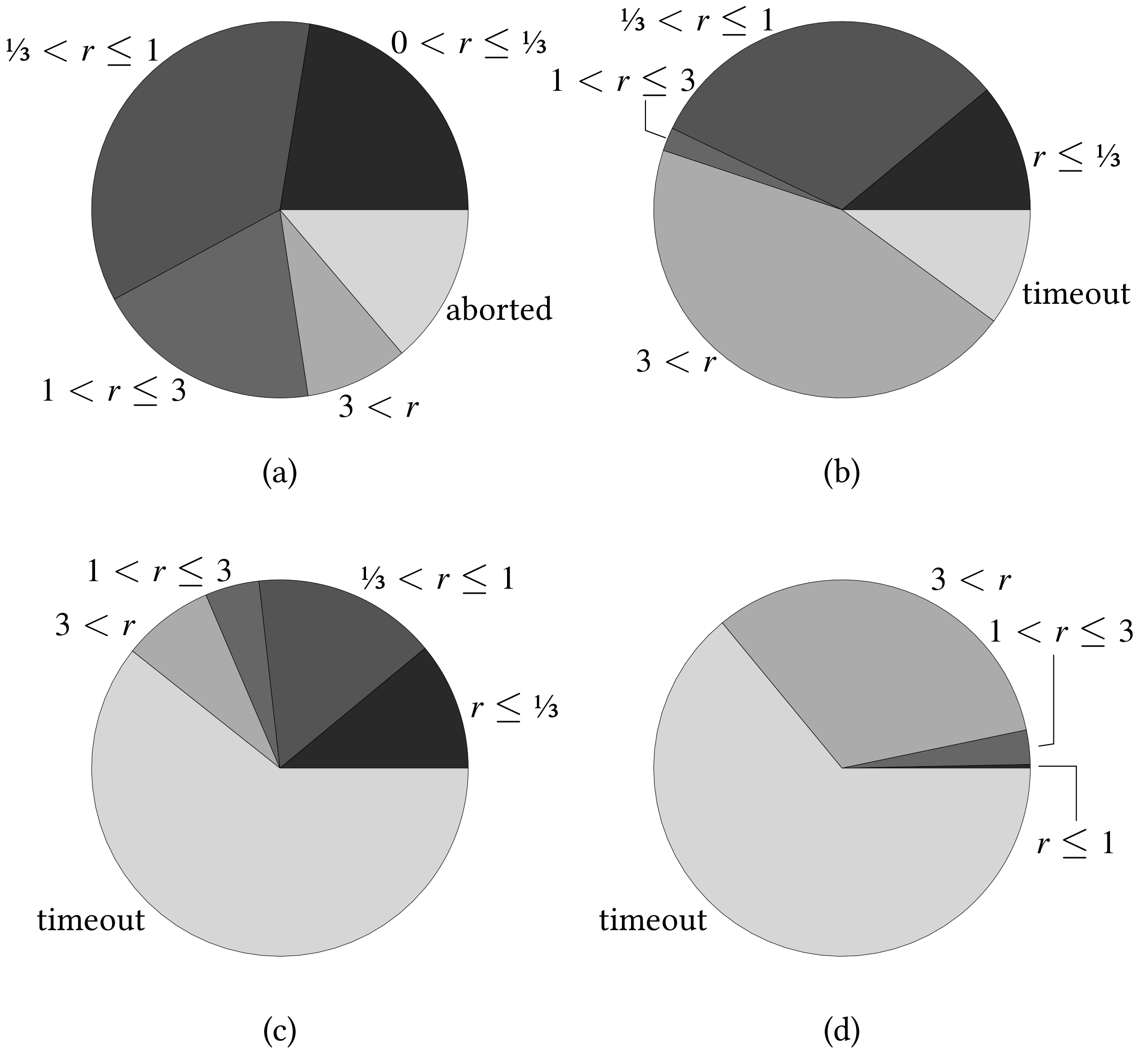


Figure 4.4: Distributions of running time ratios for (a) CAD2D, (b) CA, (c) TOP, and (d) INSULATE over ISOTOP. r gets larger (in other words, the picture gets lighter) as ISOTOP performs better than the competitive algorithms. Timeout means that the limit of 30 minutes was reached for the competitive algorithms.

multiplied by one or several of its vertical translates. The initial curve is taken either randomly, in Table 4.3, or it is a resultant of two surfaces, in Table 4.4. Table 4.5 reports results on the second class of non-generic curves of the type $f^2(x, y) + f^2(x, -y)$ for random polynomials f. For these non-generic curves, ISOTOP is typically faster than other implementations.

As a general rule, we observe that, except for random curves, that is, curves in generic position and without singular point, the ratio of the running times between other implementations and ISOTOP is increasing with the degree of the curve. Kerber's thesis [104] also presents tests comparing CA, CAD2D and ISOTOP. Results of these tests also show that ISOTOP behaves badly with random generic curves. In other words, except for random curves, ISOTOP tends to perform better, compared to others, when the degree increases. In particular, this implies that ISOTOP can push forward the complexity of the curves one can study.

4.5 Conclusion and perspectives

This chapter presented ISOTOP, a MAPLE implementation of the algorithm of Chapter 3. We described its interface and design considerations, and we presented and discussed experiments and comparisons with other similar implementations. As usual in most implementations, many aspects can be reworked to obtain better performance.

The experiments presented in Section 4.4 also served as profiling. The detailed results of these benchmarks are shown in Appendix A. They present the time spent on each step of the algorithm, as well as some information about the characteristics of each curve. We mainly used this information to understand why ISOTOP is slower than others in some cases. For the case of random curves, the computation of the Gröbner bases of the resulting systems is very slow, while the other steps are performed very quickly. For the case of curves with high tangency at isolated singularities, the most time-demanding step is the computation of the RURs. One possible approach to improve the efficiency of our algorithm in the latter case is to compute a radical decomposition of the original systems. Another approach that would improve the efficiency in random curves is to detect such situations and to avoid solving these systems using Gröbner bases in these cases. These observations led to the development of a, yet experimental, version of a bivariate solver. In preliminary tests, this solver showed to be capable of solving these complicated systems hundreds of times faster. We expect that this approach will make ISOTOP the fastest implementation in all cases, generic and non-generic.

The curve visualization in MAPLE is very fast but it is not certified in the case where the curve has very close critical points. The routines also fail to distinguish branches of the curve that are close but non-intersecting from those that do intersect. We believe that ISOTOP is a perfect candidate to replace the current implementation of algebraic curve visualization.

It should be stressed that we focused here on computing graphs isotopic to *one* algebraic curve. When we are given a set of curves, or said in other words, one curve whose equation is factorized or partially factorized, it is more efficient to compute the topology of every curve and then to combine them. This approach has been studied in detail by Eigenwillig and Kerber [60, 104] and led to a very efficient implementation [66]. Combining such approaches would be of substantial interest.

We conclude this chapter by mentioning that ISOTOP code is available in our webpage, `http://vegas.loria.fr/isotop`.

d	τ	ISOTOP	CA	CAD2D	TOP	INSULATE	CA/ISOTOP	CAD2D/ISOTOP	TOP/ISOTOP	INSULATE/ISOTOP
10	32	3.8	0.42	1.6	1.4	4.3	0.1	0.4	0.4	1.1
12	32	9.7	2.2	2.3	4.8	8.1	0.2	0.2	0.5	0.8
14	32	22	3.7	4.3	15	26	0.2	0.2	0.7	1.2
16	32	72	4.5	6.1	38	39	0.1	0.1	0.5	0.5
18	32	160	16	12	94	98	0.1	0.1	0.6	0.6
20	32	320	31	25	210	120	0.1	0.1	0.7	0.4

Table 4.1: Running times in seconds (averaged over 5 runs) for random bivariate polynomials with 50% non-zero coefficients of bitsize 32.

d	τ	ISOTOP	CA	CAD2D	TOP	INSULATE	CA/ISOTOP	CAD2D/ISOTOP	TOP/ISOTOP	INSULATE/ISOTOP
16	64	29	14	aborted	52	>600	0.5	-	1.8	>20
25	80	590	410	aborted	>1800	>1800	0.7	-	>4.3	>3

Table 4.2: Running times (averaged over five runs) in seconds for resultants of two random trivariate polynomials, both of total degree 4 or 5, with 50% non-zero coefficients of bitsize 8.

d	τ	ISOTOP	CA	CAD2D	top	insulate	CA/ISOTOP	CAD2D/ISOTOP	TOP/ISOTOP	INSULATE/ISOTOP
12	96	4.1	6.5	1.6	9 or >600	>600	1.6	0.4	2.2 or >92	>140
15	96	15	38	10	40 or >600	>600	2.5	0.7	2.7 or >16	>40
18	96	49	120	66	>600	>600	2.4	1.3	> 12	>12
21	96	140	510	510	>1200	>1200	3.5	3.6	> 8.6	>9

Table 4.3: Running times (averaged over five runs) in seconds for non-generic curves generated by the product of a curve f with two of its translates $f(x, y+1)$ and $f(x, y+2)$. The curve f is chosen randomly with degree between 4 and 7 and 50% non-zero coefficients of bitsize 32, resulting in curves of bitsize τ. The running-time discrepancy of TOP is large and is not averaged.

d_1	d_2	d	τ	ISOTOP	CA	CAD2D	TOP	INSULATE	CA / ISOTOP	CAD2D / ISOTOP	TOP / ISOTOP	INSULATE / ISOTOP
3	3	18	96	26	93	160	>600	>600	3.6	6.3	>20	>20
3	4	24	112	250	510 or >1800	aborted	>1800	>1800	2 or >7	-	>20	>20

Table 4.4: Running times (averaged over five runs) in seconds for non-generic curves generated by the product of a curve f with its translate $f(x, y+1)$. The curve f is the resultant of two random trivariate polynomials of total degree d_1 and d_2 and 50% non-zero coefficients of bitsize 8. The running-time discrepancy of CA is large for degree 24 and is not averaged.

d	τ	ISOTOP	CA	CAD2D	TOP	INSULATE	CA / ISOTOP	CAD2D / ISOTOP	TOP / ISOTOP	INSULATE / ISOTOP
10	64	39	180	25	>600	>600	4.5	0.6	>15	>15
12	64	240	300	aborted	>1800	>1800	1.2	-	>7	>7
14	64	350	>1800	aborted	>1800	>1800	> 5	-	>5	>5

Table 4.5: Running times (averaged over five runs) in seconds for polynomials of the form $g = f^2(x,y) + f^2(x,-y)$. The random polynomials f have bitsize 32, term density 50% and degrees varying between 5 and 7.

Chapter 5

CGAL Univariate Algebraic Kernel

More seriously, even perfect program verification can only establish that a program meets its specification. The hardest part of the software task is arriving at a complete and consistent specification, and much of the essence of building a program is in fact the debugging of the specification.

Frederick P. Brooks, Jr. [26]

The program presented in the previous chapter, ISOTOP, was implemented on top of a computer algebra system. From the programmer's point of view, this has enormous advantages, since MAPLE provides a complete framework for the implementation of algebraic algorithms. However, MAPLE programs cannot be used as library in some other program, as can be easily done with C or C++ implementations. Nevertheless, before implementing a program analogous to ISOTOP in C or C++, there must be an algebraic framework providing, in one hand, geometric operations and, on the other hand, real solving and handling of algebraic numbers. Such framework would permit the implementation of ISOTOP, but also most of the geometric algorithms that deal with non-linear objects.

The open-source C++ Computational Geometry Algorithms Library, CGAL [38], is an open-source project which became a standard platform for the implementation of geometric algorithms, see Section 5.1. This library follows strict design patterns, and employs cutting-edge features of C++. Since CGAL is a standard in the computational geometry community, our aim is to make this library capable of dealing with algebraic objects, such as polynomials of arbitrary degree and algebraic numbers.

The initiative of equipping CGAL with the tools needed to handle curved objects in most geometric algorithms started around ten years ago. Before, the library was able to handle curved objects only in particular algorithms. For instance, it was able to compute arrangements of conics, but this functionality was implemented inside the CGAL arrangement package and other CGAL algorithms were not able to handle conics. The aim was to create *kernels* that handle curved objects. In the CGAL terminology, a kernel encapsulates the geometric objects and arithmetic operations, and algorithms are parameterized with kernels (see Section 5.1 for details).

In a first step, the construction of a kernel capable of handling circular arcs was considered [134]. Later, a kernel handling spheres was developed [52]. These kernels handle specific curved objects, but they are not able to handle curves defined by polynomials of arbitrary degree. This is more involved, because handling such polynomials in the context of geometric algorithms essentially requires to be able to compute the real roots of univariate polynomials and systems of polynomials, and to handle (*e.g.*, compare) the corresponding algebraic numbers. Moreover, efficiency is of course critical. Efforts to develop a kernel capable of handling curved objects defined by arbitrary degree curves started in the beginning of the last decade [69]. In the last years, the specification of such a kernel, named *algebraic kernel* were discussed [19] and some experimental algebraic kernels were developed [18, 68]. These specifications reached a mature state in 2009, and were incorporated to the version 3.6 of CGAL in 2010.

This chapter presents an implementation of a CGAL *univariate algebraic kernel*, that follows the CGAL specifications. It is capable of handling univariate polynomials of arbitrary degree with integer coefficients, by providing functions for root isolation, provided by the library RS described in Section 4.2, and functions for comparing algebraic numbers and refining approximations of such numbers. The chapter is organized as follows. Section 5.1 describes the CGAL library. Section 5.2 explain the concept of CGAL algebraic kernel. Some work related to our implementation is discussed in Section 5.3. The implementation itself is presented in Section 5.4. We compare in Section 5.5 our kernel with other comparable kernels and demonstrate the efficiency of our approach. We perform experiments on large data sets including polynomials of high degree (up to 2 000) and with very large coefficients (up to 25 000 bits per coefficient). Finally, in Section 5.6, we apply our kernel to the problem of computing arrangements of x-monotone polynomial curves and demonstrate its efficiency compared to previous solutions available in CGAL.

The work presented in this chapter was published in 2009 at the annual *Symposium on Experimental Algorithms* [112]. We also distribute corresponding implementations with the CGAL library. First, our algebraic kernel itself is now part of CGAL [20, § 8.2.1, § 8.3]. We also contributed two new number types, `Gmpfr` and `Gmpfi`, to the CGAL library [91, § 5.4, § 5.7, § 5.9]; these new number types interface the two multiple-precision libraries MPFR and MPFI [129, 128]. We originally designed these number types for our algebraic kernel, but we later integrated them into the CGAL number types, which now make them available to all CGAL users.

5.1 The Computational Geometry Algorithms Library

CGAL [38] is designed in a modular fashion following the paradigm of *generic programming* [5, 10, 79]. It relies on the C++ Standard Template Library [150] and on the BOOST libraries [23]. Algorithms are typically parameterized by a *traits class* which encapsulates the geometric objects, predicates and constructions used by the algorithm. Algorithms can thus typically be implemented independently of the type of input objects. For instance, the core of a line-sweep algorithm for computing arrangements of plane curves [51] can be implemented independently of whether the curves are lines, line segments, or general curves; on the other hand, the elementary operations that depend on the type of the geometric objects (such as, comparing x-coordinates of points of intersection), which depend on the type of the objects, are implemented separately in traits classes. Similarly, the model of computation, such as exact arbitrary-length integer arithmetic or approximate fixed-precision floating-point arithmetic, are encapsulated in the concept of *kernel*. An implementation is thus typically separated in three or four layers, (i) the geometric algorithm which relies on (ii) a traits class, which itself relies on (iii) a kernel for elementary (typically geometric) operations. CGAL provides several predefined Cartesian kernels, for instance allowing standard Cartesian geometric operations on inputs defined with doubles and providing approximate constructions (*i.e.*, defined with double) but exact predicates. However, a kernel can also rely on (iv) a number type which essentially encapsulates the type of number (such as, double, arbitrary-length integers, intervals) and the associated arithmetic operations. A choice of traits classes, kernels and number types is useful as it gives freedom to the users and it makes it easier to compare and improve the various building blocks of an implementation.

5.2 Algebraic kernel

CGAL has algorithms specialized in handling curves of low fixed degrees. In the last years, the CGAL community put efforts towards handling arbitrary degree curves. The concept of the *algebraic kernel*, which includes, in particular, its specification introduced in version 3.6 of the library in 2010 [20]. Nevertheless, the specifications evolved during a long time before reaching a mature state, generating long discussions among different groups involved in the development of the specifications, notably between the CGAL Editorial Board, the Max Planck Institut für Informatik (MPII), INRIA Sophia-Antipolis, the National and Kapodistrian University of Athens (NKUA) and us. The notion of algebraic kernel for CGAL was first proposed in 2004 [69].

Formally, an algebraic kernel is defined as a set of functions aimed to handle and solve polynomials and polynomial systems, and handling the algebraic numbers resulting from these root isolations. The current CGAL specifications [20] are split in univariate and bivariate kernels though, in the future, multivariate kernels may be specified and developed.

The CGAL-compliant univariate algebraic kernel presented in this chapter provides real-root isolation of univariate integer polynomials and basic operations, *e.g.* comparisons and sign evaluations, of real algebraic numbers. This open-source kernel follows the CGAL specifications for algebraic kernels, and it was the first algebraic kernel shipped with CGAL, in 2010. The root isolation is based on the interval Descartes algorithm [44] and uses the library RS [139]. Moreover, our kernel provides various operations for polynomials, such as greatest common divisor (gcd), which are crucial for manipulating algebraic numbers. This modular gcd, in particular, was implemented at early stages of the project, since CGAL did not provide at that moment efficient functions on polynomials. The development of this CGAL algebraic kernel started in 2006.

5.3 Related work

Combining algebra and geometry for manipulating non-linear objects has been a longstanding challenge. Previous implementation include, but are not limited to, MAPC [106] a library for manipulating points that are defined algebraically and handling curves in the plane. More recently, the library EXACUS [17], which handles curves and surfaces in computational geometry and supports various algebraic operations, was developed and partially integrated into CGAL. In [69], the underlying algebraic operations were based on the SYNAPS library [125].

One CGAL algebraic kernel was developed at MPII [93] following the generic programming paradigm using the C++ template mechanism. This kernel is templated by the representation of algebraic numbers and by the real root isolation method, for which two classes have been developed; one is based on the Descartes method and the other on the Bitstream Descartes method [62]. This approach has the advantage to allow, in principle, using the best instances for both template arguments.

A related software was developed at INRIA and relies on the SYNAPS library [125].[6] In this program there are several approaches concerning real root isolation, *i.e.*, methods based on Sturm subdivisions, sleeves approximations, continued fractions, and a symbolic-numeric combination of the sleeve and continued fractions methods (see [94]). Moreover, there are specialized methods for polynomials of degree less or equal than four [156]. On the other hand, this implementation is not formally a CGAL algebraic kernel, since it does not implement comparison of algebraic numbers, as required in the CGAL algebraic kernel specification. Nevertheless, for uniformity on the nomenclature of the compared software, it will be referred to as SYNAPS kernel throughout this chapter.

Emiris et al. [94] presented some benchmarks of these various approaches in these two kernels as well as some tests on the kernel we present here. The authors mention that our kernel based on interval Descartes performs similarly to one approach (refer to as NCF2) based on continued fractions [155] for coefficients with (very) large bitsize but NCF2 is more efficient for small bitsize. They conclude that, first, dedicated algorithms for polynomials of degree less than (or equal to) four is always the most efficient approach and, second, that NCF2 always perform the best except for low-degree and high-bitsize polynomials, in which case the kernel based on the Bitstream Descartes method performs the best. We moderate here these conclusions.

5.4 Implementation

We describe in this section the implementation of our univariate algebraic kernel. The two main requirements of the CGAL specifications, which we describe here, are the isolation of real roots and their comparison. We also describe our implementation of two operations, the gcd computation and the refinement of isolating intervals, that are both

[6] SYNAPS is currently integrated as the package REALROOT in MATHEMAGIX [159], which is an open source computer algebra system that combines symbolic and numeric computations.

needed for comparing algebraic numbers.

5.4.1 Representation of numbers and polynomials

Our algebraic kernel handles univariate polynomials and algebraic numbers. The polynomials have integer coefficients and are internally represented by arrays of GMP arbitrary-length integers [83]. We implemented in the kernel the basic functions for polynomials, including basic arithmetic, evaluation, and input/output. During the development phase of our kernel, experimental versions of CGAL incorporated functionalities on polynomials with coefficients of arbitrary type. These functionalities were incorporated to CGAL in version 3.6 [90]. Because of that, we modified our kernel to now accept CGAL polynomials and our implementation of polynomial arithmetic is now only used internally.

An algebraic number that is a root of a polynomial F is represented as a data structure containing F and an isolating interval, that is an interval containing this root but no other root of F. We implemented intervals using the MPFI library [128], which represents intervals with two MPFR arbitrary-fixed-precision floating-point numbers [129]; note that MPFR is developed on top of the GMP library for multiple-precision arithmetic [83].

CGAL did not have interfaces to efficient state-of-the-art floating-point arithmetic libraries such as MPFR. The choices were CORE and LEDA; the first one not efficient for our intended use, and the second one is not free. We chose to implement interfaces to MPFR and MPFI libraries. In the CGAL terminology, this is stated as creating two *number types*. Each number type must fulfill some requirements, in particular, it must specify the properties of the algebraic structures it represents [89]. We implemented the new number types `Gmpfr` and `Gmpfi` [91], based respectively on the libraries MPFR and MPFI. Besides their fulfillment of CGAL requirements, they are very efficient and they were incorporated into the version 3.6 of CGAL in 2010 [91], not only for the algebraic kernel internal use but for all CGAL users.

5.4.2 Root isolation

For isolating the real roots of univariate polynomials with integer coefficients, we developed an interface with the library RS [139]. This library is written in C and is based on Descartes' rule (see Section 2.3.3 for details) for isolating the real roots of univariate polynomials with integer coefficients. In particular, we use the C interface of RS, which is exactly the same interface that is linked to MAPLE (see Section 4.2 for details).

For ease of reading, we recall here the general design of the RS library; see [141] for details. Rs is based on an algorithm known as *interval Descartes* [44]; namely, the coefficients of the polynomials obtained by changes of variable, sending intervals $[a, b]$ onto $[0, +\infty]$, are only approximated using interval arithmetic when this is sufficient for determining their signs. Note that the order in which these transformations are performed in RS is important for memory consumption. The intervals and operations on them are handled by the MPFI library. Another characteristic of RS is its memory management: it implements a *mark-and-sweep* garbage collector, which is well suited to RS needs.

Since the applications in computational geometry are not normally a priority when designing computer algebra software, the development of the algebraic kernel was also a challenge for the RS author. That is, RS benefited from the data produced by our benchmarks and from bug reports. On the other hand, we benefited from the fact that RS was optimized for some cases. In particular, RS was originally designed to work with polynomials of high degree and bitsize. Our tests showed that the performance of RS with small polynomials was not optimal, and some internal parameters of the library were changed according to our needs.

5.4.3 Algebraic number comparison

One of the main requirements of the CGAL algebraic kernel specifications is to compare two algebraic numbers r_1 and r_2. If we are lucky, their isolating intervals do not overlap and the comparison is straightforward. This is, of course, not always the case. If we knew that the two algebraic numbers were not equal, we could refine both isolating intervals until they are disjoint; see below for details on how we perform the refinements. Hence, the problem reduces to determining whether the algebraic numbers are equal or not.

To do so, we compute the square-free factorization of the greatest common divisor of the polynomials P_1 and P_2 associated to the algebraic numbers (see below for details). The roots of this gcd are the common roots of both polynomials. We calculate the intersection, I, of the isolating intervals of r_1 and r_2. The gcd has a root in this interval if and only if $r_1 = r_2$.

To determine whether the gcd has a root in interval I, it suffices to check the sign of the gcd at the endpoints of I: if they are different or one of them is zero, the gcd has a root in I and $r_1 = r_2$; otherwise, $r_1 \neq r_2$ and we can refine both intervals until they are disjoint.

5.4.4 Gcd computations

Computing greatest common divisors between two polynomials is not a difficult task, however, it is not trivial to do so efficiently. Indeed, a naive implementation of the Euclidean algorithm works fine for small polynomials but the intermediate coefficients suffer an exponential growth in size, which is not manageable for medium to large size polynomials. One common approach to this issue is to use *modular* gcd computation. This technique consists in calculating the gcd of polynomial modulo some prime numbers and reconstruct later the result with the help of the *Chinese remainder theorem*. Details on these algorithms can be found, for example, in [166]. Note that modular gcd is always more efficient than regular gcd and it is much more efficient when the two polynomials have no common roots.

The specification of the algebraic kernel does not require gcd functions, because they are currently provided by the polynomials package. Some benchmarks on our algebraic kernel showed that a naive gcd implementation was the bottleneck of our implementation. Nevertheless, efficient gcd computations were only recently introduced in CGAL [90, 92]. We thus implemented a *modular* gcd function; we did not use some existing implementations mainly for efficiency because converting polynomials from one representation to another is substantially costly as soon as the degree and bitsize are large. Our gcd function uses computers unsigned integers as prime type. We opted to avoid having a list of prime numbers for some reasons. First, there are roughly 2×10^8 prime numbers of 32-bits and 4×10^{17} prime numbers of 64-bits. This amount of numbers cannot be hardcoded in a list, and storing a subset of this list will reduce the size of the polynomials that the algorithm can handle. The considered approach consists in perform the Miller-Rabin primality test [137], using the Jaeschke variation [98] which states that a small set of witnesses suffice to test primality numbers up to a certain limit. Our modular gcd function is slower than the more recent CGAL's implementation for small (degree and bitsize) polynomials. The reason is that CGAL's implementation uses a table of prime numbers, which avoids calling primality testing functions for small numbers. Nevertheless, we do not use the CGAL gcd function in our algebraic kernel, since a conversion of data structures is needed. This conversion introduces an important overhead in computations we want to avoid.

A third implementation of a modular gcd function is provided by RS. This function is extremely fast for polynomials of big degree and bitsize, as RS was conceived to work with that kind of polynomials. For technical reasons, this function remained internal to RS for a long time (previous versions of it worked with the internal data structures representing polynomials). Currently, the RS gcd function is only used internally by RS and not by our algebraic kernel. The reason is that our gcd function performs very well with small polynomials and reasonably well with big polynomials, while the RS gcd function performs extremely well with very big polynomials, but for small polynomials it adds a constant computation time that we want to avoid.

The primary goal of RS is the resolution of algebraic systems and thus the isolation of the roots of univariate polynomials defined by the RUR of the systems. Such polynomials usually have (very) big bitsizes and degrees. This also explains the need of implementing functions that can handle polynomials of very large bitsizes and degrees.

5.4.5 Refining isolating intervals

As we mentioned in Section 5.4.3, refining the interval representing an algebraic number is critical for comparing such numbers. We provide two approaches for refinement.

Both approaches require that the polynomial associated to the algebraic number is square free. The first step thus consists in computing the square-free part of the polynomial. This is easily done by computing the gcd of the polynomial and its derivative.

Our first approach is a simple bisection algorithm. It consists in calculating the sign of the polynomial associated to the algebraic number at the endpoints and midpoint of the interval. Depending on these signs, we refine the isolating interval to its left of right half.

Our second approach is a quadratic interval refinement [1]. Roughly speaking, this method splits the interval in many parts and, based on a linear interpolation, guesses in which one the root lies. If the guess is correct, the algorithm divides in the next refinement step the interval in more parts and, if not, in less.

Unfortunately, even with our careful implementation this approach turns out to be, on average, only just a bit faster than the bisection approach. Our experiments showed that the bottleneck of the refinement is the evaluation of polynomials. Fine-tuning the evaluation considerably improved the two refinement functions.

It should be noted that RS also provides an experimental refinement function for algebraic numbers. At the cost of fulfilling a preliminary condition (Kantorovich criterion [101]), it provides a method in which the number of refined bits grows quadratically in each iteration. This function was not fully tested because its implementation is currently still experimental.

When asked to refine algebraic numbers, our algebraic kernel only uses the bisection method, since it is the more stable implementation and performs reasonably well. We plan, in the future, to adopt the RS refinement function, since preliminary tests showed very good performances.

5.5 Kernel benchmarks

In this section, we analyze the running time of the two main functions of our algebraic kernel, that (i) isolate the roots of a polynomial and (ii) compare two algebraic numbers that is, compare the roots of two polynomials. We also compare the performance of our kernel with the one based on the Bitstream Descartes method [62] and developed at the Max-Planck-Institut für Informatik [93] (referred to as MPII's kernel)[7] and with a kernel based on continued fractions [148, 155] and developed at INRIA on top of the SYNAPS library [125] (which we refer to as SYNAPS' kernel).

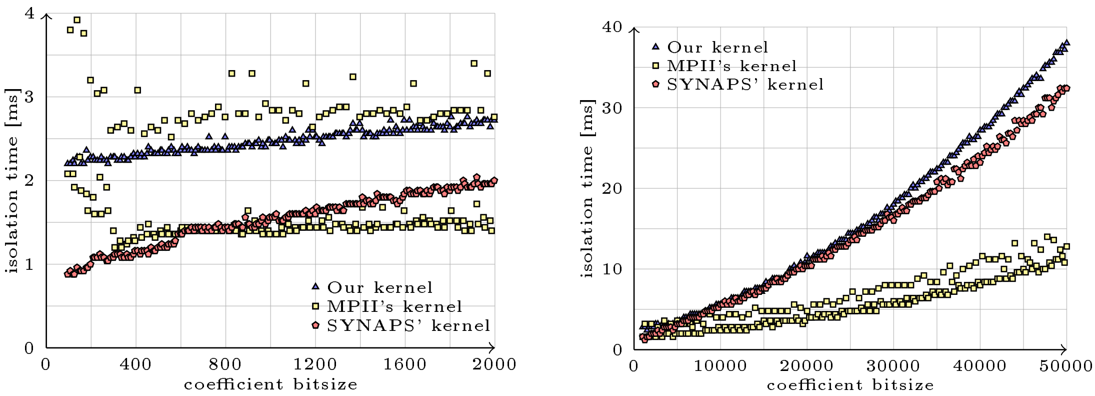
CGAL algebraic kernels had also been previously tested. [93] presents tests on the MPII's kernel, but without comparing it to other algebraic kernels. We tried to reproduce their tests on isolations in next section. [68] presents tests comparing SYNAPS' kernel using various algorithms to isolate roots, MPII's kernel using two different implementations of Descartes' algorithm to isolate roots and our kernel. They test diverse sets of polynomials, but all of them with degrees and bitsizes smaller than those considered here. We believe that considering higher degrees and bitsizes is important in geometric applications. For instance, the rotation of an object may drastically increase the bitsize of the coefficients of its equation. Moreover, the RUR of polynomial systems usually involve univariate polynomials of high degree and bitsize, despite the size of the original polynomials of the system.

All tests presented on this chapter were ran on a single-core 3.2 GHz Intel Pentium 4 with 2 Gb of RAM and 2048 kb of cache memory, using 64-bit Debian GNU/Linux.

5.5.1 Root isolation benchmarks

We consider two suites of experiments in which we either fix the degree of the polynomials and vary the bitsize of the coefficients or the converse; see Figures 5.1 and 5.2. In

[7] To compare both algebraic kernels with the same inputs, we parameterized MPII's kernel to use Bitstream Descartes as root isolator, `algebraic_real_bfi_rep` as algebraic number representation and CORE integers and rationals to represent the coefficients of the polynomials and the isolation bounds of algebraic numbers, respectively. The choice of CORE (vs. LEDA) was induced by the need of testing the kernels in the same conditions, that is, relying on GMP. The pure GMP support in MPII's kernel is still in experimental stage.



(a) small bitsizes (b) medium to high bitsizes

Figure 5.1: Running time for isolating all the real roots of degree 12 polynomials with 12 real roots in terms of the maximum bitsize of their coefficients.

each experiment, we report the running time for isolating all the roots per polynomial, averaged over different trials, for our kernel, MPII's and SYNAPS' kernel. Due to a current bug in the 64-bit version of CORE, MPII's kernel fails to find all the roots of a polynomial in some cases. This explains, in particular, the erratic results of the experiments in Figure 5.2 for MPII's kernel. This is discussed in detail at the end of the chapter.

Varying bitsize

We study here polynomials with rather low degree (12) but with no complex root and polynomials with reasonably large degree (100) with random coefficients (and thus with few real roots).

The first test sets comes from [93]. See Figure 5.1. It consists of polynomials of degree 12, each one being the product of six degree-two polynomials with two roots, at least one of them in the interval $[0, 1]$; every polynomial thus has 12 real roots. We vary the maximum bitsize of all the coefficients of the input polynomial from 100 to 50 000 and average each test over 250 trials.

Secondly, we consider random polynomials with constant degree 100 and coefficients with varying bitsize. See Figure 5.2. Note that such random polynomials have few roots: the expected number of real roots of a polynomial of degree d with coefficients in-

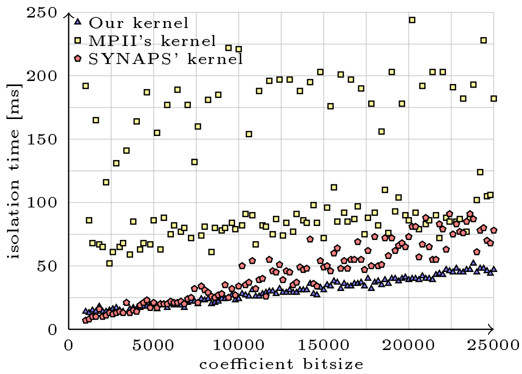
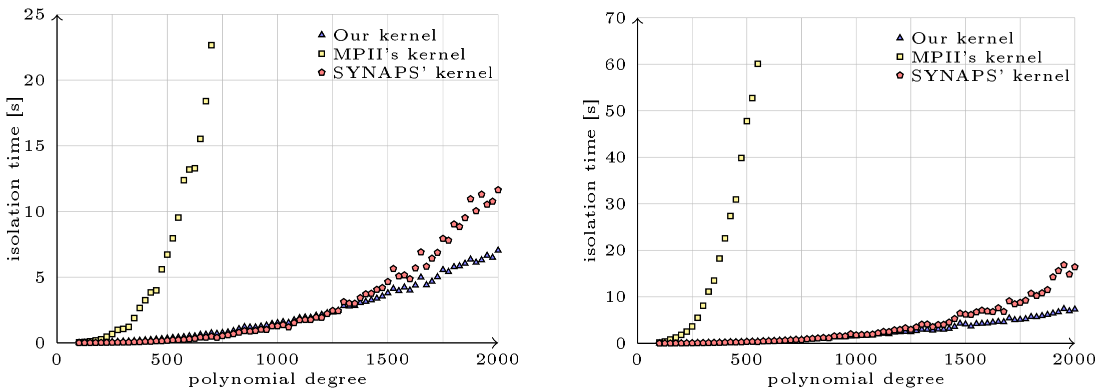


Figure 5.2: Running time for isolating all the real roots of degree 100 polynomials in terms of the maximum bitsize of their coefficients.



(a) bitsize 32 (b) bitsize 1000

Figure 5.3: Running time for isolating all the real roots of random polynomials with coefficients of fixed bitsize and depending on the degree.

dependently chosen from the standard normal distribution is $\frac{2}{\pi}\ln(d)+C+\frac{2}{\pi d}+O(1/d^2)$ where $C \approx 0.625735$ [58]; this gives, for degree 100 an average of about 3.6 roots (note that this bound matches extremely well experimental observations). We vary the maximum bitsize of all the coefficients from 2 000 to 25 000 and average each test over 100 trials.

Varying degree

We consider two sets of experiments in which we study random polynomials and Mignotte polynomials (which have two very close roots).

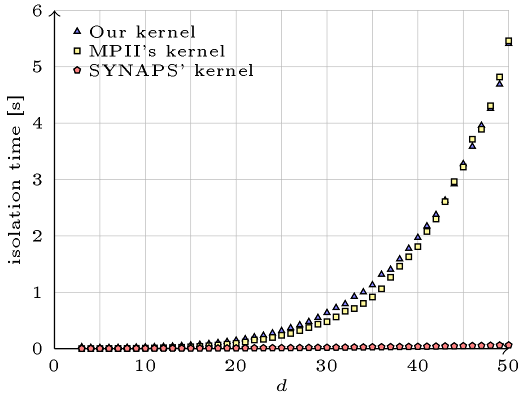


Figure 5.4: Running time for isolating all the real roots of Mignotte polynomials of the form $f = x^d - 2(kx - 1)^2$ in terms of the degree d.

We first consider polynomials with random coefficients of fixed bitsize for various values between 32 and 1 000. We then vary the degree of the polynomials from 100 to 2 000 and average our experiments over 100 trials (see Figure 5.3). Note that the above formula gives an expected number of roots varying from 3.6 to 5.5. We observe that the running time is almost independent of the bitsize in the considered range.

Finally, we test Mignotte polynomials, that is nearly degenerate polynomials of the form $x^d - 2(kx - 1)^2$. Such polynomials are known to be challenging for Descartes algorithms because two of their roots are very close to each other; the isolating intervals for these two roots are thus very small. For these tests, we used Mignotte polynomials with coefficients of bitsize 50, with varying degree d from 5 to 50. See Figure 5.4. We averaged the running time over 5 trials for each degree. We observed essentially no difference between our kernel and MPII's one; they take roughly 0.2 and 5.5 seconds for Mignotte polynomials of degree 20 and 50, respectively. However, SYNAPS' kernel is much more efficient as the continued fractions algorithm is not so affected by the closeness of the roots.

Discussion

We observe (Figure 5.1 (a)) that SYNAPS' kernel is more efficient than both our and MPII's kernel in the case of polynomials of small degree (*e.g.*, twelve) and small to moderately large coefficients (up to 2 000 bits per coefficient). However, for extremely large coefficients MPII's kernel is substantially more efficient (by a factor of up to 3 for coefficients of up to 50 000 bits) than both our and SYNAPS' kernels, which perform similarly.

For polynomials of reasonable large degree, both our and SYNAPS' kernels are much more efficient that MPII's kernel; furthermore these two kernels behave similarly for degrees up to 1 500 and our kernel becomes more efficient for higher degrees (by a factor 2 for degree 2 000).

We also observe that the running time is *highly* dependent of the various settings. For instance, our kernel is up to 5 times slower when using approximate evaluation for high-degree and high-bitsize polynomials. Also, MPII's kernel is in some cases about 10 times slower when changing the arithmetic kernel to LEDA, the representation of algebraic numbers and some internal algorithms such as the refinement function. This explains why our benchmarks on both MPII's and SYNAPS' kernels are substantially better than in Emiris et al. experiments [94].

We also observe that the running time of MPII's kernel is unstable in our experiments (Figures 5.1 and 5.2); surprisingly, this instability occurs when the experiments are performed on a 64-bits architecture, but it is stable on 32-bits architecture as shown in previous experiments [94]. As noted before, this is caused by a current bug in the implementation of CORE in 64-bits.

5.5.2 Benchmarks on comparison of algebraic numbers

We consider three suites of experiments for comparing algebraic numbers; see Figure 5.5. Recall that an algebraic number ρ is here represented by a polynomial F that vanishes at ρ and an isolating interval containing ρ but no other root of F. Recall also that the comparison of two algebraic numbers is done by (i) testing whether the intervals are disjoint; if so, report the ordering, otherwise (ii) compute the gcd of the two polynomials and test whether the gcd vanishes in the intersection of the two intervals; if so, report the equality of the numbers, otherwise (iii) refine the intervals until they are disjoint.

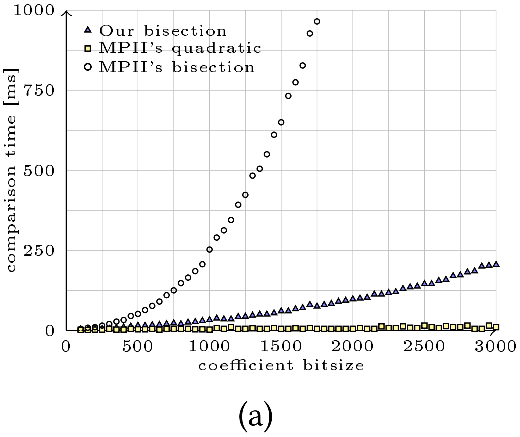


Figure 5.5: Running time for comparing two distinct close roots of two almost identical polynomials of degree 20 with (a) no common roots and (b) a common factor of degree 10.

First, we analyze the cost of trivial comparisons that is, when the two intervals representing the numbers are disjoint. For that we compare the roots of two random polynomials. We observe that, as expected, the comparison time is negligible and independent of both the degree of the polynomials and the bitsize of their coefficients.

Second, we analyze the cost of comparing roots that are very close to each other but whose associate polynomials have no common root. This case is expensive because we need to refine the intervals until they do not overlap; this is, however, not the worst situation because the gcd of the two polynomials is 1 which is tested efficiently with a modular gcd. We perform these experiments as follows. We generate pairs of polynomials, one with random coefficients and the other by only adding 1 to one of the coefficients of the first polynomial. Such polynomials are such that the i-th roots of both polynomials are very close to each other. We generate such pairs of polynomials with constant degree (equal to 20) and vary the maximum bitsize of the coefficients. As the bitsize increases, the pairs of roots that are close become even closer and thus the comparison time increases. The results of these experiments are presented in Figure 5.5 (a), which reports the average running time for comparing two close roots.[8] We show in this figure three curves, one corresponding to our bisection algorithm, and two cor-

[8]According to our first set of experiments, we can neglect the time for comparing two roots that are not close.

responding to the two refinement methods implemented in the MPII's kernel: the usual bisection and a quadratic refinement algorithm [1].

Third, we consider the, a priori, most expensive scenario in which we compare roots that are either equal or very close to each others and such that their associate polynomials have some roots in common. In this case, we accumulate the cost of computing a non-trivial gcd of the two polynomials with the cost of refining intervals when comparing two non-equal roots. In practice, we generate pairs of degree-20 polynomials each defined as the product of two degree-10 terms; one of these factors is random and common to the two polynomials; the other factor is random in one of the polynomials and slightly modified in the other polynomial where, slightly modified means, as above, that we add 1 to one of the coefficients. We then vary the maximum bitsize of the coefficients and average each test over four trials.

Discussion

We see in Figure 5.5 that the MPII's quadratic refinement algorithm largely outperforms the two bisection methods. However, our bisection method is faster than MPII's one, by a factor up to 10. We also observed that the running time for comparing equal roots is negligible compared to the cost of comparing close but distinct roots. (The running time reported in Figure 5.5 (b) is actually the total time for comparing all pairs of roots divided by the number of comparisons of close but distinct roots.) This explains why our kernel behaves similarly in Figures 5.5 (a) and 5.5 (b). Overall, it appears that comparing algebraic numbers that are very close is fairly time consuming and that the most time-consuming part of the comparison is the evaluation of polynomials performed during the interval refinements.

We mentioned in Section 5.4.5 that we implemented two refinement algorithms, but they behave similarly in our tests. It should be stressed that the MPII's kernel provides the same two approaches, and their quadratic refinement clearly outperforms their bisection algorithm. We do not have at the time an explanation for this discrepancy in the behavior of similar algorithms other than a maybe better implementation.

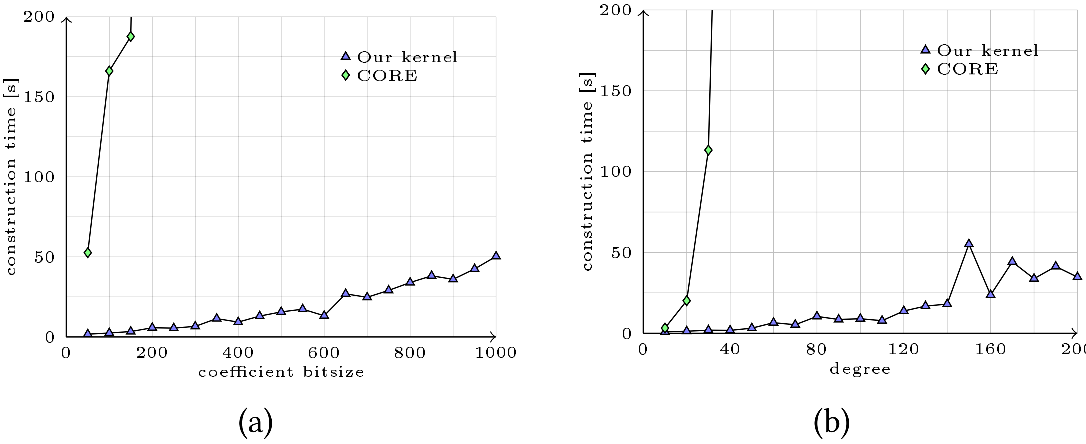


Figure 5.6: Arrangements of five polynomials, shifted four times each, (a) of degree 20 and varying bitsize and (b) of bitsize 32 and varying degree.

5.6 Arrangements

As an example of possible benefit of having efficient algebraic kernels in CGAL, we used our implementation to construct arrangements of polynomial functions. Wein and Fogel [164, 165] provided a CGAL package for calculating arrangements of general curves which requires as parameter a *traits class* containing the data structures to store the curves and various primitive operations, such as comparing the relative positions of points of intersection. We implemented a traits class which uses the functions of our algebraic kernel and compared its performance with another traits classes which comes with CGAL's arrangement package and uses the CORE library [45].

In order to generate challenging data sets we proceed as follows. First we generate n random polynomials. To each of them we add 1 to the constant coefficient, m times, thus producing a data set of $n(m+1)$ univariate polynomials. Notice that the arrangement of the graphs of these polynomials is guaranteed to be degenerate, *i.e.*, there are intersections with the same x-coordinate. The arrangements generated this way have four parameters: the number n of initial polynomials, the number m of "shifts" that we perform, the degree d of the polynomials, and the bitsize τ of their coefficients. We ran experiments varying the values of the last three of these parameters and setting $n = 5$.

Figure 5.6 (a) shows the running time in terms of the bitsize τ for a data set where

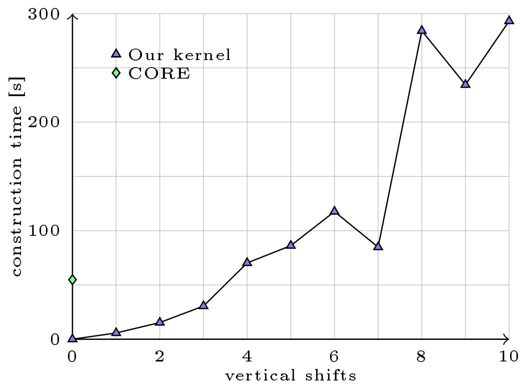


Figure 5.7: Arrangement generated from five random polynomials of bitsize 1000 and degree 20, varying the number of shifts performed.

$d = 20$ and $m = 4$ (giving 25 polynomials). Figure 5.6 (b) shows the running time in terms of the degree d for a second data set where $\tau = 32$ and $m = 4$. We see from these experiments that running time using CORE is considerably higher than when using our kernel. We also make the following observations.

Figure 5.6 (a) shows that the running time depends on the bitsize. When we change the bitsize of the coefficients of the random polynomials, the size of the arrangement does not change; that means that the number of comparisons and root isolations the kernel must perform is roughly the same in all the arrangements of the test suite. The isolation time for random polynomials does not depend much on the bitsize (as shown in Figure 5.2), but the comparison time does. It follows that the running time increases with the bitsize.

Figure 5.6 (b) shows that the running time depends also on the degree of the input polynomials. As we saw in Section 5.5, the formula that approximates the expected number of real roots of a random polynomial depends on its degree. The size of the arrangement thus increases with the degree of the input polynomials: each vertex is the root of the difference between two input polynomials, therefore there will be more vertices. Thus, when we increment the degree of the inputs, the number of comparisons and isolations increases; furthermore, the running time for each of these operations increases with the degree of the input.

We ran additional tests to see the impact of the input shifts in the calculation time. We generated five random polynomials of bitsize 1000 and degree 20. We calculated arrangements, then, varying the number of shifts we perform to each polynomial. As Figure 5.7 shows, we were only able to solve, using CORE, the first arrangement, generated without shifts (note the point on the vertical axis). We note that the running time increases fast with the number of shifts. This is reasonable since, each time we increase by 1 the number of shifts, we add to the arrangement n polynomials, hence increasing the number of vertices of the arrangement. Since the root isolation and comparison time remains the same (because the degree and the bitsize are constant), the running time increases with the number of these operations.

Note that these experiments on arrangements were performed before Kerber's thesis [104]. He presents the CGAL implementation of the computation of arrangements of algebraic plane curves, based on a bivariate algebraic kernel and using the CGAL arrangements package.

5.7 Conclusions and perspectives

We presented in this chapter a CGAL univariate algebraic kernel, which is now part of the CGAL library. We described its implementation and performed thorough experiments, comparing it to other similar implementations. Finally, we validated the algebraic kernel approach by comparing the performance of our algebraic kernel applied to the computation of arrangements of curves defined by univariate polynomials.

Our experiments exposed an erratic behavior on MPII's kernel when running in 64-bit architectures. This permitted the discovery of a bug in the 64-bit version of CORE. This proved again, as shown in Section 4.2, that thorough and competitive testing provide a framework to expose bugs in algebraic software. It is worth pointing out that CORE is a widely used tool in exact computing, and this bug was, up to our knowledge, not reported before.

The natural sequel of our work is the implementation of a CGAL bivariate algebraic kernel. Such kernel would give CGAL the ability to handle curves defined by bivariate polynomials. That is, CGAL would provide the framework for the implementation of algorithms like ISOTOP. Currently there exists a bivariate algebraic kernel developed at MPII. CA, tested in Chapter 4 was developed on top of this bivariate algebraic kernel. Future efforts of the CGAL community include the development of algebraic kernels capable of handling curves defined by polynomials on any number of variables, as well as interoperating existing kernels, with the aim of obtaining more efficient implementations.

Chapter 6

Complexity Analysis

It was noted so far that the bit-complexity model offers a way of analysing algorithms whose complexity is closer to the reality than analyses based on the real-RAM model. This closeness to the reality is given by the cost of arithmetic operators used in the analysis, which reflects more accurately the time spent by computers in multiple-precision arithmetic operations. It should be stressed, nevertheless, that the bit-complexity analysis of algebraic algorithms often produces important overestimations. They are produced by the fact that one usually analyzes worst-case execution of algorithms, and algebraic algorithms perform in practice much better than the worst-case. For instance, it is assumed in worst-case analyses that a univariate polynomial of degree n has n roots, while in practice it often has much less roots (see Section 5.5.1 for details on this fact). This gap is very common in the bit-complexity analysis of algebraic algorithms. Note however that (worst-case) real-RAM analyses also often overestimate because of worst-case considerations. One standard technique to reduce the gap between standard worst-case analyses and practical performances is to perform output-sensitive analyses, that is, analyses based on the size of the output of the algorithms. Unfortunately, output-sensitive analyses tend to be harder to realize. Note that such analyses are also done in the worst-case.

This chapter introduces output-sensitive bit-complexity analyses of two algorithms considered in this thesis. Section 6.1, states that the complexity of the algorithm for computing the topology of planar algebraic curves is $\widetilde{\mathcal{O}}_B(R\, d^{22}\, \tau^2)$, where $\widetilde{\mathcal{O}}_B$ denotes the bit-complexity in which (poly-)logarithmic factors are ignored, R is the number of critical points of the curve and d and τ are the degree and maximum bitsize of the polynomial which defines the curve. In Section 6.2, we prove that the complexity of computing an arrangement of n curves defined by univariate polynomials of degree at most d and bitsize at most τ is $\widetilde{\mathcal{O}}_B((n+k)\, d^2\, u(\tau + d\, u))$, where k is the number of intersection points between the curves and u is a bound on the bitsizes of the separation bounds of the differences between two of the input polynomials.

The bit-complexity analysis presented in Section 6.1 appeared in [40]. The bit-complexity analysis introduced in Section 6.2 was done in collaboration with Elias Tsigaridas and has not been published yet.

6.1 Topology of planar algebraic curves

The main algorithmic contribution of this thesis is the algorithm to compute the topology of curves presented in Chapter 3. In this section, the complexity of this algorithm is stated as follows.

Theorem 19. *Let an algebraic curve $\mathcal{C}$ be given by a square-free polynomial $f \in \mathbb{Z}[x, y]$ of total degree bounded by d and coefficients of bitsize bounded by τ. Let R be the number of critical points of the curve. The bit complexity of our algorithm for the computation of the topology of the curve $\mathcal{C}$ is $\widetilde{\mathcal{O}}_B(R\, d^{22}\, \tau^2)$, which is $\widetilde{\mathcal{O}}_B(N^{26})$, where $N = \max\{d, \tau\}$.*

6.1.1 Overview of the complexity analysis

In order to derive the bit complexity of the algorithm presented in Section 3.3, we analyze the complexity of each step. The first two steps consist of computing the isolating boxes of the critical points and multiplicities. The analysis of these steps is presented in Section 6.1.4. In the third step, we refine the isolating boxes of the x-extreme points.

The complexity of this step is bounded by the complexity of the next one, so we do not consider it explicitly. During the first part of the fourth step we refine the isolating boxes of the singular points with respect to their isolating curve. The analysis of this step is done in Section 6.1.5. During the second part, we refine the isolating boxes of the singular points, until the curve does not intersect them on the top and on the bottom. The analysis of this operation is done in Section 6.1.6. We conclude the proof of Theorem 19 in Section 6.1.7.

6.1.2 Definitions and notation

Throughout this chapter, combinatorial and bit complexities will be denoted by $\mathcal{O}$ and $\mathcal{O}_B$, respectively. The $\widetilde{\mathcal{O}}$ and $\widetilde{\mathcal{O}}_B$ notations refer to complexities in which (poly-)logarithmic factors are ignored.

The bitsize of a rational number is defined as the maximum bitsize of its numerator and denominator. The bitsize of a polynomial is the maximum bitsize of its coefficients. Bitsizes of numbers and polynomials will be denoted with the Greek letter τ. The degrees of polynomials will be denoted with the letter d. In order to simplify notation, we assume in the sequel that $d = \mathcal{O}(\tau)$. However, we still express complexities in terms of d and τ when it is simple enough since d is often much smaller than τ. We may also assume that the univariate polynomials that we compute with, are square free. This assumption does not change the complexity since the computation of their square-free part and computations with their square-free part, is of no extra cost. Indeed, for a polynomial of degree d and bitsize τ, its square-free part has degree $\mathcal{O}(d)$ and bitsize $\mathcal{O}(d+\tau) = \mathcal{O}(\tau)$ and it can be computed in $\widetilde{\mathcal{O}}_B(d^2\tau)$ [115]. We use the notion of separation bound of a polynomial (or of a zero-dimensional system of polynomial equations) which is a lower bound on the minimum distance between any two (possibly complex) roots. We call the bitsize of a separation bound s the minimum integer $\sigma \geq 0$ such that $s > 2^{-\sigma}$. In other words, the bitsize of the separation bound is the number of bits needed to represent the largest lower bound of the form $2^{-\sigma}$ that is smaller than s.

Isolations of roots of systems via a RUR require some machinery from interval analysis, we briefly recall the basics and refer to [4] for additional details. For an interval $A = [a_1, a_2]$, let its width be $w(A) = a_2 - a_1$ and its absolute value be $|A| = \max(|a_1|, |a_2|)$. For a two-dimensional box $A \times B$, let $w(A \times B) = \max(w(A), w(B))$ and $|A \times B| = \max(|A|, |B|)$. We denote by $I(\mathbb{R}^n)$ the set of products of n real intervals. A polynomial in n variables has a natural extension to a function over $I(\mathbb{R}^n)$ by a process we call evaluation with interval arithmetic. More precisely, for a polynomial f, we denote by the bold letter $\mathbf{f}$ the corresponding interval function defined over $I(\mathbb{R}^n)$ by replacing the usual operations $+, -, \times$ by interval operations (note that the order in which the operations are processed can change the result, but this issue is irrelevant for our computations). In the sequel, we bound the number of times we refine the isolating intervals of the roots of the univariate polynomial of the RUR. We assume that every refinement divides by at least two the interval width.

6.1.3 Preliminaries

Before analyzing each step of the algorithm, we state the bitsize complexity of some recurrent basic computations. Let f be a univariate integer polynomial of degree d and bitsize τ, and x be a rational of bitsize σ.

Lemma 20 ([166]). *The bitsize of the separation bound of f is in $\mathcal{O}(d\tau)$. Similarly, the bitsize of the endpoints of isolating intervals of the roots of f is in $\mathcal{O}(d\tau)$. Moreover, the absolute value of the roots of f is in $\mathcal{O}(2^\tau)$.*

The following lemma will be proven using Horner's scheme. Note that the complexity also holds for other schemas.

Lemma 21. *The evaluation of f over x has complexity $\widetilde{\mathcal{O}}_B(d(\tau + d\sigma))$, while the number $f(x)$ has bitsize $\mathcal{O}(\tau + d\sigma)$.*

Proof. Evaluating the polynomial $f = \sum_{i=0}^{d} a_i x^i$ using Horner's method gives raise to the sequence

$$
\begin{aligned}
b_d &= a_d \\
b_{d-1} &= a_{d-1} + b_d x \\
&\vdots \\
b_0 &= a_0 + b_1 x
\end{aligned}
$$

where $b_0 = f(x)$ is the desired evaluation.

Given that the coefficients of f have bitsize $\mathcal{O}(\tau)$ and x has bitsize $\mathcal{O}(\sigma)$, we see that b_{d-i} has bitsize $\mathcal{O}(\tau + i\sigma)$ for $0 \leq i \leq d$. Thus, b_0 has bitsize $\mathcal{O}(\tau + d\sigma)$.

To bound the complexity of the evaluation, note that each step introduces the computation of a multiplication and an addition. The latter will be neglected in the analysis since, in this case, it is majored by the cost of the multiplication. The cost of the computation of b_{d-i} for $1 \leq i \leq d$ is the cost of multiplying b_{d-i+1} times x. That is, it is the cost of multiplying numbers of bitsize $\tau + i\sigma$ and σ. Since this operation is repeated $\mathcal{O}(d)$, the cost of this operation is d times the complexity of the multiplication of numbers of maximum bitsize $\mathcal{O}(\tau + d\sigma)$. This yields, using FFT multiplication [81, Chapter 8], $\widetilde{\mathcal{O}}_B(d(\tau + d\sigma))$. □

Since the interval evaluation of the operations $+, -, \times$ are a constant time more expansive that their usual counterparts we obtain the corollary:

Corollary 22. *The evaluation of f using interval arithmetic over an interval I with endpoints of the same bitsize as x, i.e. the computation of $\mathbf{f}(I)$, has complexity $\widetilde{\mathcal{O}}_B(d(\tau + d\sigma))$, while the interval $\mathbf{f}(I)$ has endpoints with bitsize $\mathcal{O}(\tau + d\sigma)$.*

In our algorithm we need, several times, to evaluate univariate and bivariate polynomials over intervals. This is done using classical interval arithmetic operations. Theoretically, we need to control how large an interval becomes when a polynomial operation is performed. The following lemma bounds the increase of the width of an interval by evaluation by interval arithmetic of a polynomial.

Lemma 23. *Let P be a univariate rational polynomial of degree d and bitsize τ, and A be an interval such that $|A| \leq 2^{\sigma}$ with $\sigma \geq 0$; then*

$$w(\mathbf{P}(A)) \leq 2^{\tau+d\sigma} d^2 w(A).$$

Let Q be a bivariate rational polynomial of total degree d and bitsize τ, and B be an interval such that $|B| \leq 2^{\sigma}$ with $\sigma \geq 0$; then

$$w(\mathbf{Q}(A,B)) \leq 2^{\tau+d\sigma+1} d^3 w(A \times B).$$

Proof. We apply the basic formulas for the sum and the product of intervals [4, Theorem 9, p.15], which are for any real number a and integer $n \geq 1$:

$$\begin{aligned} w(A \pm B) &= w(A) + w(B), & w(aA) &= |a| w(A), \\ w(AB) &\leq w(A)|B| + |A| w(B), & w(A^n) &\leq n|A|^{n-1} w(A). \end{aligned}$$

Let $P(x) = \sum_{i=0}^{d} c_i x^i$ with $|c_i| \leq 2^{\tau}$ and $Q(x,y) = \sum_{i,j\geq 0}^{i+j\leq d} c_{ij} x^i y^j$ with $|c_{ij}| \leq 2^{\tau}$. We have:

$$\begin{aligned} w(\mathbf{P}(A)) &= \sum_{i=0}^{d} |c_i| w(A^i) \leq 2^{\tau} \sum_{i=0}^{d} i|A|^{i-1} w(A) \leq 2^{\tau} w(A) d \sum_{i=1}^{d} |A|^{i-1} \\ &\leq 2^{\tau} w(A) d^2 \max(1, |A|^{d-1}) \leq 2^{\tau} w(A) d^2 2^{d\sigma} \leq 2^{\tau+d\sigma} d^2 w(A). \\ w(\mathbf{Q}(A,B)) &= \sum |c_{ij}| w(A^i B^j) \leq 2^{\tau} \sum w(A^i)|B|^j + w(B^j)|A|^i \\ &\leq 2^{\tau} \sum i w(A)|A|^{i-1}|B|^j + j w(B)|A|^i |B|^{j-1} \\ &\leq 2^{\tau} d^3 2 w(A \times B) \max(1, 2^{\sigma(d-1)}) \leq 2^{\tau+d\sigma+1} d^3 w(A \times B). \end{aligned}$$

□

6.1.4 Computation of the isolating boxes

The first step of the algorithm is the computation of the isolating boxes of the singular and the extreme points. For the complexity of this step it suffices to compute the complexity of solving the system of critical points $I_c = \mathrm{I}(f, f_y)$.

The steps for solving the system are the followings. We compute the Gröbner basis of I_c and the RUR of the system. We solve the univariate polynomial of the RUR and, using the isolating intervals of its real roots, we compute boxes that contain the real solutions of the systems. Finally, we refine the boxes until they are all disjoint.

Gröbner basis and RUR

We compute the Gröbner basis of I_c with the degree reverse lexicographic order in $\widetilde{\mathcal{O}}_B(d^8\tau)$ [111]. Next we compute the RUR representation of the solution of the system and multiplicities in $\widetilde{\mathcal{O}}_B(d^{13}\tau)$, the reader may refer to [140] for more details. Besides I_c we have to solve the systems $I_{s,k}$, for $1 \leq k \leq d$, to determine the multiplicities of the singular points. We may assume that each of them could be solved in $\widetilde{\mathcal{O}}_B(d^{13}\tau)$, as in the case of I_c, since the polynomials have degrees bounded by d and the bitsize of the coefficient is bounded by $\widetilde{\mathcal{O}}(\tau)$. Hence the total complexity is $\widetilde{\mathcal{O}}_B(d^{14}\tau)$. However, this is an overestimation since the systems are over-determined and thus the Gröbner bases could be computed faster [12] and in practice we don't need to solve all d of them.

The RUR representation has the following form: $h(T) = 0, x = \frac{g_x(T)}{g_0(T)}, y = \frac{g_y(T)}{g_0(T)}$, where h, g_0, g_x, $g_y \in \mathbb{Q}[T]$. The polynomial $h(T)$ is actually the so called *u-resultant* [37]. It has degree $\mathcal{O}(d^2)$ and bitsize $\mathcal{O}(d^2 + d\tau) = \mathcal{O}(d\tau)$. One way to see this is to consider a curve (and thus the system I_c) in generic position, this is without loss of generality since shearing the curve in generic position does not increase the bitsizes more than by a factor in $\mathcal{O}(\lg(d))$. In this case we can consider as the polynomial $h(T)$ the projection of the system on the x-axis or, in other words, the resultant of the system with respect to y. Under this notion, g_0 and g_x, respectively g_0 and g_y could be seen as the coefficients of the first non-vanishing sub-resultant of f and f_y, with respect to y, respectively with respect to x. Thus, the degree of these polynomials is $\mathcal{O}(d^2)$ and their bitsize is $\mathcal{O}(d\tau)$.

We can simplify the expressions of x and y, if we take into account that $h(T)$ and $g_0(T)$ are relative prime. Indeed, we can compute a polynomial $g(T)$ such that $g \cdot g_0 = 1 \mod h$ and thus express the coordinates of the solutions as $x = h_x(T) = g_x(T)\, g(T)$ and $y = h_y(T) = g_y(T)\, g(T)$. The degree of g is $\mathcal{O}(d^2)$ and its bitsize is $\mathcal{O}(d^3\tau)$, as g is a Bézout coefficient of the extended Euclidean division of g_0 and h [81]. The same bounds hold for $h_x(T)$ and $h_y(T)$.

Computation of the boxes

In order to solve the system, it suffices to compute the isolating intervals of the real roots of h, and substitute them in h_x and h_y using interval arithmetic. This gives isolating boxes that contain the real roots of the system. Finally, we refine these boxes until they become disjoint.

Lemma 24. *To ensure that the boxes of the critical points are disjoint, it is sufficient to refine $\mathcal{O}(d^3\tau)$ times each isolating interval of the corresponding root of the univariate polynomial of the RUR of the critical points.*

Proof. One needs to refine the isolating intervals of the real roots of h until the corresponding boxes of the system, computed by interval evaluation with h_x and h_y, become disjoint. In other words, the isolating boxes should have width smaller than 2^{-s_c}, where s_c is the separation bound of the system of critical points. If $2^{-\mu}$ is a lower bound on the width of the isolating intervals of h, Lemma 23 yields a value μ, so that the evaluations by h_x and h_y give intervals of width at most 2^{-s_c}. We consider only the polynomial h_x, since the computation is similar for h_y. Lemma 23 applied with h_x of bitsize $\mathcal{O}(d^3\tau)$ and degree $\mathcal{O}(d^2)$ evaluated at the roots of h of absolute value $\mathcal{O}(2^{d\tau})$ (by Lemma 20) yields:

$$2^{\mathcal{O}(d^3\tau)+\mathcal{O}(d^2)\mathcal{O}(d\tau)} d^2 2^{-\mu} \leq 2^{-s_c}.$$

Thus, it suffices to consider μ in $\mathcal{O}(s_c + d^3\tau)$.

On the other hand, the separation bound of the critical points is larger than the separation bounds of the x (or y) coordinates of these points. The coordinates are roots of the resultant of f and f_y with respect to y (or x), which is a polynomial of degree $\mathcal{O}(d^2)$ and bitsize $\mathcal{O}(d\tau)$. Hence the separation bound of the coordinates is of bitsize $\mathcal{O}(d^3\tau)$ (by Lemma 20). This is also a bound for the separation bound of the critical points thus s_c is in $\mathcal{O}(d^3\tau)$ and $\mathcal{O}(s_c + d^3\tau)$ is also in $\mathcal{O}(d^3\tau)$.

By Lemma 20, the isolating intervals of h have initial width $\mathcal{O}(2^{d\tau})$. Thus, to reduce them to the width $2^{-\mu}$, it suffices to perform $\mathcal{O}(d\tau + d^3\tau) = \mathcal{O}(d^3\tau)$ refinements, provided that the refinements divide the interval widths by at least two. □

6.1.5 Refinement with respect to the isolating curve

We require that the isolating boxes of singular points avoid their associated curve $f_k = \partial^k f/\partial y^k$. This is ensured by refining the isolating boxes of the singular points so that the evaluation of f_k, using interval arithmetic, results an interval that does not contain zero.

Before stating the main lemma of this section, we need to compute the separation bound needed for the proof of Lemma 27. We first need a refinement, due to Yap [166], of the Gap theorem by Canny [37].

Theorem 25 ([166] Gap theorem 11.45). *Let $\Sigma = \{A_1, \ldots, A_n\} \subseteq \mathbb{Z}[x_1, \ldots, x_n]$ be a system of n polynomials, not necessarily homogeneous. Suppose that Σ has finitely many complex zeros and $(\xi_1, \ldots, \xi_n)$ is one of these zeros. Assume $d_i = \deg(A_i)$ and*

$$K := \max\{\sqrt{n+1}, \max\{\|A_i\|_2 \mid 1 \leq i \leq n\}\},$$

where $\|A_i\|_2$ is the usual Euclidean norm of the vector of coefficients of the polynomial A_i. If $|\xi_i| \neq 0$, $i = 1, \ldots, n$, then

$$|\xi_i| > (2^{2/3}NK)^{-D}\, 2^{-(n+1)d_1\cdots d_n},$$

where

$$N := \binom{1 + \sum_{i=1}^n d_i}{n}, \quad D := \left(1 + \sum_{i=1}^n \frac{1}{d_i}\right) \prod_{i=1}^n d_i.$$

We now prove, following closely [168], the lemma we use in the proof of Lemma 27.

Lemma 26. *Let* p *be a critical point of a curve* $\mathscr{C}_f$*, without vertical lines, defined by a square-free polynomial* f *of degree* d *and bitsize* τ*. Let* $f_k = \partial^k f/\partial y^k$*, where* $2 \leq k \leq d$*. If* p *is not a point on* f_k*, then* $|f_k(\mathrm{p})| > 2^{-s_v}$ *with* s_v *in* $\widetilde{\mathcal{O}}(d^4\tau)$*.*

Proof. We consider the following system for $k \in \{2, \ldots, d\}$

$$\begin{aligned} A_1 &: f(\mathrm{p}) = 0, \\ A_2 &: f_y(\mathrm{p}) = 0, \\ A_3 &: h - f_k(\mathrm{p}) = 0. \end{aligned}$$

The system is zero-dimensional because the number of critical point is finite, and we can apply Theorem 25, where

$$d_1 = d, \quad d_2 = d - 1, \quad d_3 \leq 2(d-k),$$

$$N \leq \binom{4d-2k}{3} \leq 16d^3, \quad D \leq (d+2)(d-1)(2d-1) \leq 4d^3.$$

The bitsize of the norm of a polynomial is the bitsize if the polynomial itself. Since the bitsize of f and f_y is in $\mathcal{O}(\tau)$ and that of f_k is in $\widetilde{\mathcal{O}}(d\tau)$, we conclude that the bitsize of K is in $\widetilde{\mathcal{O}}(d\tau)$. The factor that gives the bitsize in the lower bound of Theorem 25 is thus K^{-D} and h is bounded by a value of bitsize in $\widetilde{\mathcal{O}}(d^4\tau)$. □

We are now ready to introduce the lemma that bounds the number of refinements needed to avoid box overlaps.

Lemma 27. *To ensure that the boxes of singular points do not overlap their associated curve* f_k*, it is sufficient to refine* $\widetilde{\mathcal{O}}(d^4\tau)$ *times the corresponding roots of the univariate polynomial of the RUR.*

Proof. Consider a box $B = I \times J$ that isolates a singular point p. Assume, without loss of generality, that the width of B is $2^{-\mu}$. Let k be such that f_k is the isolating curve for

p. We need to ensure that the evaluation of f_k over B does not contain 0. A sufficient condition is that $w(\mathbf{f_k}(B)) < \delta$ with $\delta < |f_k(p)|$. Defining s_v as the maximum bitsize of the values of f_k at the singular points of f (more precisely $s_v = \max_{s\in\mathbb{N}^*}\{|f_k(p)| > 2^{-s}, p$ singular point of f and $f_k(p) \neq 0\}$), we can choose $\delta = 2^{-s_v}$.

f_k is of degree $\mathcal{O}(d)$ and bitsize $\widetilde{\mathcal{O}}(d\tau)$ (by Stirling formula). The absolute value of a box of a singular point is $\mathcal{O}(2^{d\tau})$ since the x or y coordinates of a box are roots of the resultant of f and f_y with respect to y or x, and such a resultant has bitsize $\mathcal{O}(d\tau)$ (Lemma 20). Lemma 23 applied for the evaluation of f_k over a singular point box yields:

$$2^{\widetilde{\mathcal{O}}(d\tau)+\mathcal{O}(d)\mathcal{O}(d\tau)+1}d^3 2^{-\mu} \leq 2^{-s_v}.$$

Thus, it suffices to consider μ in $\mathcal{O}(s_v + d^2\tau)$.

On the other hand, Lemma 26 gives that s_v is in $\widetilde{\mathcal{O}}(d^4\tau)$, hence μ is in $\widetilde{\mathcal{O}}(d^4\tau)$.

Let $2^{-\nu}$ be the width of the isolating interval of h that corresponds to the singular point via the RUR. Then ν should be such that the box computed via the RUR has width $2^{-\mu}$. Applying Lemma 23 again, for the evaluation of h_x of degree $\mathcal{O}(d^2)$ and bitsize $\mathcal{O}(d^3\tau)$ over this isolating interval of absolute value $\mathcal{O}(2^{d\tau})$, ν should satisfy:

$$2^{\mathcal{O}(d^3\tau)+\mathcal{O}(d^2)\mathcal{O}(d\tau)}d^2 2^{-\nu} \leq 2^{-\mu}.$$

Thus, it suffices to consider ν in $\widetilde{\mathcal{O}}(d^4\tau)$.

We conclude, as in the proof of Lemma 24, that it suffices to perform $\widetilde{\mathcal{O}}(d^4\tau)$ refinements. □

6.1.6 Refinement of the singular points to avoid top/bottom crossings

The last step that we need to consider is the analysis of the refinement of the isolating boxes of the singular points, until there is no intersection between the curve and their top and bottom sides.

Lemma 28. *To avoid top/bottom intersection between the curve and the boxes of singular points, it is sufficient to refine $\widetilde{\mathcal{O}}(d^9\tau)$ times the corresponding roots of the univariate polynomial of the RUR.*

Proof. Consider a singular point and its isolating box refined according to Lemmas 24 and 27. We further refine the x-coordinate of the box until the top (or equivalently the bottom) side does not intersect the curve. The line supporting the top side is of the form $y = c$, for some constant c. Hence, the x-coordinates of the intersections of the curve with the top side are among the roots of $f(x, c)$. Consider the polynomial P whose roots are the roots of $f(x, c)$ and the x-coordinates of the critical points. A sufficient condition to avoid intersection on the top side is to ensure that the x-width of the box is smaller than the separating bound of P.

The bitsize of c is the same as that of the evaluation of h_x over an end-point a of an isolating interval of a root of h. From Lemmas 24 and 27, the bitsize of a is in $\widetilde{\mathcal{O}}(d^4\tau)$. Since h_x is of degree $\mathcal{O}(d^2)$ and bitsize $\mathcal{O}(d^3\tau)$, c has bitsize $\widetilde{\mathcal{O}}(d^6\tau)$ (Lemma 21). Hence $f(x, c)$ is a polynomial of degree $\mathcal{O}(d)$ and bitsize $\widetilde{\mathcal{O}}(d^7\tau)$. The polynomial P is the product of $f(x, c)$ and the resultant with respect to y of f and f_y, thus its degree is in $\mathcal{O}(d^2)$ and its bitsize is in $\widetilde{\mathcal{O}}(d^7\tau)$. The bitsize of the separation bound of P is thus δ in $\widetilde{\mathcal{O}}(d^9\tau)$.

Let $2^{-\mu}$ be the width of the isolating intervals of h corresponding to the singular point by the RUR. Lemma 23 applied with h_x of bitsize $\mathcal{O}(d^3\tau)$ and degree $\mathcal{O}(d^2)$ yields:

$$2^{\mathcal{O}(d^3\tau)+\mathcal{O}(d^2)\mathcal{O}(d\tau)} d^2 2^{-\mu} \leq 2^{-\delta}$$

Thus, it suffices to consider μ in $\widetilde{\mathcal{O}}(d^9\tau)$.

We conclude, as in the proof of Lemma 24, that it suffices to perform $\widetilde{\mathcal{O}}(d^9\tau)$ refinements.

□

6.1.7 Overall complexity

Combining the results from Section 6.1.4, 6.1.5 and 6.1.6 we can now prove Theorem 19.

Proof of Theorem 19. Combining the results of Lemmas 24, 27, and 28, we prove that the algorithm performs $\widetilde{\mathcal{O}}(d^9\tau)$ refinements. Each refinement consists of an evaluation of h, h_x and h_y over a rational number of bitsize $\widetilde{\mathcal{O}}(d^9\tau)$. Using Horner's rule, each evaluation of these polynomials of degree $\mathcal{O}(d^2)$ and bitsize $\mathcal{O}(d^3\tau)$ has complexity $\widetilde{\mathcal{O}}_B(d^2(d^3\tau + d^2d^9\tau)) = \widetilde{\mathcal{O}}_B(d^{13}\tau)$ (Lemma 21). The complexity of the $\widetilde{\mathcal{O}}(d^9\tau)$ refinements is thus in $\widetilde{\mathcal{O}}_B(d^{13}\tau d^9\tau) = \widetilde{\mathcal{O}}_B(d^{22}\tau^2)$. If there are R singular points the total cost is thus $\widetilde{\mathcal{O}}_B(Rd^{22}\tau^2)$.

Note that the costs of Gröbner and RUR computations are dominated. Finally, the complexity of dealing with vertical asymptotes (Step 5), vertical lines and the connection part of the algorithm (Step 6) is dominated by the complexity of the other steps.

Finally, if $N = \max\{d, \tau\}$, note that R is in $\mathcal{O}(d^2) = \mathcal{O}(N^2)$, and so the total complexity of the algorithm is $\widetilde{\mathcal{O}}_B(N^{26})$. □

6.2 Arrangements of x-monotone algebraic curves

In this section we present an output-sensitive analysis of the bit complexity of the standard sweep-line algorithm [16] for computing arrangements of graphs of univariate polynomial functions. Kerber's thesis [104] introduces an algorithm to compute arrangements of algebraic plane curves defined by bivariate polynomials. He also proves an output sensitive worst-case bit-complexity bound of this algorithm. His algorithm is based on the analysis of a pair of curves, and the total complexity of the algorithm is given in function of the amounts of pairs of curves that need to be analyzed. On the other hand, our bit-complexity bound is given in terms of the size of the arrangement and of separation bounds. The main result of this section is stated in the following theorem.

Theorem 29. *The arrangement of n curves, defined by univariate polynomials of degree at most d with integer coefficients of bitsize at most τ can be computed in time* $\widetilde{\mathcal{O}}_B((n +$

$k)d^2 u(\tau + d\,u))$, where k is the number of intersection points between the curves and u is a bound on the bitsize of the separation bounds of all differences between two of the input polynomials.

Note that, if $N = \max\{n, k, d, \tau, u\}$, this bound is in $\widetilde{\mathcal{O}}_B(N^6)$ which is quadratic in the size of the input in the worst case. Before proceeding to the proof, some auxiliary results need to be stated. First, we note that the results of [63, 70] can easily be generalized to express the complexities of isolating the real roots of a polynomial in terms of the separation bounds of the considered instances of polynomials rather than in terms of worst-case separation bounds.

Proposition 30. *The real roots of a univariate polynomial of degree d with integer coefficients of bitsize at most τ and separation bound of bitsize s can be isolated, with their multiplicities, in $\widetilde{\mathcal{O}}_B(d^2\, s(\tau + d\,s))$ time. The bitsize of the endpoints of the isolating intervals is in $\mathcal{O}(s)$.*

Proof. In [63], it was proven that the worst-case complexity is $\widetilde{\mathcal{O}}_B(d^6 + d^4\,\tau^2)$. For completeness, we show a proof of this result.

Let s_k be the bitsize of the local separation bound of k-th root of the polynomial, that is $s_k = -\log|\gamma_k - \gamma_{c(k)}|$, where $\gamma_{c(k)}$ is the closest root to γ_k. Then it holds, *e.g.* [50, 57, 155],

$$\begin{aligned}\sum_{k=1}^{d} s_k &= -\sum_{k=1}^{d} \log|\gamma_k - \gamma_{c(k)}| \\ &= -\log\prod_{k=1}^{d} |\gamma_k - \gamma_{c(k)}| \\ &= \mathcal{O}(d^2 + d\tau).\end{aligned}$$

To isolate γ_k, one needs to perform s_k shift operations. At each step we perform a polynomial shift with a number of bitsize $\mathcal{O}(s_k)$. Each shift operation costs $\widetilde{\mathcal{O}}_B(d\tau +$

$d^2 s_k)$ [80]. We sum over all the roots to get the total complexity,

$$\begin{aligned}\sum_{k=1}^{d} \widetilde{\mathcal{O}}_B(d\tau s_k + d^2 s_k^2) &= \widetilde{\mathcal{O}}_B\Big(d\tau \sum_{k=1}^{d} s_k + d^2 \sum_{k=1}^{d} s_k^2\Big) \\ &= \widetilde{\mathcal{O}}_B\Big(d\tau(d^2 + d\tau) + d^2\Big(\sum_{k=1}^{d} s_k\Big)^2\Big) \\ &= \widetilde{\mathcal{O}}_B(d^3\tau + d^2\tau^2 + d^2(d^2 + d\tau)^2) \\ &= \widetilde{\mathcal{O}}_B(d^6 + d^4\tau^2).\end{aligned}$$

To derive an output sensitive result, we now replace the worst-case separation bound by s. We let $s = \max_k(s_k)$, then the complexity becomes

$$\sum_{k=1}^{d} \widetilde{\mathcal{O}}_B(d\tau s + d^2 s^2) = \widetilde{\mathcal{O}}_B(d^2\tau s + d^3 s^2).$$

□

We also recall an output-sensitive result on the complexity of comparing the roots of two polynomials, whose proof will be shown for completeness.

Proposition 31 ([70]). *Two real algebraic numbers defined as roots of polynomials of degree at most d with integer coefficients of bitsize at most τ and separation bounds of bitsize at most s can be compared in $\widetilde{\mathcal{O}}_B(d^2(\tau + s))$ time.*

Proof. Let γ_1 and γ_2 be the two real algebraic numbers, f_1 and f_2 the two defining polynomials, and I_1 and I_2 the isolating intervals.

Let $J = I_1 \cap I_2$. When $J = \emptyset$, or only one of γ_1 and γ_2 belongs to J, then we can easily compare them. If $\gamma_1, \gamma_2 \in J$, then $\gamma_1 \geq \gamma_2 \Leftrightarrow f_2(\gamma_1) \cdot f_2'(\gamma_2) \geq 0$. The sign of $f_2'(\gamma_2)$ is know from the root isolation process. It remains to compute the sign of $f_2(\gamma_2)$. For this, we need to evaluate the Sturm-Habicht sequence of f_1 and f_2 over the endpoints of J. The evaluation of such a sequence, of polynomials of degree d and bitsize τ, over a number of bitsize s costs $\widetilde{\mathcal{O}}_B(d^2(\tau + s))$. □

These complexity results yield, almost directly, the output-sensitive bit complexity of the standard sweep-line algorithm for computing arrangements of graphs of univariate polynomial functions, stated in Theorem 29.

Proof of Theorem 29. Recall first that the combinatorial complexity (*i.e.*, the complexity in the real-RAM model) of the standard sweep-line algorithm [16] for computing arrangements of n algebraic curves of bounded degree is $\mathcal{O}((n+k)\ \log n)$, where k is the number of intersection points (see, *e.g.*, [51, 76]). To evaluate the bit complexity of the algorithm, we split the analysis in two parts. In the first part, we consider the complexity of the construction of the intersection points of the curves. In the second part, we consider the cost of comparing the x-coordinates of the intersection points.

In order to compute (that is, to isolate) the intersection points of two curves $y = f_1(x)$ and $y = f_2(x)$, represented by polynomials $f_1, f_2 \in \mathbb{Z}[x]$ of degree at most d with integer coefficients of bitsize at most τ and separation bound of bitsize at most s, we can first isolate the real roots of the polynomial $f(x) = f_1(x) - f_2(x)$, which can be done in time $\widetilde{\mathcal{O}}_B(d^2\, u(\tau + d\, u)$, where u is a bound on the bitsize of the separation bound of $f(x) = f_1(x) - f_2(x)$ (by Proposition 30). We can then compute the image by f of these intervals in time $\widetilde{\mathcal{O}}_B(d(\tau + du))$, by Corollary 22. In the sequel, we will denote by u a bound on the bitsize of the separation bounds on *all* pairs of polynomials.

To begin the algorithm we need to compute a vertical line that is to the left all the intersection points between the curves. The x-coordinates of the intersection points of a pair of polynomials f_1 and f_2 are the roots of $f_1 - f_2$. A simple bound on the roots of a polynomial can be obtained by taking the greatest absolute value between its coefficients. The biggest absolute value of the coefficient of $f_1 - f_2$ will be smaller or equal than two times the maximum absolute value of the coefficients of these two polynomials. Thus the x-coordinate of such a vertical line can be obtained by taking minus two times the absolute biggest absolute value of the coefficients of the n polynomials. To compute a lower bound on the roots of the n polynomials, it suffices thus to compare all the $n\, d$ coefficients. The cost doing this is $\widetilde{\mathcal{O}}_B(n\, d\, \tau)$ and is dominated by the other steps of the algorithm.

We then compute the intersection points of this line with all the curves, so that to order the curves along the sweep line. The intersection of a vertical line with one curve is obtained by evaluating the polynomial defining the curve in the value of the x-coordinate of the line. The x-coordinate of the line has bitsize $\tau + 1$ since it corresponds to a coefficient times two, thus each evaluation can be done in time $\widetilde{\mathcal{O}}_B(d^2\,\tau)$, by Lemma 21. Thus, computing all the intersections is done in time $\widetilde{\mathcal{O}}_B(n\,d^2\,\tau)$. Moreover, sorting all these intersection points can be done in $\widetilde{\mathcal{O}}_B(\tau + d\,u)$ since the bitsize of their y coordinates is in $\mathcal{O}(\tau + d\,u)$ by Lemma 21.

We then compute the intersection between the $n - 1$ pairs of adjacent curves along the sweep line. Thus, we initially perform $\mathcal{O}(n)$ intersections between pairs of curves.

Then, during the sweep, every time an intersection point is encountered by the sweep line, we exchange two curves is the list of curves intersected by the sweep line and we compute the intersection between (at most) two new pairs of adjacent curves in this list. Hence, we perform, in total, $\mathcal{O}(n + k)$ intersections between pairs of curves in $\widetilde{\mathcal{O}}_B((n + k)\,d^2\,u(\tau + d\,u))$ time.

We now consider the cost of comparing the x-coordinates of the intersection points when updating the event list. Every time two curves become adjacent along the vertical line of sweep, we insert their first intersection point that is to the right of the line. Since we only insert one intersection point (rather than d), this requires in total $\mathcal{O}((n+k)\ \log n)$ comparisons which can be done in $\widetilde{\mathcal{O}}_B((n+k)\,d^2(\tau+u))$ time by Proposition 31.

□

Note finally that the same analysis, and thus the same complexity bound, applies also in the case of rational univariate functions.

6.3 Conclusion

We derived in this chapter output-sensitive bit-complexity analyses of the algorithm for computing the topology of algebraic planar curves introduced in Chapter 3 and of the standard sweep-line algorithm for computing arrangements of algebraic x-monotone curves defined by univariate polynomials.

As mentioned at the beginning of this chapter, bit-complexity analyses often introduce important overestimations due to worst-case considerations. Average-case complexity analyses are more difficult to perform in the bit-complexity model than on the real-RAM model. For instance, an average-case bit-complexity analysis would imply to consider expected values on the number of roots of the involved polynomials. These statistic analyses are known to be hard, see for example [100, 71]. More recently, Emiris et al. [67] used these results to study root separations, and the expected number of roots of Bernstein polynomials. These statistical analyses would permit to obtain tight average-case complexity bounds based on average-case separation bounds. However, it should be stressed that, even though such bounds would be of real interest, they would likely underestimate the complexity of the algorithms on real instances because the expected number of roots of random polynomials is very small [58], which tends not to be the case for input coming from real geometric problems.

Chapter 7

Conclusion

The main problem tackled in the thesis was the computation of the topology of plane curves, defined by bivariate polynomials with rational coefficients. We presented an algorithm based on Gröbner bases, rational univariate representations, univariate polynomial root isolations and interval arithmetic. These techniques permitted to avoid computations with algebraic numbers, and to be insensitive to the non-genericity of the curve.

We also addressed the implementation of this algorithm. First, we presented an implementation in MAPLE. Thorough experimentations and comparisons with other similar programs showed that our algorithm, and in its implementation, ISOTOP, permits to push forward the classes of curves that one can study. In particular, curves with many degeneracies, such as high-degree curves with many critical points with the same x-coordinate, represented a challenge to other algorithms, whether they shear the curve of not. Benchmarks thus validated our approach, as well as exposing weaknesses of our algorithm in some classes of examples, such as random curves. However, these cases are usually the simplest in other approaches. The study of the classes of curves in which ISOTOP is slow is currently leading to the development of a bivariate solver, that detect the cases where Gröbner bases or RUR computations are slow and solves the system using other techniques, see [25] for details.

It must be noted that our algorithm deals with single curves. More precisely, even if an input curve can be factorized, that is if it is the union of several algebraic curves, our algorithm considers it as a unique curve. However, as discussed previously, computing the topology of every (algebraic) component and combining them afterward using a sweep-line algorithm, is more efficient than directly computing the topology of the product. It should be stressed that this is the reason why we required that the x-extreme points of every curve should be computed in the original coordinate system. This problem of computing the topology of the arrangement of algebraic curves knowing their topologies was studied in detailed by Michael Kerber in his thesis[9] [104] (see also [61]). Another important related problem is the visualization of the curve from its topology. This problem has also been recently studied in Kerber's thesis (see also [64, 65]).

A natural extension of our algorithm is the development of an algorithm capable of handling curves in three dimensions. This assertion is founded in the fact that the algebraic techniques used in our algorithm in the plane extend directly to finding and identifying boxes containing all critical points of the curve without performing projections. Avoiding projections is important because it avoids the consideration of all the critical points of the projection which correspond to two branches of the curve that do not intersect in space but intersect in projection. However, determining the topology of the curve in space after computing such boxes remains an open problem.

Since MAPLE is not a standard for software development in the computational geometry community, we addressed the problem of the implementation of algebraic-based geometric algorithms in CGAL. We contributed an univariate algebraic kernel which constitute a first brick toward the completion of an algebraic framework in CGAL by implementing an univariate algebraic kernel. We presented extensive benchmarks for this algebraic kernel and other similar software, as well as an example of application of this algebraic kernel to the computation of arrangements of x-monotone curves defined by univariate integer polynomials.

Our algebraic kernel represents a step toward making CGAL able to handle general

[9]M. Kerber also addressed in his thesis the problem of computing the topology of a single curve.

algebraic non-linear objects. Future directions include the development of a bivariate algebraic kernel, which would permit to implement ISOTOP in CGAL. More generally, the development of an algebraic kernel capable of handling systems of polynomials in many variables would be an interesting challenge. However, such a kernel is not yet part of the CGAL specifications.

Throughout this thesis, experimentations helped considerably in the development process. Benchmarking exposed issues in implementations, providing means to choose which parts of the algorithm should be optimized, changed, or even rewritten. This happened, for instance, with the computation of the multiplicity of critical points in vertical fibers in the topology algorithm. It also showed the necessity of writing an efficient gcd function in the algebraic kernel. On the other hand, experimentation exposed flaws in some libraries used by our implementation (namely, it led to the discovery of a bug in the RS library, a bug in the 64-bit version of the CORE library and, of course, of many bugs in our implementations).

Finally, we presented output-sensitive bit-complexity analyses of the algorithm to determine the topology of a planar curve and of the standard sweep-line algorithm to compute arrangements of x-monotone curves defined by univariate polynomials. Algebraic techniques often need multiple-precision arithmetic, and the bit-complexity model of computation reflects the real cost of multiple-precision arithmetic operations better than the real-RAM model. However, output-sensitive analyses are usually hard and they are often skipped. Such analyses are important since they give tighter bounds than the standard worst-case complexity analyses when the algorithms depend on the size of the output (or other specific parameters such as separation bounds).

In a general context, we addressed in this thesis the problem of the computation of the topology of planar algebraic curves. We tackled this problem from three points of view, that are fundamental and complementary in computer science: algorithmic development, implementation and complexity analysis. In other words, implementation and benchmarking validated the approach taken in the design of the algorithm, while complexity analysis is theoretically interesting, since it provides another insight on the performance of algorithms.

Appendix A

Benchmarks

This appendix details the results of the experiments in the ISOTOP benchmarks, discussed in Section 4.4. These benchmarks produced large text files. We show in Section A.1 the result of parsing these files and eliminating some irrelevant data. Section A.2 shows the polynomial equations defining the tested curves.

A.1 Benchmark results

Each set of benchmarks is presented as a table. Each row of a table contain information about one curve. Columns of the table show information about the tested curve and the time spent for each implementation to compute its topology.

The first column, labelled $\mathscr{C}$, specifies the name of the curve. For reasons of readability, polynomials defining curves are not shown in the tables. They are presented in Section A.2.

The six columns of each table containing information about the curve are labelled as follows:

- d is the degree of the curve,
- τ is the bitsize of the curve,

- S is the number of singular points of the curve,
- E is the number of x-extreme points of the curve,
- V is the number of vertical lines in the curve, and
- A is the number of vertical asymptotes in the curve.

The remaining columns shows the time spent on the computation by each implementation, either total or partial. The column labeled IsoT shows the total ISOTOP computation time. The eight following columns contain a detail on the time spent by ISOTOP on each stage of the algorithm. These columns are labeled as follows:

- IB is the total time spent by ISOTOP to isolate the critical points in boxes,
- IG is the time spent by FGB to calculate the Gröbner bases,
- IR is the time spent by RS to calculate the RURs,
- IF is the time spent in refinement of boxes,
- IV is the time spent in processing the vertical lines,
- IA is the time spent in processing the vertical asymptotes,
- IS is the time spent in the sweep-line algorithm that computes the vertical subdivision of the plane in rectangles, and
- IC is the time spent in the connexion of the points.

The last five columns detail the time spent by the rest of the implementations, as follows:

- INS is the total time spent by INSULATE,
- TOP is the total time spent by TOP,
- CA0 is the total time spent by CA, optimized for generic curves by setting to 0 the parameter `CGAL_ACK_RESULTANT_FIRST_STRATEGY`,

- CA1 is the total time spent by CA, optimized for non-generic curves by setting to 1 the parameter `CGAL_ACK_RESULTANT_FIRST_STRATEGY`, and
- C2D is the time spent by CAD2D without considering initialization.

A.1.1 Benchmarks of the ACS and F series

The following table shows the results of the benchmarks of the curves contained in the ACS and F suites.

$\mathscr{C}$	d	τ	S	E	V	A	Isot	Ib	Ig	Ir	If	Iv	Ia	Is	Ic	Ins	Top	CA0	CA1	C2D
A_1	5	0	1	4	0	0	0.17	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.16	0.08	0.03	0.04	0.32
A_2	7	25	4	3	0	0	0.32	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.0	3.81	0.22	0.08	0.11	0.42
A_3	8	6	0	11	0	0	0.55	0.1	0.1	0.0	0.0	0.0	0.0	0.2	0.0	1.86	0.58	0.29	0.29	0.41
A_4	8	2	2	4	0	0	0.34	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.0	1.53	0.29	0.32	0.43	0.35
A_5	4	8	3	2	0	0	0.19	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.21	0.08	0.03	0.03	0.32
A_6	5	4	4	2	0	0	0.16	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.37	0.05	0.02	0.02	0.32
A_7	7	8	0	10	0	0	0.46	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.0	1.61	0.39	0.20	0.19	0.37
A_8	6	46	4	3	0	0	0.45	0.2	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.62	0.17	0.08	0.09	0.38
A_9	6	25	0	6	0	0	0.33	0.0	0.1	0.0	0.0	0.0	0.0	0.1	0.0	1.44	0.15	0.05	0.06	4.87
A_{10}	12	8	6	4	0	0	0.66	0.1	0.0	0.0	0.0	0.0	0.0	0.2	0.0	1.80	0.41	0.16	0.21	0.40
F_1	8	6	21	7	0	0	1.27	0.1	0.0	0.0	0.1	0.0	0.0	0.3	0.0	3.84	1.19	0.41	0.46	0.40
F_2	5	0	0	4	0	0	0.13	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.11	0.07	0.03	0.03	0.32
F_3	5	0	1	4	0	0	0.18	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.16	0.08	0.03	0.03	0.33
F_4	4	5	1	5	0	0	0.21	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.14	0.07	0.09	0.09	0.32
F_5	4	1	1	6	0	0	0.14	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.12	0.05	0.06	0.07	0.32
F_6	8	2	2	4	0	0	0.32	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.0	1.51	0.28	0.33	0.40	0.35
F_7	5	0	0	2	0	1	0.13	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.22	0.07	0.06	0.07	0.32
F_8	6	25	0	6	0	0	0.34	0.0	0.1	0.0	0.0	0.0	0.0	0.1	0.0	1.44	0.14	0.06	0.06	4.82
F_9	8	3	5	2	0	0	0.36	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.36	0.27	0.26	0.28	0.32
F_{10}	6	0	1	8	0	0	0.27	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.56	0.15	0.26	0.28	0.33
F_{11}	11	3	0	4	0	0	0.68	0.2	0.1	0.0	0.0	0.0	0.1	0.1	0.0	1.57	0.36	0.14	0.15	0.38
F_{12}	9	30	0	4	0	0	0.51	0.0	0.1	0.0	0.0	0.0	0.1	0.0	0.0	9.07	1.07	0.29	0.28	1.84
F_{13}	8	0	1	2	0	0	0.30	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.12	0.05	0.02	0.02	0.32
F_{14}	9	0	2	3	0	2	0.52	0.2	0.0	0.0	0.1	0.0	0.1	0.1	0.0	2.31	0.56	0.46	0.55	0.36
F_{15}	3	0	2	1	0	0	0.14	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.09	0.03	0.01	0.01	0.32
F_{16}	6	2	1	4	0	0	0.30	0.2	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.39	0.13	0.13	0.16	0.32
F_{17}	4	5	1	5	0	0	0.20	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.15	0.08	0.09	0.09	0.32

Benchmarks of the ACS and F series (continued on the next page).

$\mathscr{C}$	d	τ	S	E	V	A	ISOT	IB	IG	IR	IF	IV	IA	IS	IC	INS	TOP	CA0	CA1	C2D
F_{18}	6	104	2	8	0	0	0.40	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.0	2.06	0.41	0.31	0.36	0.36
F_{19}	8	21	4	18	0	0	1.00	0.1	0.0	0.0	0.0	0.0	0.0	0.3	0.0	3.38	1.59	1.02	1.10	0.44
F_{20}	8	3	5	8	0	0	0.50	0.1	0.0	0.0	0.0	0.0	0.0	0.2	0.0	1.76	0.47	0.44	0.52	1.40
F_{21}	8	23	0	14	0	0	0.78	0.0	0.1	0.0	0.1	0.0	0.0	0.2	0.0	2.92	0.75	0.78	0.88	0.42
F_{22}	8	23	0	16	0	0	0.71	0.0	0.1	0.0	0.0	0.0	0.0	0.2	0.0	3.70	0.82	0.90	0.94	0.41
F_{23}	8	20	9	8	0	0	0.94	0.1	0.0	0.0	0.1	0.0	0.1	0.2	0.0	6.32	0.87	0.82	0.92	ERR
F_{24}	16	22	9	8	0	0	4.08	0.6	0.4	0.0	0.3	0.0	1.0	0.6	0.0	243.27	64.04	12.20	15.40	ERR
F_{25}	9	36	0	4	0	0	0.55	0.1	0.2	0.0	0.0	0.0	0.0	0.0	0.0	11.49	1.21	0.36	0.36	2.16

Benchmarks of the ACS and F series.

A.1.2 Benchmarks on Labs' tough curves

Labs [109] gives a list of curves that are difficult to visualize. The following tables show the results of the benchmarks performed on those curves. The curves are named $L_{i,j}$, where i is the *challenge* number on the paper and j is a subindex.

$\mathscr{C}$	d	τ	S	E	V	A	ISOT	IB	IG	IR	IF	IV	IA	IS	IC	INS	TOP	CA0	CA1	C2D
$L_{1,1}$	4	3	3	1	0	0	0.12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.16	0.05	0.02	0.02	0.33
$L_{1,2}$	5	4	6	1	0	0	0.24	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.43	0.20	0.11	0.13	0.32
$L_{1,3}$	6	4	10	1	0	0	0.34	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.72	0.32	0.26	0.28	0.34
$L_{1,4}$	7	6	15	1	0	0	0.72	0.1	0.0	0.0	0.0	0.0	0.1	0.2	0.0	2.07	0.96	0.49	0.54	0.39
$L_{1,5}$	8	7	21	1	0	0	1.15	0.1	0.0	0.0	0.0	0.0	0.1	0.3	0.0	4.75	1.38	0.96	1.03	0.44
$L_{1,6}$	9	7	28	1	0	0	1.92	0.1	0.1	0.0	0.0	0.0	0.2	0.6	0.0	9.04	X	2.52	2.08	0.58
$L_{1,7}$	10	8	36	1	0	0	3.08	0.1	0.1	0.0	0.0	0.0	0.3	0.9	0.0	20.24	X	4.97	3.98	0.86

Benchmarks L_1.

$\mathscr{C}$	d	τ	S	E	V	A	ISOT	IB	IG	IR	IF	IV	IA	IS	IC	INS	TOP	CA0	CA1	C2D
$L_{2,1}$	2	0	1	0	0	0	0.07	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.01	0.00	0.00	0.32
$L_{2,2}$	4	0	1	0	0	0	0.08	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.33
$L_{2,3}$	6	0	1	0	0	0	0.08	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.32
$L_{2,4}$	8	0	1	0	0	0	0.07	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.32
$L_{2,5}$	10	0	1	0	0	0	0.08	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.01	0.32
$L_{2,6}$	12	0	1	0	0	0	0.08	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.01	0.91
$L_{2,7}$	14	0	1	0	0	0	0.07	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.01	0.33
$L_{2,8}$	4	0	1	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.02	0.00	0.00	0.33

Benchmarks L_2 (continued on the next page).

$\mathscr{C}$	d	τ	s	e	v	a	Isot	Ib	Ig	Ir	If	Iv	Ia	Is	Ic	Ins	Top	CA0	CA1	C2d
$L_{2,9}$	4	0	1	0	0	0	0.08	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.32
$L_{2,10}$	6	0	1	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.32
$L_{2,11}$	8	0	1	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.01	0.34
$L_{2,12}$	10	0	1	0	0	0	0.08	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.01	0.32
$L_{2,13}$	12	0	1	0	0	0	0.08	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.01	0.33
$L_{2,14}$	14	0	1	0	0	0	0.09	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.02	0.33
$L_{2,15}$	6	0	1	0	0	0	0.07	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.02	0.00	0.00	0.32
$L_{2,16}$	6	0	1	0	0	0	0.08	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.01	0.32
$L_{2,17}$	6	0	1	0	0	0	0.07	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.01	0.32
$L_{2,18}$	8	0	1	0	0	0	0.07	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.01	0.32
$L_{2,19}$	10	0	1	0	0	0	0.08	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.02	0.33
$L_{2,20}$	12	0	1	0	0	0	0.09	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.02	0.32
$L_{2,21}$	14	0	1	0	0	0	0.09	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.02	0.33
$L_{2,22}$	8	0	1	0	0	0	0.08	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.02	0.00	0.00	0.32
$L_{2,23}$	8	0	1	0	0	0	0.08	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.56
$L_{2,24}$	8	0	1	0	0	0	0.08	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.01	0.33
$L_{2,25}$	8	0	1	0	0	0	0.09	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.01	0.33
$L_{2,26}$	10	0	1	0	0	0	0.08	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.02	0.33
$L_{2,27}$	12	0	1	0	0	0	0.10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.02	0.33
$L_{2,28}$	14	0	1	0	0	0	0.09	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.02	0.04	0.35
$L_{2,29}$	10	0	1	0	0	0	0.07	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.02	0.00	0.00	0.34
$L_{2,30}$	10	0	1	0	0	0	0.08	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.32
$L_{2,31}$	10	0	1	0	0	0	0.07	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.01	0.32
$L_{2,32}$	10	0	1	0	0	0	0.08	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.02	0.33
$L_{2,33}$	10	0	1	0	0	0	0.09	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.03	0.33
$L_{2,34}$	12	0	1	0	0	0	0.11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.04	0.34
$L_{2,35}$	14	0	1	0	0	0	0.12	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.05	0.36
$L_{2,36}$	12	0	1	0	0	0	0.07	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.01	0.00	0.00	0.33
$L_{2,37}$	12	0	1	0	0	0	0.08	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.01	0.32
$L_{2,38}$	12	0	1	0	0	0	0.08	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.01	0.32
$L_{2,39}$	12	0	1	0	0	0	0.09	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.02	0.33
$L_{2,40}$	12	0	1	0	0	0	0.10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.04	0.34
$L_{2,41}$	12	0	1	0	0	0	0.13	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.05	0.32
$L_{2,42}$	14	0	1	0	0	0	0.15	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.02	0.07	0.38
$L_{2,43}$	14	0	1	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.01	0.00	0.00	0.32
$L_{2,44}$	14	0	1	0	0	0	0.08	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.01	0.32
$L_{2,45}$	14	0	1	0	0	0	0.10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.02	0.32
$L_{2,46}$	14	0	1	0	0	0	0.10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.03	0.34
$L_{2,47}$	14	0	1	0	0	0	0.12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.05	0.34

Benchmarks L_2 (continued on the next page).

$\mathscr{C}$	d	τ	S	E	V	A	ISOT	IB	IG	IR	IF	IV	IA	IS	IC	INS	TOP	CA0	CA1	C2D
$L_{2,48}$	14	0	1	0	0	0	0.16	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.06	0.36
$L_{2,49}$	14	0	1	0	0	0	0.20	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	INT	ERR	0.02	0.09	0.38
$L_{2,50}$	8	0	1	0	0	0	0.12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.03	0.01	0.03	0.33
$L_{2,51}$	16	3	1	0	0	0	0.37	0.0	0.2	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.04	0.12	0.50	0.64
$L_{2,52}$	24	4	1	0	0	0	2.73	0.0	2.6	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.05	0.88	4.29	0.78
$L_{2,53}$	32	7	1	0	0	0	13.52	0.0	13.3	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.06	4.24	24.30	18.19
$L_{2,54}$	40	8	1	0	0	0	47.70	0.0	47.4	0.0	0.0	0.0	0.0	0.1	0.0	ERR	0.08	15.70	104.00	23.50
$L_{2,55}$	10	0	1	0	0	0	0.14	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.04	0.03	0.06	0.34
$L_{2,56}$	20	3	1	0	0	0	0.97	0.0	0.8	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.05	0.32	1.28	1.16
$L_{2,57}$	30	4	1	0	0	0	9.19	0.0	9.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.06	2.22	12.50	1.55
$L_{2,58}$	40	7	1	0	0	0	47.45	0.0	47.1	0.0	0.0	0.0	0.0	0.1	0.0	ERR	0.07	11.20	78.50	55.48
$L_{2,59}$	50	8	1	0	0	0	174.61	0.1	174.2	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.09	43.30	353.00	67.57
$L_{2,60}$	12	0	1	0	0	0	0.22	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.04	0.06	0.14	0.39
$L_{2,61}$	24	3	1	0	0	0	2.65	0.0	2.5	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.04	0.64	2.96	2.49
$L_{2,62}$	36	4	1	0	0	0	25.74	0.0	25.4	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.06	4.96	31.80	3.35
$L_{2,63}$	48	7	1	0	0	0	138.48	0.0	138.0	0.0	0.0	0.0	0.1	0.0	0.0	ERR	0.08	23.60	206.00	134.40
$L_{2,64}$	60	8	1	0	0	0	514.90	0.0	514.3	0.0	0.0	0.0	0.0	0.0	0.0	INT	0.08	89.10	937.00	165.65
$L_{2,65}$	14	0	1	0	0	0	0.31	0.0	0.2	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.04	0.12	0.25	3.72
$L_{2,66}$	28	3	1	0	0	0	6.24	0.0	6.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.05	1.36	6.30	6.84
$L_{2,67}$	42	4	1	0	0	0	62.96	0.0	62.6	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.06	10.10	72.70	ERR
$L_{2,68}$	56	7	1	0	0	0	341.61	0.0	341.1	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.10	46.90	479.00	292.96
$L_{2,69}$	70	8	1	0	0	0	1285.41	0.0	1284.7	0.0	0.0	0.0	0.1	0.1	0.0	INT	0.12	170.00	2200.00	ERR
$L_{2,70}$	16	0	1	0	0	0	0.56	0.0	0.3	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.02	0.22	0.51	0.60
$L_{2,71}$	32	3	1	0	0	0	13.15	0.0	12.8	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.05	2.71	12.60	11.31
$L_{2,72}$	48	4	1	0	0	0	137.10	0.0	136.6	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.08	19.00	145.00	18.90
$L_{2,73}$	64	7	1	0	0	0	759.13	0.0	758.3	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.10	83.60	989.00	563.50
$L_{2,74}$	80	8	1	0	0	0	2829.81	0.0	2828.4	0.0	0.0	0.0	0.1	0.1	0.0	X	0.16	298.00	4560.00	ERR
$L_{2,75}$	18	0	1	0	0	0	0.78	0.0	0.6	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.04	0.42	0.81	0.84
$L_{2,76}$	36	3	1	0	0	0	25.82	0.0	25.4	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.05	5.10	23.30	18.84
$L_{2,77}$	54	4	1	0	0	0	275.25	0.0	274.7	0.0	0.0	0.0	0.0	0.1	0.0	ERR	0.08	31.40	277.00	26.27
$L_{2,78}$	72	7	1	0	0	0	1537.19	0.0	1536.3	0.0	0.0	0.0	0.0	0.1	0.0	INT	0.10	144.00	1890.00	ERR
$L_{2,79}$	90	8	1	0	0	0	ERR	ERR	ERR	ERR	ERR	ERR	ERR	ERR	ERR	X	0.18	507.00	8590.00	ERR
$L_{2,80}$	20	0	1	0	0	0	1.40	0.0	1.1	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.04	0.78	1.49	1.15
$L_{2,81}$	40	3	1	0	0	0	47.86	0.0	47.3	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.06	8.69	40.90	33.64
$L_{2,82}$	60	4	1	0	0	0	523.52	0.0	522.6	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.08	54.90	501.00	43.38
$L_{2,83}$	80	7	1	0	0	0	2906.49	0.0	2905.0	0.0	0.0	0.0	0.1	0.1	0.0	INT	0.14	249.00	3360.00	ERR

Benchmarks L_2.

$\mathscr{C}$	d	τ	S	E	V	A	ISOT	IB	IG	IR	IF	IV	IA	IS	IC	INS	TOP	CA0	CA1	C2D
$L_{3,1}$	4	5	0	8	0	0	0.18	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.25	0.08	0.04	0.04	0.33
$L_{3,2}$	4	10	0	9	0	0	0.19	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.23	0.10	0.05	0.05	0.34
$L_{3,3}$	4	20	0	9	0	0	0.22	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.47	0.11	0.05	0.06	0.33
$L_{3,4}$	4	34	0	9	0	0	0.23	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.51	0.12	0.06	0.06	0.32
$L_{3,5}$	4	50	0	9	0	0	0.26	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.51	0.12	0.06	0.06	0.32
$L_{3,6}$	5	4	0	14	0	0	0.35	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.55	0.22	0.11	0.11	0.34
$L_{3,7}$	5	10	0	15	0	0	0.37	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.59	0.23	0.12	0.13	0.35
$L_{3,8}$	5	20	0	16	0	0	0.38	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.60	0.27	0.14	0.14	0.35
$L_{3,9}$	5	34	0	16	0	0	0.43	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.0	1.73	0.30	0.14	0.14	0.36
$L_{3,10}$	5	50	0	16	0	0	0.42	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.0	1.82	0.31	0.16	0.15	0.37
$L_{3,11}$	6	6	0	22	0	0	0.65	0.1	0.0	0.0	0.0	0.0	0.0	0.2	0.0	1.82	0.45	0.26	0.26	0.38
$L_{3,12}$	6	12	0	23	0	0	0.71	0.1	0.0	0.0	0.0	0.0	0.0	0.2	0.0	1.94	0.55	0.26	0.28	0.39
$L_{3,13}$	6	22	0	24	0	0	0.78	0.1	0.0	0.0	0.0	0.0	0.0	0.2	0.0	2.10	0.64	0.28	0.29	0.41
$L_{3,14}$	6	36	0	25	0	0	0.98	0.1	0.0	0.0	0.1	0.0	0.0	0.3	0.0	2.27	0.75	0.31	0.34	27.15
$L_{3,15}$	6	52	0	25	0	0	0.87	0.1	0.1	0.0	0.0	0.0	0.0	0.3	0.0	5.73	0.86	0.34	0.33	63.75
$L_{3,16}$	7	7	0	32	0	0	1.13	0.1	0.0	0.0	0.0	0.0	0.1	0.4	0.0	3.33	0.98	0.46	0.50	0.44
$L_{3,17}$	7	14	0	33	0	0	1.24	0.1	0.1	0.0	0.0	0.0	0.1	0.4	0.0	3.64	1.18	0.49	0.52	0.47
$L_{3,18}$	7	24	0	34	0	0	1.40	0.2	0.1	0.0	0.0	0.0	0.1	0.4	0.0	3.95	1.38	0.53	0.54	0.50
$L_{3,19}$	7	37	0	35	0	0	1.86	0.2	0.1	0.0	0.2	0.0	0.2	0.5	0.0	4.47	1.72	0.58	0.60	182.52
$L_{3,20}$	7	54	0	36	0	0	2.19	0.2	0.1	0.0	0.5	0.0	0.2	0.5	0.0	4.96	2.05	0.64	0.64	PRIM
$L_{3,21}$	8	7	0	44	0	0	2.32	0.2	0.2	0.0	0.2	0.0	0.1	0.7	0.0	14.59	2.05	0.89	0.89	0.66
$L_{3,22}$	8	11	0	45	0	0	2.34	0.2	0.2	0.0	0.0	0.0	0.1	0.8	0.0	14.98	2.41	0.89	0.96	0.65
$L_{3,23}$	8	21	0	46	0	0	2.60	0.3	0.2	0.0	0.0	0.0	0.2	0.8	0.0	15.58	2.95	0.94	0.99	363.05
$L_{3,24}$	8	34	0	47	0	0	3.65	0.3	0.4	0.0	0.6	0.0	0.2	0.8	0.0	17.05	3.65	1.09	1.07	2.13
$L_{3,25}$	8	51	0	48	0	0	4.95	0.4	0.7	0.0	1.3	0.0	0.3	0.9	0.0	18.42	4.49	1.22	1.19	2.37
$L_{3,26}$	9	7	0	58	0	0	3.68	0.3	0.3	0.0	0.0	0.0	0.2	1.2	0.0	21.68	3.99	1.52	1.70	0.84
$L_{3,27}$	9	14	0	59	0	0	3.92	0.4	0.4	0.0	0.0	0.0	0.2	1.3	0.0	22.66	4.86	1.61	1.70	3.06
$L_{3,28}$	9	24	0	60	0	0	4.39	0.4	0.6	0.0	0.0	0.0	0.3	1.3	0.0	23.59	6.20	1.75	1.78	3.10
$L_{3,29}$	9	37	0	61	0	0	6.24	0.6	1.0	0.0	0.9	0.0	0.4	1.3	0.0	24.98	7.74	1.80	1.94	3.59
$L_{3,30}$	9	54	0	62	0	0	8.45	0.7	1.5	0.0	2.0	0.0	0.5	1.4	0.0	27.52	10.08	2.00	2.30	3.68
$L_{3,31}$	9	29	0	65	0	0	7.06	0.8	1.9	0.0	0.0	0.0	0.2	2.0	0.0	33.22	14.01	3.42	3.61	3.38
$L_{3,32}$	9	35	0	66	0	0	7.50	0.8	2.1	0.0	0.0	0.0	0.3	2.0	0.0	33.97	X	2.92	3.14	3.92
$L_{3,33}$	9	45	0	67	0	0	8.13	0.8	2.4	0.0	0.0	0.0	0.4	2.1	0.0	37.29	593.87	3.08	3.20	3.93
$L_{3,34}$	9	59	0	68	0	0	10.70	0.9	2.9	0.0	1.5	0.0	0.5	2.3	0.0	39.31	X	3.70	3.51	4.26
$L_{3,35}$	9	75	0	69	0	0	13.47	0.9	3.6	0.0	3.0	0.0	0.6	2.3	0.0	39.00	X	3.23	3.41	4.76

Benchmarks L_3.

$\mathscr{C}$	d	τ	s	e	v	a	Isot	Ib	Ig	Ir	If	Iv	Ia	Is	Ic	Ins	Top	CA0	CA1	C2d
$L_{4,1}$	4	11	0	4	0	0	0.10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.10	0.06	0.02	0.02	0.32
$L_{4,2}$	4	21	0	4	0	0	0.13	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.11	0.06	0.02	0.02	0.32
$L_{4,3}$	4	11	0	4	0	0	0.12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.12	0.06	0.02	0.02	0.33
$L_{4,4}$	4	21	0	4	0	0	0.12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.13	0.07	0.02	0.02	0.33
$L_{4,5}$	4	15	0	4	0	0	0.12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.14	0.06	0.02	0.02	0.33
$L_{4,6}$	4	21	0	4	0	0	0.14	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.15	0.06	0.02	0.02	0.33
$L_{4,7}$	6	22	0	4	0	0	0.14	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.16	0.08	0.03	0.04	0.34
$L_{4,8}$	6	42	0	4	0	0	0.14	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.18	0.08	0.03	0.04	0.34
$L_{4,9}$	6	22	0	4	0	0	0.16	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.17	0.08	0.03	0.04	0.34
$L_{4,10}$	6	42	0	4	0	0	0.15	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.21	0.08	0.03	0.04	0.35
$L_{4,11}$	6	22	0	6	0	0	0.18	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.29	0.12	0.04	0.05	0.34
$L_{4,12}$	6	42	0	4	0	0	0.17	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.25	0.09	0.03	0.04	0.35
$L_{4,13}$	8	36	0	2	0	0	0.20	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.27	0.13	0.02	0.06	0.39
$L_{4,14}$	8	70	0	4	0	0	0.30	0.1	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.49	0.20	0.06	0.09	0.45
$L_{4,15}$	8	36	0	4	0	0	0.25	0.1	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.38	0.16	0.04	0.08	0.39
$L_{4,16}$	8	70	0	4	0	0	0.30	0.1	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.51	0.20	0.05	0.09	0.45
$L_{4,17}$	8	36	0	4	0	0	0.26	0.1	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.44	0.16	0.04	0.08	0.39
$L_{4,18}$	8	70	0	4	0	0	0.30	0.1	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.56	0.22	0.05	0.08	0.46
$L_{4,19}$	10	54	0	2	0	0	0.33	0.1	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.51	0.36	0.05	0.14	0.59
$L_{4,20}$	10	103	0	4	0	0	0.53	0.1	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.96	0.68	0.10	0.25	0.85
$L_{4,21}$	10	54	0	4	0	0	0.38	0.1	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.67	0.41	0.08	0.17	0.60
$L_{4,22}$	10	103	0	4	0	0	0.52	0.1	0.0	0.0	0.0	0.0	0.2	0.0	0.0	1.00	0.65	0.10	0.26	0.86
$L_{4,23}$	10	54	0	4	0	0	0.41	0.1	0.0	0.0	0.0	0.0	0.1	0.1	0.0	0.73	0.41	0.07	0.17	0.61
$L_{4,24}$	10	103	0	4	0	0	0.54	0.1	0.0	0.0	0.0	0.0	0.2	0.1	0.0	1.06	0.66	0.10	0.26	0.85
$L_{4,25}$	12	75	0	2	0	0	0.66	0.3	0.1	0.0	0.0	0.0	0.2	0.0	0.0	0.91	1.24	0.12	0.40	1.23
$L_{4,26}$	12	144	0	4	0	0	1.25	0.5	0.1	0.0	0.0	0.0	0.4	0.1	0.0	1.88	2.50	0.26	0.84	2.15
$L_{4,27}$	12	75	0	4	0	0	0.72	0.2	0.1	0.0	0.0	0.0	0.2	0.1	0.0	1.16	1.27	0.16	0.43	1.23
$L_{4,28}$	12	144	0	4	0	0	1.25	0.5	0.1	0.0	0.0	0.0	0.4	0.1	0.0	1.96	2.48	0.26	0.84	2.18
$L_{4,29}$	12	75	0	4	0	0	0.72	0.2	0.1	0.0	0.0	0.0	0.2	0.1	0.0	1.28	1.28	0.16	0.42	1.24
$L_{4,30}$	12	144	0	4	0	0	1.27	0.5	0.1	0.0	0.0	0.0	0.4	0.1	0.0	2.10	2.48	0.25	0.83	2.18

Benchmarks L_4.

$\mathscr{C}$	d	τ	s	e	v	a	Isot	Ib	Ig	Ir	If	Iv	Ia	Is	Ic	Ins	Top	CA0	CA1	C2d
$L_{5,1}$	2	4	0	0	0	0	0.04	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.01	0.00	0.00	0.32
$L_{5,2}$	2	7	0	0	0	0	0.05	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.02	0.00	0.00	0.35
$L_{5,3}$	2	10	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.01	0.00	0.00	0.33
$L_{5,4}$	2	14	0	0	0	0	0.04	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.02	0.00	0.00	0.32
$L_{5,5}$	4	4	0	0	0	0	0.05	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.32

Benchmarks L_5 (continued on the next page).

$\mathscr{C}$	d	τ	S	E	V	A	Isot	Ib	Ig	Ir	If	Iv	Ia	Is	Ic	Ins	Top	CA0	CA1	C2d
$L_{5,6}$	4	7	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.32
$L_{5,7}$	4	10	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.32
$L_{5,8}$	4	14	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.32
$L_{5,9}$	6	4	0	0	0	0	0.05	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.33
$L_{5,10}$	6	7	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.32
$L_{5,11}$	6	10	0	0	0	0	0.07	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.32
$L_{5,12}$	6	14	0	0	0	0	0.05	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.32
$L_{5,13}$	8	4	0	0	0	0	0.04	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.32
$L_{5,14}$	8	7	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.32
$L_{5,15}$	8	10	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.32
$L_{5,16}$	8	14	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.33
$L_{5,17}$	10	4	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.32
$L_{5,18}$	10	7	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.32
$L_{5,19}$	10	10	0	0	0	0	0.07	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.32
$L_{5,20}$	10	14	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.33
$L_{5,21}$	12	4	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.32
$L_{5,22}$	12	7	0	0	0	0	0.07	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.32
$L_{5,23}$	12	10	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.01	0.33
$L_{5,24}$	12	14	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.33
$L_{5,25}$	4	4	0	0	0	0	0.05	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.02	0.00	0.00	0.34
$L_{5,26}$	4	7	0	0	0	0	0.04	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.02	0.00	0.00	0.32
$L_{5,27}$	4	10	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.02	0.00	0.00	0.33
$L_{5,28}$	4	14	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.01	0.00	0.00	0.32
$L_{5,29}$	4	4	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.32
$L_{5,30}$	4	7	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.32
$L_{5,31}$	4	10	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.33
$L_{5,32}$	4	14	0	0	0	0	0.05	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.32
$L_{5,33}$	6	4	0	0	0	0	0.05	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.32
$L_{5,34}$	6	7	0	0	0	0	0.05	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.32
$L_{5,35}$	6	10	0	0	0	0	0.07	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.32
$L_{5,36}$	6	14	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.01	0.33
$L_{5,37}$	8	4	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.32
$L_{5,38}$	8	7	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.33
$L_{5,39}$	8	10	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.33
$L_{5,40}$	8	14	0	0	0	0	0.08	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.32
$L_{5,41}$	10	4	0	0	0	0	0.07	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.01	0.33
$L_{5,42}$	10	7	0	0	0	0	0.05	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.01	0.33
$L_{5,43}$	10	10	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.33
$L_{5,44}$	10	14	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.01	0.34

Benchmarks L_5 (continued on the next page).

$\mathscr{C}$	d	τ	S	E	V	A	Isot	Ib	Ig	Ir	If	Iv	Ia	Is	Ic	Ins	Top	CA0	CA1	C2d
$L_{5,45}$	12	4	0	0	0	0	0.07	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.01	0.33
$L_{5,46}$	12	7	0	0	0	0	0.07	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.01	0.35
$L_{5,47}$	12	10	0	0	0	0	0.09	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.01	0.34
$L_{5,48}$	12	14	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.01	0.34
$L_{5,49}$	6	4	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.02	0.00	0.00	0.61
$L_{5,50}$	6	7	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.02	0.00	0.00	0.32
$L_{5,51}$	6	10	0	0	0	0	0.05	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.02	0.00	0.00	0.32
$L_{5,52}$	6	14	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.02	0.00	0.00	0.32
$L_{5,53}$	6	4	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.32
$L_{5,54}$	6	7	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.32
$L_{5,55}$	6	10	0	0	0	0	0.07	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.32
$L_{5,56}$	6	14	0	0	0	0	0.05	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.32
$L_{5,57}$	6	4	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.32
$L_{5,58}$	6	7	0	0	0	0	0.05	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.01	0.32
$L_{5,59}$	6	10	0	0	0	0	0.07	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.33
$L_{5,60}$	6	14	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.33
$L_{5,61}$	8	4	0	0	0	0	0.07	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.01	0.32
$L_{5,62}$	8	7	0	0	0	0	0.08	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.01	0.33
$L_{5,63}$	8	10	0	0	0	0	0.07	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.01	0.33
$L_{5,64}$	8	14	0	0	0	0	0.05	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.01	0.34
$L_{5,65}$	10	4	0	0	0	0	0.08	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.01	0.33
$L_{5,66}$	10	7	0	0	0	0	0.07	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.01	0.34
$L_{5,67}$	10	10	0	0	0	0	0.07	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.02	0.34
$L_{5,68}$	10	14	0	0	0	0	0.08	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.01	0.34
$L_{5,69}$	12	4	0	0	0	0	0.09	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.02	0.34
$L_{5,70}$	12	7	0	0	0	0	0.09	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.01	0.34
$L_{5,71}$	12	10	0	0	0	0	0.09	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.01	0.36
$L_{5,72}$	12	14	0	0	0	0	0.09	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.02	0.37
$L_{5,73}$	8	4	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.03	0.00	0.00	0.32
$L_{5,74}$	8	7	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.02	0.00	0.00	0.32
$L_{5,75}$	8	10	0	0	0	0	0.07	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.02	0.00	0.00	0.32
$L_{5,76}$	8	14	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.02	0.00	0.00	0.32
$L_{5,77}$	8	4	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.32
$L_{5,78}$	8	7	0	0	0	0	0.05	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.33
$L_{5,79}$	8	10	0	0	0	0	0.05	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.33
$L_{5,80}$	8	14	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.32
$L_{5,81}$	8	4	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.01	0.32
$L_{5,82}$	8	7	0	0	0	0	0.05	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.01	0.32
$L_{5,83}$	8	10	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.01	0.33

Benchmarks L_5 (continued on the next page).

$\mathscr{C}$	d	τ	S	E	V	A	Isot	Ib	Ig	Ir	If	Iv	Ia	Is	Ic	Ins	Top	CA0	CA1	C2d
$L_{5,84}$	8	14	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.01	0.33
$L_{5,85}$	8	4	0	0	0	0	0.08	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.01	0.33
$L_{5,86}$	8	7	0	0	0	0	0.09	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.01	0.33
$L_{5,87}$	8	10	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.01	0.34
$L_{5,88}$	8	14	0	0	0	0	0.08	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.02	0.34
$L_{5,89}$	10	4	0	0	0	0	0.09	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.02	0.34
$L_{5,90}$	10	7	0	0	0	0	0.09	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.02	0.34
$L_{5,91}$	10	10	0	0	0	0	0.10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.02	0.36
$L_{5,92}$	10	14	0	0	0	0	0.09	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.02	0.37
$L_{5,93}$	12	4	0	0	0	0	0.12	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.02	0.36
$L_{5,94}$	12	7	0	0	0	0	0.12	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.02	0.36
$L_{5,95}$	12	10	0	0	0	0	0.12	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.03	0.38
$L_{5,96}$	12	14	0	0	0	0	0.10	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.03	0.41
$L_{5,97}$	10	4	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.02	0.00	0.00	0.32
$L_{5,98}$	10	7	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.02	0.00	0.00	0.33
$L_{5,99}$	10	10	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.02	0.00	0.00	0.33
$L_{5,100}$	10	14	0	0	0	0	0.03	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.02	0.00	0.00	0.32
$L_{5,101}$	10	4	0	0	0	0	0.07	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.32
$L_{5,102}$	10	7	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.32
$L_{5,103}$	10	10	0	0	0	0	0.07	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.32
$L_{5,104}$	10	14	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.01	0.32
$L_{5,105}$	10	4	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.01	0.33
$L_{5,106}$	10	7	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.01	0.33
$L_{5,107}$	10	10	0	0	0	0	0.07	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.01	0.34
$L_{5,108}$	10	14	0	0	0	0	0.08	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.01	0.34
$L_{5,109}$	10	4	0	0	0	0	0.09	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.02	0.33
$L_{5,110}$	10	7	0	0	0	0	0.10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.02	0.34
$L_{5,111}$	10	10	0	0	0	0	0.08	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.02	0.35
$L_{5,112}$	10	14	0	0	0	0	0.07	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.02	0.36
$L_{5,113}$	10	4	0	0	0	0	0.11	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.02	0.34
$L_{5,114}$	10	7	0	0	0	0	0.10	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.02	0.36
$L_{5,115}$	10	10	0	0	0	0	0.11	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.03	0.38
$L_{5,116}$	10	14	0	0	0	0	0.13	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.03	0.39
$L_{5,117}$	12	4	0	0	0	0	0.15	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.03	0.36
$L_{5,118}$	12	7	0	0	0	0	0.16	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.03	0.40
$L_{5,119}$	12	10	0	0	0	0	0.16	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.04	0.41
$L_{5,120}$	12	14	0	0	0	0	0.16	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.02	0.04	0.44
$L_{5,121}$	12	4	0	0	0	0	0.05	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.02	0.00	0.00	0.32
$L_{5,122}$	12	7	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.02	0.00	0.00	0.32

Benchmarks L_5 (continued on the next page).

$\mathscr{C}$	d	τ	S	E	V	A	Isot	Ib	Ig	Ir	If	Iv	Ia	Is	Ic	Ins	Top	CA0	CA1	C2d
$L_{5,123}$	12	10	0	0	0	0	0.04	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.02	0.00	0.00	0.32
$L_{5,124}$	12	14	0	0	0	0	0.05	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.03	0.00	0.00	0.32
$L_{5,125}$	12	4	0	0	0	0	0.08	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.32
$L_{5,126}$	12	7	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.33
$L_{5,127}$	12	10	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.01	0.32
$L_{5,128}$	12	14	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.00	0.33
$L_{5,129}$	12	4	0	0	0	0	0.09	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.01	0.32
$L_{5,130}$	12	7	0	0	0	0	0.08	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.00	0.01	0.32
$L_{5,131}$	12	10	0	0	0	0	0.08	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.01	0.33
$L_{5,132}$	12	14	0	0	0	0	0.08	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.02	0.38
$L_{5,133}$	12	4	0	0	0	0	0.12	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.02	0.33
$L_{5,134}$	12	7	0	0	0	0	0.10	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.02	0.35
$L_{5,135}$	12	10	0	0	0	0	0.11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.02	0.36
$L_{5,136}$	12	14	0	0	0	0	0.10	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.02	0.36
$L_{5,137}$	12	4	0	0	0	0	0.16	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.04	0.36
$L_{5,138}$	12	7	0	0	0	0	0.16	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.04	0.37
$L_{5,139}$	12	10	0	0	0	0	0.16	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.03	0.40
$L_{5,140}$	12	14	0	0	0	0	0.16	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.04	0.41
$L_{5,141}$	12	4	0	0	0	0	0.23	0.0	0.2	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.04	0.38
$L_{5,142}$	12	7	0	0	0	0	0.24	0.0	0.2	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.05	0.42
$L_{5,143}$	12	10	0	0	0	0	0.22	0.0	0.2	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.02	0.05	0.45
$L_{5,144}$	12	14	0	0	0	0	0.22	0.0	0.2	0.0	0.0	0.0	0.0	0.0	0.0	ERR	ERR	0.01	0.06	0.49
$L_{5,145}$	2	5	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.01	0.00	0.00	0.32
$L_{5,146}$	2	8	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.02	0.00	0.00	0.32
$L_{5,147}$	2	11	0	0	0	0	0.05	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.02	0.00	0.00	0.32
$L_{5,148}$	2	15	0	0	0	0	0.05	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.01	0.00	0.00	0.32
$L_{5,149}$	4	6	0	0	0	0	0.05	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.03	0.01	0.02	0.33
$L_{5,150}$	4	10	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.03	0.01	0.02	0.32
$L_{5,151}$	4	13	0	0	0	0	0.08	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.04	0.01	0.01	0.32
$L_{5,152}$	4	16	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.03	0.01	0.01	0.32
$L_{5,153}$	6	8	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.03	0.02	0.03	0.33
$L_{5,154}$	6	11	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.04	0.03	0.03	0.33
$L_{5,155}$	6	15	0	0	0	0	0.08	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.04	0.08	0.04	0.33
$L_{5,156}$	6	18	0	0	0	0	0.07	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.05	0.02	0.04	0.34
$L_{5,157}$	4	5	0	0	0	0	0.07	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.02	0.00	0.00	0.32
$L_{5,158}$	4	8	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.03	0.00	0.00	0.32
$L_{5,159}$	4	11	0	0	0	0	0.05	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.03	0.00	0.00	0.32
$L_{5,160}$	4	15	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.02	0.00	0.00	0.32
$L_{5,161}$	8	6	0	0	0	0	0.10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.02	0.04	0.06	0.35

Benchmarks L_5 (continued on the next page).

$\mathscr{C}$	d	τ	S	E	V	A	ISOT	IB	IG	IR	IF	IV	IA	IS	IC	INS	TOP	CA0	CA1	C2D
$L_{5,162}$	8	10	0	0	0	0	0.12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.04	0.04	0.07	0.36
$L_{5,163}$	8	13	0	0	0	0	0.11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.03	0.04	0.06	0.36
$L_{5,164}$	8	16	0	0	0	0	0.10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.03	0.04	0.06	0.38
$L_{5,165}$	12	8	0	0	0	0	0.44	0.0	0.4	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.04	0.39	0.47	0.64
$L_{5,166}$	12	11	0	0	0	0	0.44	0.0	0.4	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.04	0.35	0.46	0.69
$L_{5,167}$	12	15	0	0	0	0	0.49	0.0	0.4	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.05	0.38	0.48	0.70
$L_{5,168}$	12	18	0	0	0	0	0.51	0.0	0.4	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.06	0.35	0.46	0.79
$L_{5,169}$	6	5	0	0	0	0	0.08	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.02	0.01	0.02	0.33
$L_{5,170}$	6	8	0	0	0	0	0.08	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.03	0.01	0.02	0.33
$L_{5,171}$	6	11	0	0	0	0	0.08	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.02	0.01	0.02	0.33
$L_{5,172}$	6	15	0	0	0	0	0.07	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.02	0.01	0.02	0.33
$L_{5,173}$	12	6	0	0	0	0	0.40	0.0	0.3	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.04	0.35	0.40	0.62
$L_{5,174}$	12	10	0	0	0	0	0.42	0.0	0.3	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.04	0.32	0.42	0.70
$L_{5,175}$	12	13	0	0	0	0	0.44	0.0	0.3	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.05	0.32	0.42	0.69
$L_{5,176}$	12	16	0	0	0	0	0.45	0.0	0.4	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.04	0.39	0.44	0.71
$L_{5,177}$	18	8	0	0	0	0	3.60	0.0	3.4	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.10	3.81	4.44	3.36
$L_{5,178}$	18	11	0	0	0	0	3.79	0.0	3.5	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.14	3.91	4.57	2.54
$L_{5,179}$	18	15	0	0	0	0	3.77	0.0	3.6	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.11	3.74	4.39	3.16
$L_{5,180}$	18	18	0	0	0	0	4.04	0.0	3.7	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.18	3.82	4.62	4.16
$L_{5,181}$	8	5	0	0	0	0	0.10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.03	0.03	0.04	0.34
$L_{5,182}$	8	8	0	0	0	0	0.10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.03	0.04	0.05	0.35
$L_{5,183}$	8	11	0	0	0	0	0.09	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.03	0.04	0.04	0.35
$L_{5,184}$	8	15	0	0	0	0	0.12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.03	0.03	0.05	0.36
$L_{5,185}$	16	6	0	0	0	0	1.92	0.0	1.8	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.04	2.04	2.10	1.50
$L_{5,186}$	16	10	0	0	0	0	1.98	0.0	1.9	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.08	1.78	2.05	1.93
$L_{5,187}$	16	13	0	0	0	0	2.12	0.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.08	1.80	2.18	1.74
$L_{5,188}$	16	16	0	0	0	0	2.07	0.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.06	1.71	2.06	2.34
$L_{5,189}$	24	8	0	0	0	0	19.40	0.0	17.8	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.72	27.09	28.41	42.13
$L_{5,190}$	24	11	0	0	0	0	18.16	0.0	18.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.14	24.02	27.12	13.71
$L_{5,191}$	24	15	0	0	0	0	19.39	0.0	18.3	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.55	26.91	30.56	15.99
$L_{5,192}$	24	18	0	0	0	0	19.31	0.0	18.8	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.34	24.37	27.46	17.76
$L_{5,193}$	10	5	0	0	0	0	0.18	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.04	0.12	0.16	0.41
$L_{5,194}$	10	8	0	0	0	0	0.19	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.02	0.13	0.14	0.46
$L_{5,195}$	10	11	0	0	0	0	0.18	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.03	0.10	0.13	0.45
$L_{5,196}$	10	15	0	0	0	0	0.20	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.03	0.10	0.14	0.44
$L_{5,197}$	20	6	0	0	0	0	6.32	0.0	5.6	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.29	7.52	8.18	5.34
$L_{5,198}$	20	10	0	0	0	0	6.06	0.0	5.8	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.12	7.27	8.16	4.42
$L_{5,199}$	20	13	0	0	0	0	6.48	0.0	5.9	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.22	7.94	8.28	5.59
$L_{5,200}$	20	16	0	0	0	0	6.25	0.0	6.1	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.08	7.62	8.41	5.26

Benchmarks L_5 (continued on the next page).

$\mathscr{C}$	d	τ	S	E	V	A	Isot	Ib	Ig	Ir	If	Iv	Ia	Is	Ic	Ins	Top	CA0	CA1	C2d
$L_{5,201}$	30	8	0	0	0	0	73.20	0.0	71.6	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.72	123.65	134.61	63.25
$L_{5,202}$	30	11	0	0	0	0	73.55	0.0	72.1	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.70	122.00	129.66	117.97
$L_{5,203}$	30	15	0	0	0	0	74.54	0.0	73.2	0.0	0.0	0.0	0.0	0.0	0.0	X	0.69	127.74	132.31	59.91
$L_{5,204}$	30	18	0	0	0	0	75.48	0.0	74.9	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.45	120.74	131.57	62.97
$L_{5,205}$	12	5	0	0	0	0	0.41	0.0	0.3	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.04	0.34	0.41	0.61
$L_{5,206}$	12	8	0	0	0	0	0.37	0.0	0.3	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.04	0.30	0.38	0.54
$L_{5,207}$	12	11	0	0	0	0	0.42	0.0	0.4	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.03	0.35	0.42	0.66
$L_{5,208}$	12	15	0	0	0	0	0.42	0.0	0.3	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.04	0.28	0.36	0.65
$L_{5,209}$	24	6	0	0	0	0	21.84	0.0	20.9	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.40	24.65	26.48	11.05
$L_{5,210}$	24	10	0	0	0	0	21.46	0.0	21.3	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.08	29.14	27.84	16.73
$L_{5,211}$	24	13	0	0	0	0	22.39	0.0	21.8	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.30	24.13	28.78	15.43
$L_{5,212}$	24	16	0	0	0	0	22.40	0.0	22.1	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.16	23.76	25.89	18.50
$L_{5,213}$	36	8	0	0	0	0	256.13	0.0	253.7	0.0	0.0	0.0	0.0	0.0	0.0	ERR	1.18	503.75	509.83	239.50
$L_{5,214}$	36	11	0	0	0	0	263.11	0.0	256.6	0.0	0.0	0.0	0.0	0.0	0.0	X	2.86	506.05	534.93	164.84
$L_{5,215}$	36	15	0	0	0	0	259.94	0.0	259.5	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.55	528.12	511.02	152.70
$L_{5,216}$	36	18	0	0	0	0	266.87	0.0	264.3	0.0	0.0	0.0	0.0	0.0	0.0	X	1.55	INT	INT	158.70

Benchmarks L_5.

$\mathscr{C}$	d	τ	S	E	V	A	Isot	Ib	Ig	Ir	If	Iv	Ia	Is	Ic	Ins	Top	CA0	CA1	C2d
$L_{6,1}$	3	5	0	4	0	0	0.11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.07	0.04	0.02	0.02	0.33
$L_{6,2}$	3	9	0	4	0	0	0.10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.07	0.05	0.02	0.02	0.32
$L_{6,3}$	3	12	0	4	0	0	0.10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.07	0.06	0.02	0.02	0.32
$L_{6,4}$	3	15	0	4	0	0	0.10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.08	0.04	0.02	0.02	0.33
$L_{6,5}$	4	6	0	7	0	0	0.16	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.22	0.08	0.04	0.05	0.32
$L_{6,6}$	4	10	0	7	0	0	0.17	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.22	0.09	0.04	0.05	0.32
$L_{6,7}$	4	13	0	7	0	0	0.17	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.21	0.08	0.04	0.04	0.33
$L_{6,8}$	4	16	0	7	0	0	0.17	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.22	0.09	0.04	0.04	0.32
$L_{6,9}$	5	7	0	12	0	0	0.33	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.71	0.20	0.11	0.12	0.34
$L_{6,10}$	5	10	0	12	0	0	0.34	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.73	0.20	0.11	0.10	0.34
$L_{6,11}$	5	14	0	12	0	0	0.34	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.72	0.20	0.10	0.11	0.35
$L_{6,12}$	5	17	0	12	0	0	0.32	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.72	0.20	0.10	0.11	0.34
$L_{6,13}$	6	8	0	19	0	0	0.76	0.1	0.0	0.0	0.1	0.0	0.0	0.2	0.0	1.94	0.44	0.24	0.25	0.52
$L_{6,14}$	6	11	0	19	0	0	0.75	0.1	0.0	0.0	0.1	0.0	0.0	0.2	0.0	1.92	0.43	0.23	0.24	0.52
$L_{6,15}$	6	15	0	19	0	0	0.75	0.1	0.0	0.0	0.1	0.0	0.0	0.2	0.0	1.93	0.44	0.24	0.27	0.52
$L_{6,16}$	6	18	0	19	0	0	0.76	0.1	0.0	0.0	0.1	0.0	0.0	0.2	0.0	1.92	0.45	0.25	0.24	0.53
$L_{6,17}$	7	9	0	26	0	0	1.88	0.1	0.0	0.0	0.6	0.0	0.1	0.4	0.0	7.61	0.96	0.47	0.49	0.72
$L_{6,18}$	7	12	0	26	0	0	1.88	0.1	0.1	0.0	0.6	0.0	0.0	0.4	0.0	7.79	0.98	0.47	0.49	ERR
$L_{6,19}$	7	16	0	26	0	0	1.89	0.1	0.1	0.0	0.6	0.0	0.1	0.4	0.0	7.73	0.99	0.47	0.49	0.71

Benchmarks L_6 (continued on the next page).

$\mathscr{C}$	d	τ	S	E	V	A	ISOT	IB	IG	IR	IF	IV	IA	IS	IC	INS	TOP	CA0	CA1	C2D
$L_{6,20}$	7	19	0	26	0	0	1.90	0.1	0.1	0.0	0.6	0.0	0.1	0.4	0.0	7.77	0.98	0.50	0.47	0.74
$L_{6,21}$	8	10	0	37	0	0	5.00	0.2	0.1	0.0	2.3	0.0	0.1	0.8	0.0	27.04	2.00	1.02	1.10	1.40
$L_{6,22}$	8	13	0	37	0	0	5.02	0.2	0.1	0.0	2.3	0.0	0.1	0.8	0.0	27.64	2.02	1.00	1.00	1.38
$L_{6,23}$	8	17	0	37	0	0	5.00	0.2	0.1	0.0	2.3	0.0	0.1	0.8	0.0	27.84	2.05	1.04	1.05	1.40
$L_{6,24}$	8	20	0	37	0	0	5.08	0.2	0.1	0.0	2.3	0.0	0.1	0.8	0.0	28.20	2.07	1.05	1.01	1.38
$L_{6,25}$	9	11	0	46	0	0	11.77	0.3	0.3	0.0	6.3	0.0	0.2	1.2	0.0	83.47	144.42	2.24	2.29	3.09
$L_{6,26}$	9	14	0	46	0	0	11.72	0.3	0.3	0.0	6.3	0.0	0.2	1.2	0.0	83.35	149.21	2.24	2.30	3.15
$L_{6,27}$	9	17	0	46	0	0	11.76	0.3	0.3	0.0	6.3	0.0	0.2	1.2	0.0	82.99	152.91	2.37	2.33	3.16
$L_{6,28}$	9	21	0	46	0	0	11.92	0.3	0.4	0.0	6.3	0.0	0.2	1.2	0.0	82.83	155.93	2.31	2.32	3.12
$L_{6,29}$	10	12	0	61	0	0	34.12	0.6	0.8	0.0	23.1	0.0	0.3	2.4	0.0	210.56	INT	4.92	5.01	PRIM
$L_{6,30}$	10	15	0	61	0	0	34.01	0.6	0.8	0.0	23.0	0.0	0.3	2.5	0.0	209.25	INT	5.06	5.01	PRIM
$L_{6,31}$	10	18	0	61	0	0	33.94	0.6	0.8	0.0	23.2	0.0	0.3	2.4	0.0	212.15	X	5.04	5.06	PRIM
$L_{6,32}$	10	22	0	61	0	0	34.13	0.6	0.8	0.0	23.1	0.0	0.3	2.4	0.0	213.55	X	5.14	5.19	PRIM
$L_{6,33}$	3	5	0	0	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.04	0.00	0.00	0.32
$L_{6,34}$	3	9	0	0	0	0	0.07	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.02	0.00	0.00	0.32
$L_{6,35}$	3	12	0	0	0	0	0.07	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.04	0.00	0.00	0.32
$L_{6,36}$	3	15	0	0	0	0	0.05	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.03	0.00	0.00	0.32
$L_{6,37}$	4	6	0	1	0	0	0.10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.04	0.01	0.01	0.32
$L_{6,38}$	4	10	0	1	0	0	0.09	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.04	0.01	0.01	0.32
$L_{6,39}$	4	13	0	1	0	0	0.08	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.06	0.01	0.01	0.32
$L_{6,40}$	4	16	0	1	0	0	0.07	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.05	0.01	0.01	0.32
$L_{6,41}$	5	7	0	2	0	0	0.15	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.28	0.06	0.03	0.03	0.33
$L_{6,42}$	5	10	0	2	0	0	0.15	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.28	0.06	0.03	0.03	0.33
$L_{6,43}$	5	14	0	2	0	0	0.16	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.28	0.07	0.02	0.03	0.34
$L_{6,44}$	5	17	0	2	0	0	0.16	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.27	0.07	0.03	0.03	0.34
$L_{6,45}$	6	8	0	3	0	0	0.30	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.65	0.14	0.06	0.07	0.41
$L_{6,46}$	6	11	0	3	0	0	0.30	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.64	0.12	0.06	0.07	0.40
$L_{6,47}$	6	15	0	3	0	0	0.32	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.64	0.12	0.06	0.07	0.40
$L_{6,48}$	6	18	0	3	0	0	0.30	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.64	0.14	0.08	0.08	0.40
$L_{6,49}$	7	9	0	6	0	0	1.08	0.1	0.1	0.0	0.4	0.0	0.0	0.1	0.0	4.65	0.35	0.18	0.20	0.57
$L_{6,50}$	7	12	0	6	0	0	1.07	0.1	0.1	0.0	0.4	0.0	0.1	0.1	0.0	4.59	0.36	0.20	0.20	0.56
$L_{6,51}$	7	16	0	6	0	0	1.09	0.1	0.1	0.0	0.4	0.0	0.1	0.1	0.0	4.67	0.36	0.19	0.20	0.58
$L_{6,52}$	7	19	0	6	0	0	1.11	0.1	0.1	0.0	0.4	0.0	0.1	0.1	0.0	4.78	0.37	0.19	0.20	0.58
$L_{6,53}$	8	10	0	7	0	0	1.86	0.2	0.1	0.0	0.6	0.0	0.1	0.1	0.0	15.72	0.68	0.36	0.38	271.00
$L_{6,54}$	8	13	0	7	0	0	2.21	0.2	0.1	0.0	0.9	0.0	0.1	0.1	0.0	15.77	0.68	0.38	0.39	278.14
$L_{6,55}$	8	17	0	7	0	0	1.87	0.2	0.1	0.0	0.6	0.0	0.1	0.1	0.0	15.94	0.70	0.38	0.39	289.49
$L_{6,56}$	8	20	0	7	0	0	1.91	0.2	0.1	0.0	0.6	0.0	0.1	0.1	0.0	16.02	0.72	0.37	0.39	308.80
$L_{6,57}$	9	11	0	12	0	0	8.05	0.4	0.3	0.0	4.6	0.0	0.2	0.3	0.0	64.73	1.85	1.24	1.32	2.46
$L_{6,58}$	9	14	0	12	0	0	7.96	0.4	0.3	0.0	4.6	0.0	0.2	0.3	0.0	64.02	1.90	1.32	1.32	2.49

Benchmarks L_6 (continued on the next page).

$\mathscr{C}$	d	τ	S	E	V	A	ISOT	IB	IG	IR	IF	IV	IA	IS	IC	INS	TOP	CA0	CA1	C2D
$L_{6,59}$	9	17	0	12	0	0	7.99	0.4	0.3	0.0	4.6	0.0	0.2	0.3	0.0	64.66	1.96	1.28	1.32	2.52
$L_{6,60}$	9	21	0	12	0	0	8.12	0.3	0.4	0.0	4.6	0.0	0.2	0.3	0.0	64.38	2.03	1.27	1.34	2.61
$L_{6,61}$	10	12	0	13	0	0	15.08	0.6	0.8	0.0	8.7	0.0	0.2	0.5	0.0	159.42	7.94	2.89	2.90	PRIM
$L_{6,62}$	10	15	0	13	0	0	15.07	0.6	0.8	0.0	8.6	0.0	0.3	0.5	0.0	158.91	7.98	2.78	2.90	PRIM
$L_{6,63}$	10	18	0	13	0	0	15.05	0.6	0.8	0.0	8.6	0.0	0.3	0.5	0.0	157.90	8.26	2.91	2.79	PRIM
$L_{6,64}$	10	22	0	13	0	0	14.99	0.6	0.8	0.0	8.5	0.0	0.3	0.5	0.0	158.44	8.55	2.87	2.79	PRIM

Benchmarks L_6.

$\mathscr{C}$	d	τ	S	E	V	A	ISOT	IB	IG	IR	IF	IV	IA	IS	IC	INS	TOP	CA0	CA1	C2D
$L_{7,1}$	4	1	1	2	0	0	0.14	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.05	0.04	0.01	0.01	0.32
$L_{7,2}$	6	1	1	2	0	0	0.13	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.10	0.06	0.02	0.03	0.32
$L_{7,3}$	8	1	1	2	0	0	0.16	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.14	0.06	0.04	0.06	0.32
$L_{7,4}$	10	1	1	2	0	0	0.21	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.17	0.07	0.06	0.10	0.33
$L_{7,5}$	12	1	1	2	0	0	0.27	0.1	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.22	0.08	0.12	0.20	0.34
$L_{7,6}$	14	1	1	2	0	0	0.36	0.1	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.31	0.09	0.22	0.35	0.37
$L_{7,7}$	16	1	1	2	0	0	0.56	0.1	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.66	0.10	0.38	0.63	0.39
$L_{7,8}$	18	1	1	2	0	0	0.80	0.1	0.6	0.0	0.0	0.0	0.0	0.0	0.0	1.56	0.15	0.71	1.19	0.46
$L_{7,9}$	20	1	1	2	0	0	1.43	0.1	1.1	0.0	0.0	0.0	0.0	0.0	0.0	1.99	0.17	1.26	1.85	0.48
$L_{7,10}$	22	1	1	2	0	0	2.27	0.1	1.9	0.0	0.0	0.0	0.0	0.0	0.0	2.68	0.22	2.00	2.98	0.80
$L_{7,11}$	24	1	1	2	0	0	3.70	0.1	3.2	0.0	0.0	0.0	0.0	0.0	0.0	3.79	0.30	3.11	4.32	0.73

Benchmarks L_7.

$\mathscr{C}$	d	τ	S	E	V	A	ISOT	IB	IG	IR	IF	IV	IA	IS	IC	INS	TOP	CA0	CA1	C2D
$L_{9,1}$	4	1	2	4	0	0	0.16	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.22	0.08	0.07	0.06	0.32
$L_{9,2}$	6	2	2	4	0	0	0.18	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.30	0.10	0.14	0.14	0.32
$L_{9,3}$	8	3	4	4	0	0	0.33	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.0	1.40	0.32	0.30	0.38	0.33
$L_{9,4}$	10	4	4	4	0	0	0.43	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.0	2.25	0.32	0.55	0.77	0.37
$L_{9,5}$	12	8	6	12	0	0	1.05	0.1	0.1	0.0	0.0	0.0	0.0	0.3	0.0	5.98	1.01	1.46	1.80	0.38
$L_{9,6}$	14	9	6	8	0	0	1.22	0.1	0.1	0.0	0.0	0.0	0.0	0.3	0.0	9.42	1.30	1.82	2.17	0.75
$L_{9,7}$	16	7	8	12	0	0	2.15	0.4	0.2	0.0	0.0	0.0	0.0	0.6	0.0	31.52	2.80	4.60	4.46	0.50
$L_{9,8}$	18	7	8	8	0	0	2.45	0.5	0.3	0.0	0.0	0.0	0.0	0.6	0.0	47.61	3.01	4.58	6.63	1.76
$L_{9,9}$	20	13	10	20	0	0	5.46	0.1	1.0	0.0	0.1	0.0	0.1	1.3	0.0	77.50	ERR	15.80	19.70	1.98
$L_{9,10}$	22	14	10	12	0	0	6.64	0.1	1.7	0.0	0.2	0.0	0.1	1.4	0.0	116.39	INT	18.70	15.80	10.20
$L_{9,11}$	24	14	12	20	0	0	11.99	1.2	2.5	0.0	0.6	0.0	0.1	2.4	0.0	300.38	X	34.10	61.10	6.44

Benchmarks L_9.

$\mathscr{C}$	d	τ	S	E	V	A	Isot	Ib	Ig	Ir	If	Iv	Ia	Is	Ic	Ins	Top	CA0	CA1	C2d
$L_{10,1,0}$	4	3	3	2	0	0	0.17	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	X	X	0.02	0.02	0.33
$L_{10,2,0}$	6	3	2	2	0	0	0.15	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	X	X	0.02	0.03	0.32
$L_{10,3,0}$	6	6	5	2	0	0	0.29	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	X	X	0.04	0.06	0.32
$L_{10,4,0}$	8	8	2	2	0	0	0.27	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	X	X	0.06	0.07	0.33
$L_{10,5,0}$	8	6	3	2	0	0	0.24	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	X	X	0.04	0.06	0.33
$L_{10,6,0}$	8	10	4	8	0	0	0.43	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.0	X	X	0.14	0.15	0.34
$L_{10,7,0}$	10	8	2	2	0	0	0.32	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	X	X	0.06	0.09	0.34
$L_{10,8,0}$	10	6	3	2	0	0	0.27	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.0	X	X	0.07	0.10	0.34
$L_{10,9,0}$	10	10	4	8	0	0	0.52	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.0	X	X	0.20	0.22	ERR
$L_{10,10,0}$	10	14	5	10	0	0	0.72	0.1	0.0	0.0	0.0	0.0	0.0	0.2	0.0	X	X	0.27	0.32	0.49
$L_{10,11,0}$	12	13	2	2	0	0	0.55	0.1	0.1	0.0	0.0	0.0	0.0	0.0	0.0	X	X	0.18	0.26	0.36
$L_{10,12,0}$	12	15	3	2	0	0	0.60	0.1	0.1	0.0	0.0	0.0	0.0	0.0	0.0	X	X	0.22	0.30	0.38
$L_{10,13,0}$	12	10	4	6	0	0	0.60	0.1	0.1	0.0	0.0	0.0	0.0	0.1	0.0	X	X	0.22	0.24	0.39
$L_{10,14,0}$	12	14	5	10	0	0	0.85	0.1	0.1	0.0	0.0	0.0	0.0	0.2	0.0	X	X	0.36	0.41	0.56
$L_{10,15,0}$	12	19	6	12	0	0	1.22	0.1	0.1	0.0	0.1	0.0	0.1	0.2	0.0	X	X	0.51	0.60	0.61
$L_{10,16,0}$	14	13	2	2	0	0	0.64	0.1	0.2	0.0	0.0	0.0	0.0	0.0	0.0	X	X	0.28	0.40	0.41
$L_{10,17,0}$	14	15	3	2	0	0	0.74	0.1	0.2	0.0	0.0	0.0	0.0	0.1	0.0	X	X	0.30	0.42	0.44
$L_{10,18,0}$	14	10	4	6	0	0	0.66	0.1	0.1	0.0	0.0	0.0	0.0	0.1	0.0	X	X	0.28	0.32	0.46
$L_{10,19,0}$	14	14	5	10	0	0	1.00	0.1	0.1	0.0	0.1	0.0	0.1	0.1	0.0	X	X	0.48	0.50	0.61
$L_{10,20,0}$	14	19	6	12	0	0	1.42	0.1	0.2	0.0	0.1	0.0	0.1	0.2	0.0	X	X	0.68	0.78	0.69
$L_{10,21,0}$	14	24	7	14	0	0	2.06	0.1	0.3	0.0	0.2	0.0	0.1	0.2	0.0	X	X	0.87	1.16	0.75
$L_{10,22,0}$	16	18	2	2	0	0	1.33	0.1	0.6	0.0	0.0	0.0	0.1	0.0	0.0	X	X	0.69	0.95	0.55
$L_{10,23,0}$	16	15	3	2	0	0	0.89	0.1	0.3	0.0	0.0	0.0	0.1	0.0	0.0	X	X	0.41	0.57	0.48
$L_{10,24,0}$	16	23	4	6	0	0	1.79	0.1	0.6	0.0	0.1	0.0	0.1	0.1	0.0	X	X	1.03	1.11	0.69
$L_{10,25,0}$	16	14	5	10	0	0	1.15	0.1	0.2	0.0	0.1	0.0	0.0	0.2	0.0	X	X	0.59	0.64	0.64
$L_{10,26,0}$	16	19	6	12	0	0	1.67	0.1	0.3	0.0	0.1	0.0	0.1	0.2	0.0	X	X	0.92	0.96	0.72
$L_{10,27,0}$	16	24	7	14	0	0	2.39	0.1	0.4	0.0	0.3	0.0	0.1	0.3	0.0	X	X	1.28	1.44	0.82
$L_{10,28,0}$	16	29	8	16	0	0	3.20	0.1	0.6	0.0	0.4	0.0	0.2	0.3	0.0	X	X	1.66	1.80	0.90
$L_{10,29,0}$	18	18	2	2	0	0	1.63	0.1	0.9	0.0	0.0	0.0	0.1	0.0	0.0	X	X	0.88	1.26	0.57
$L_{10,30,0}$	18	23	3	2	0	0	2.29	0.1	1.2	0.0	0.1	0.0	0.1	0.1	0.0	X	X	1.28	1.70	0.64
$L_{10,31,0}$	18	23	4	6	0	0	2.18	0.1	0.9	0.0	0.1	0.0	0.1	0.1	0.0	X	X	1.18	1.48	0.74
$L_{10,32,0}$	18	14	5	10	0	0	1.35	0.1	0.2	0.0	0.1	0.0	0.1	0.2	0.0	X	X	0.74	0.88	0.61
$L_{10,33,0}$	18	19	6	12	0	0	1.95	0.1	0.4	0.0	0.1	0.0	0.1	0.2	0.0	X	X	1.18	1.21	0.76
$L_{10,34,0}$	18	24	7	14	0	0	2.76	0.1	0.6	0.0	0.3	0.0	0.1	0.3	0.0	X	X	1.38	1.64	0.90
$L_{10,35,0}$	18	29	8	16	0	0	3.80	0.1	0.9	0.0	0.4	0.0	0.2	0.3	0.0	X	X	2.14	2.42	1.07
$L_{10,36,0}$	18	34	9	18	0	0	5.32	0.1	1.3	0.0	0.7	0.0	0.3	0.4	0.0	X	X	2.76	3.45	1.35
$L_{10,37,0}$	20	23	2	2	0	0	3.45	0.1	2.2	0.0	0.1	0.0	0.1	0.0	0.0	X	X	2.13	2.76	0.78
$L_{10,38,0}$	20	23	3	2	0	0	2.92	0.1	1.7	0.0	0.1	0.0	0.1	0.1	0.0	X	X	1.76	2.31	0.72
$L_{10,39,0}$	20	23	4	6	0	0	2.69	0.1	1.2	0.0	0.1	0.0	0.1	0.1	0.0	X	X	1.56	1.92	0.80

Benchmarks L_{10} (continued on the next page).

$\mathscr{C}$	d	τ	S	E	V	A	ISOT	IB	IG	IR	IF	IV	IA	IS	IC	INS	TOP	CA0	CA1	C2D
$L_{10,40,0}$	20	31	5	10	0	0	5.00	0.1	2.4	0.0	0.2	0.0	0.2	0.2	0.0	X	X	3.35	3.79	1.29
$L_{10,41,0}$	20	19	6	12	0	0	2.26	0.1	0.5	0.0	0.1	0.0	0.1	0.2	0.0	X	X	1.24	1.46	0.72
$L_{10,42,0}$	20	24	7	14	0	0	3.24	0.1	0.9	0.0	0.3	0.0	0.1	0.3	0.0	X	X	1.84	2.11	ERR
$L_{10,43,0}$	20	29	8	16	0	0	4.49	0.1	1.2	0.0	0.5	0.0	0.2	0.3	0.0	X	X	2.66	3.20	2.59
$L_{10,44,0}$	20	34	9	18	0	0	6.31	0.1	1.7	0.0	0.8	0.0	0.3	0.4	0.0	X	X	4.65	5.29	ERR
$L_{10,45,0}$	20	39	10	20	0	0	9.71	0.1	2.4	0.0	2.2	0.0	0.5	0.5	0.0	X	X	7.70	8.37	4.10
$L_{10,46,0}$	22	23	2	2	0	0	4.38	0.1	3.0	0.0	0.2	0.0	0.2	0.1	0.0	X	X	2.86	3.72	1.17
$L_{10,47,0}$	22	23	3	2	0	0	3.66	0.1	2.3	0.0	0.1	0.0	0.2	0.1	0.0	X	X	2.37	3.08	1.26
$L_{10,48,0}$	22	23	4	6	0	0	3.26	0.1	1.6	0.0	0.1	0.0	0.1	0.1	0.0	X	X	2.37	2.58	1.27
$L_{10,49,0}$	22	31	5	10	0	0	6.04	0.1	3.1	0.0	0.3	0.0	0.2	0.2	0.0	X	X	5.06	6.18	3.18
$L_{10,50,0}$	22	19	6	12	0	0	2.58	0.1	0.7	0.0	0.1	0.0	0.1	0.3	0.0	X	X	1.50	1.86	0.92
$L_{10,51,0}$	22	24	7	14	0	0	3.78	0.1	1.1	0.0	0.4	0.0	0.2	0.3	0.0	X	X	2.68	3.02	2.12
$L_{10,52,0}$	22	29	8	16	0	0	5.31	0.1	1.7	0.0	0.6	0.0	0.2	0.3	0.0	X	X	4.19	5.18	3.48
$L_{10,53,0}$	22	34	9	18	0	0	7.61	0.1	2.4	0.0	1.0	0.0	0.3	0.4	0.0	X	X	6.96	7.60	3.72
$L_{10,54,0}$	22	39	10	20	0	0	11.60	0.1	3.2	0.0	2.7	0.0	0.5	0.5	0.0	X	X	7.91	9.09	4.16
$L_{10,55,0}$	22	45	11	22	0	0	15.34	0.1	4.2	0.0	3.9	0.0	0.8	0.5	0.0	X	X	11.10	10.50	ERR
$L_{10,56,0}$	24	28	2	2	0	0	8.67	0.1	6.6	0.0	0.2	0.0	0.3	0.1	0.0	X	X	7.02	8.14	2.50
$L_{10,57,0}$	24	32	3	2	0	0	9.21	0.1	6.7	0.0	0.3	0.0	0.3	0.1	0.0	X	X	6.88	8.49	2.38
$L_{10,58,0}$	24	36	4	6	0	0	10.05	0.1	6.9	0.0	0.3	0.0	0.3	0.2	0.0	X	X	7.98	9.67	3.36
$L_{10,59,0}$	24	31	5	10	0	0	7.30	0.1	4.1	0.0	0.3	0.0	0.3	0.2	0.0	X	X	6.68	7.68	3.31
$L_{10,60,0}$	24	41	6	12	0	0	11.78	0.1	7.1	0.0	0.5	0.0	0.5	0.3	0.0	X	X	9.78	11.50	4.05
$L_{10,61,0}$	24	24	7	14	0	0	4.41	0.1	1.5	0.0	0.5	0.0	0.2	0.3	0.0	X	X	3.71	4.27	2.52
$L_{10,62,0}$	24	29	8	16	0	0	6.32	0.1	2.2	0.0	0.8	0.0	0.2	0.4	0.0	X	X	6.04	6.49	3.63
$L_{10,63,0}$	24	34	9	18	0	0	8.97	0.1	3.1	0.0	1.2	0.0	0.4	0.5	0.0	X	X	6.70	7.47	4.10
$L_{10,64,0}$	24	39	10	20	0	0	13.74	0.1	4.2	0.0	3.2	0.0	0.6	0.5	0.0	X	X	9.23	10.10	4.12
$L_{10,65,0}$	24	45	11	22	0	0	18.39	0.1	5.6	0.0	4.7	0.0	0.9	0.5	0.0	X	X	13.60	14.30	9.83
$L_{10,66,0}$	24	50	12	24	0	0	24.50	0.1	7.2	0.0	6.8	0.0	1.1	0.6	0.0	X	X	17.50	19.50	12.38

Benchmarks L_{10}.

$\mathscr{C}$	d	τ	S	E	V	A	ISOT	IB	IG	IR	IF	IV	IA	IS	IC	INS	TOP	CA0	CA1	C2D
$L_{11,1}$	4	2	2	4	0	0	0.17	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.18	0.08	0.02	0.03	0.32
$L_{11,2}$	6	2	2	4	0	0	0.20	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.20	0.09	0.03	0.05	0.34
$L_{11,3}$	6	5	3	6	0	0	0.28	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.41	0.15	0.06	0.07	0.32
$L_{11,4}$	8	5	2	4	0	0	0.36	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.37	0.14	0.07	0.09	0.32
$L_{11,5}$	8	5	3	6	0	0	0.31	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.47	0.16	0.08	0.09	0.34
$L_{11,6}$	8	7	4	8	0	0	0.40	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.70	0.24	0.13	0.14	0.33
$L_{11,7}$	10	5	2	4	0	0	0.41	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.47	0.18	0.09	0.12	0.33
$L_{11,8}$	10	5	3	6	0	0	0.37	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.56	0.20	0.11	0.12	0.32

Benchmarks L_{11} (continued on the next page).

$\mathscr{C}$	d	τ	s	e	v	a	Isot	Ib	Ig	Ir	If	Iv	Ia	Is	Ic	Ins	Top	CA0	CA1	C2d
$L_{11,9}$	10	7	4	8	0	0	0.42	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.85	0.30	0.17	0.19	0.34
$L_{11,10}$	10	10	5	10	0	0	0.63	0.0	0.0	0.0	0.0	0.0	0.1	0.1	0.0	1.60	0.42	0.23	0.26	0.34
$L_{11,11}$	12	8	2	4	0	0	0.64	0.1	0.1	0.0	0.0	0.0	0.0	0.1	0.0	1.07	0.29	0.24	0.32	0.33
$L_{11,12}$	12	10	3	6	0	0	0.73	0.1	0.1	0.0	0.0	0.0	0.0	0.1	0.0	1.25	0.33	0.30	0.39	0.36
$L_{11,13}$	12	7	4	8	0	0	0.52	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.0	1.03	0.38	0.28	0.28	0.34
$L_{11,14}$	12	10	5	10	0	0	0.80	0.1	0.1	0.0	0.0	0.0	0.1	0.1	0.0	1.89	0.53	0.34	0.40	0.36
$L_{11,15}$	12	12	6	12	0	0	ERR	ERR	ERR	ERR	ERR	ERR	ERR	ERR	ERR	2.81	0.75	0.52	0.57	0.36
$L_{11,16}$	14	8	2	4	0	0	0.76	0.1	0.2	0.0	0.0	0.0	0.0	0.1	0.0	1.52	0.36	0.30	0.42	0.36
$L_{11,17}$	14	10	3	6	0	0	0.91	0.1	0.2	0.0	0.0	0.0	0.0	0.1	0.0	1.65	0.42	0.34	0.46	0.36
$L_{11,18}$	14	7	4	8	0	0	0.59	0.1	0.1	0.0	0.0	0.0	0.0	0.1	0.0	1.41	0.46	0.32	0.36	0.35
$L_{11,19}$	14	10	5	10	0	0	0.94	0.0	0.1	0.0	0.0	0.0	0.1	0.1	0.0	2.62	0.62	0.44	0.52	0.37
$L_{11,20}$	14	12	6	12	0	0	1.14	0.1	0.2	0.0	0.0	0.0	0.0	0.2	0.0	3.62	0.93	0.62	0.72	0.39
$L_{11,21}$	14	15	7	14	0	0	1.58	0.1	0.2	0.0	0.1	0.0	0.1	0.2	0.0	6.25	1.20	0.81	0.94	0.50
$L_{11,22}$	16	11	2	4	0	0	1.62	0.1	0.5	0.0	0.0	0.0	0.0	0.1	0.0	3.84	0.59	0.58	0.86	ERR
$L_{11,23}$	16	10	3	6	0	0	1.10	0.1	0.3	0.0	0.0	0.0	0.0	0.1	0.0	2.30	0.47	0.51	0.66	0.40
$L_{11,24}$	16	15	4	8	0	0	1.68	0.1	0.5	0.0	0.0	0.0	0.0	0.1	0.0	6.79	1.04	0.75	0.97	0.62
$L_{11,25}$	16	10	5	10	0	0	1.16	0.1	0.1	0.0	0.0	0.0	0.1	0.2	0.0	3.48	0.75	0.58	0.67	0.38
$L_{11,26}$	16	12	6	12	0	0	1.34	0.1	0.2	0.0	0.0	0.0	0.0	0.2	0.0	5.30	1.11	0.74	0.90	ERR
$L_{11,27}$	16	15	7	14	0	0	1.89	0.1	0.3	0.0	0.1	0.0	0.2	0.3	0.0	7.64	1.44	0.95	1.13	ERR
$L_{11,28}$	16	17	8	16	0	0	2.29	0.1	0.5	0.0	0.1	0.0	0.1	0.3	0.0	10.18	2.02	1.25	1.52	ERR
$L_{11,29}$	18	11	2	4	0	0	1.86	0.1	0.8	0.0	0.0	0.0	0.0	0.1	0.0	5.23	0.70	0.80	1.10	0.47
$L_{11,30}$	18	16	3	6	0	0	2.61	0.1	1.0	0.0	0.0	0.0	0.2	0.1	0.0	7.96	13.50	1.24	1.45	0.57
$L_{11,31}$	18	15	4	8	0	0	2.19	0.1	0.8	0.0	0.1	0.0	0.0	0.1	0.0	8.44	1.29	0.92	1.38	0.52
$L_{11,32}$	18	10	5	10	0	0	1.42	0.0	0.2	0.0	0.1	0.0	0.1	0.2	0.0	3.94	0.85	0.71	0.78	0.41
$L_{11,33}$	18	12	6	12	0	0	1.64	0.1	0.3	0.0	0.1	0.0	0.0	0.2	0.0	6.06	1.33	0.87	1.06	0.42
$L_{11,34}$	18	15	7	14	0	0	2.23	0.1	0.5	0.0	0.1	0.0	0.2	0.3	0.0	9.92	1.74	1.24	1.40	0.48
$L_{11,35}$	18	17	8	16	0	0	2.78	0.1	0.7	0.0	0.1	0.0	0.1	0.3	0.0	13.47	X	1.62	1.96	0.54
$L_{11,36}$	18	20	9	18	0	0	3.68	0.1	0.9	0.0	0.1	0.0	0.3	0.4	0.0	19.89	55.42	2.10	2.60	0.71
$L_{11,37}$	20	14	2	4	0	0	3.41	0.1	2.0	0.0	0.0	0.0	0.1	0.1	0.0	13.16	X	1.59	2.16	0.68
$L_{11,38}$	20	16	3	6	0	0	3.21	0.1	1.3	0.0	0.1	0.0	0.2	0.1	0.0	10.24	X	1.43	1.86	0.83
$L_{11,39}$	20	15	4	8	0	0	2.76	0.1	1.0	0.0	0.1	0.0	0.1	0.2	0.0	12.06	INT	1.24	1.66	0.52
$L_{11,40}$	20	20	5	10	0	0	4.77	0.1	2.0	0.0	0.1	0.0	0.1	0.2	0.0	21.23	X	2.39	2.70	0.79
$L_{11,41}$	20	12	6	12	0	0	1.94	0.1	0.5	0.0	0.1	0.0	0.0	0.2	0.0	7.86	1.53	1.06	1.30	0.73
$L_{11,42}$	20	15	7	14	0	0	2.61	0.1	0.6	0.0	0.1	0.0	0.2	0.3	0.0	12.38	10.94	1.39	1.72	0.86
$L_{11,43}$	20	17	8	16	0	0	3.50	0.1	1.0	0.0	0.1	0.0	0.1	0.3	0.0	17.60	X	1.92	2.34	0.99
$L_{11,44}$	20	20	9	18	0	0	4.46	0.1	1.3	0.0	0.2	0.0	0.4	0.4	0.0	28.52	X	2.48	3.45	1.37
$L_{11,45}$	20	22	10	20	0	0	6.02	0.1	1.9	0.0	0.8	0.0	0.1	0.4	0.0	40.77	504.99	2.99	3.60	1.45
$L_{11,46}$	22	14	2	4	0	0	4.19	0.1	2.7	0.0	0.0	0.0	0.1	0.1	0.0	17.68	X	2.08	2.86	0.91
$L_{11,47}$	22	16	3	6	0	0	3.76	0.1	1.8	0.0	0.0	0.0	0.3	0.1	0.0	13.42	X	1.74	2.46	1.06

Benchmarks L_{11} (continued on the next page).

$\mathscr{C}$	d	τ	s	e	v	a	Isot	Ib	Ig	Ir	If	Iv	Ia	Is	Ic	Ins	Top	CA0	CA1	C2d
$L_{11,48}$	22	15	4	8	0	0	3.40	0.1	1.4	0.0	0.1	0.0	0.1	0.1	0.0	13.45	X	1.55	2.02	1.00
$L_{11,49}$	22	20	5	10	0	0	5.92	0.1	2.6	0.0	0.1	0.0	0.1	0.2	0.0	34.26	X	2.67	3.32	1.29
$L_{11,50}$	22	12	6	12	0	0	2.30	0.1	0.6	0.0	0.1	0.0	0.0	0.2	0.0	12.30	1.78	1.34	1.57	0.86
$L_{11,51}$	22	15	7	14	0	0	3.19	0.1	0.9	0.0	0.1	0.0	0.2	0.3	0.0	15.96	14.28	1.72	2.20	1.03
$L_{11,52}$	22	17	8	16	0	0	4.24	0.1	1.4	0.0	0.1	0.0	0.1	0.4	0.0	22.55	X	2.60	2.81	1.22
$L_{11,53}$	22	20	9	18	0	0	5.27	0.0	1.7	0.0	0.2	0.0	0.4	0.4	0.0	39.07	X	2.74	3.75	1.72
$L_{11,54}$	22	22	10	20	0	0	7.31	0.1	2.7	0.0	1.0	0.0	0.1	0.4	0.0	45.57	X	3.85	4.59	2.01
$L_{11,55}$	22	25	11	22	0	0	8.84	0.1	3.0	0.0	1.0	0.0	0.8	0.5	0.0	60.61	X	4.85	6.02	2.64
$L_{11,56}$	24	17	2	4	0	0	8.17	0.1	6.0	0.0	0.0	0.0	0.1	0.1	0.0	41.42	X	3.88	5.59	1.48
$L_{11,57}$	24	21	3	6	0	0	9.04	0.1	5.9	0.0	0.1	0.0	0.2	0.2	0.0	37.60	X	4.20	5.62	1.66
$L_{11,58}$	24	23	4	8	0	0	9.64	0.1	6.0	0.0	0.1	0.0	0.2	0.2	0.0	44.66	X	4.52	6.82	1.84
$L_{11,59}$	24	20	5	10	0	0	7.22	0.1	3.5	0.0	0.1	0.0	0.1	0.2	0.0	34.99	INT	3.02	4.46	1.48
$L_{11,60}$	24	25	6	12	0	0	10.31	0.1	6.0	0.0	0.2	0.0	0.2	0.3	0.0	59.05	X	5.70	7.24	2.62
$L_{11,61}$	24	15	7	14	0	0	3.74	0.1	1.1	0.0	0.1	0.0	0.2	0.3	0.0	20.85	18.74	2.08	2.93	1.22
$L_{11,62}$	24	17	8	16	0	0	5.00	0.1	1.8	0.0	0.2	0.0	0.1	0.4	0.0	28.62	575.03	2.59	3.14	1.37
$L_{11,63}$	24	20	9	18	0	0	6.34	0.1	2.3	0.0	0.2	0.0	0.5	0.5	0.0	41.70	X	3.78	4.32	1.84
$L_{11,64}$	24	22	10	20	0	0	8.56	0.1	3.4	0.0	1.1	0.0	0.1	0.5	0.0	55.15	X	4.60	5.48	2.18
$L_{11,65}$	24	25	11	22	0	0	10.44	0.1	4.0	0.0	1.1	0.0	0.9	0.5	0.0	75.11	586.40	6.81	8.76	2.66
$L_{11,66}$	24	27	12	24	0	0	34.90	0.1	5.8	0.0	10.8	0.0	0.2	0.6	0.0	102.16	X	9.79	10.60	2.78

Benchmarks L_{11}.

$\mathscr{C}$	d	τ	s	e	v	a	Isot	Ib	Ig	Ir	If	Iv	Ia	Is	Ic	Ins	Top	CA0	CA1	C2d
$L_{12,1}$	2	0	0	0	1	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.02	0.00	0.00	0.32
$L_{12,2}$	3	2	1	0	0	0	0.11	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.02	0.00	0.00	0.32
$L_{12,3}$	4	0	1	0	1	0	0.15	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.02	0.00	0.00	0.32
$L_{12,4}$	5	4	1	0	0	0	0.20	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.02	0.01	0.01	0.33
$L_{12,5}$	6	4	1	0	1	0	0.22	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.02	0.01	0.01	0.32
$L_{12,6}$	7	6	1	0	0	0	0.30	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.02	0.01	0.03	0.32
$L_{12,7}$	8	3	1	0	1	0	0.35	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.03	0.02	0.03	0.32
$L_{12,8}$	9	7	1	0	0	0	0.44	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.03	0.02	0.07	0.32
$L_{12,9}$	10	7	1	0	1	0	0.50	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.03	0.03	0.07	0.32
$L_{12,10}$	11	9	1	0	0	0	0.66	0.4	0.1	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.03	0.04	0.15	0.38
$L_{12,11}$	12	8	1	0	1	0	0.68	0.4	0.1	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.04	0.04	0.15	0.32
$L_{12,12}$	4	1	1	2	0	0	0.16	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.18	0.05	0.00	0.01	0.32
$L_{12,13}$	4	1	1	4	0	0	0.17	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.13	0.05	0.02	0.02	0.33
$L_{12,14}$	6	2	1	4	0	0	0.30	0.2	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.40	0.07	0.04	0.04	0.34
$L_{12,15}$	6	2	1	6	0	0	0.33	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.28	0.10	0.05	0.06	0.33
$L_{12,16}$	8	5	1	6	0	0	0.53	0.3	0.0	0.0	0.0	0.0	0.0	0.1	0.0	1.11	0.11	0.09	0.12	0.35

Benchmarks L_{12} (continued on the next page).

$\mathscr{C}$	d	τ	S	E	V	A	ISOT	IB	IG	IR	IF	IV	IA	IS	IC	INS	TOP	CA0	CA1	C2D
$L_{12,17}$	8	3	1	8	0	0	0.52	0.3	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.63	0.15	0.11	0.14	0.35
$L_{12,18}$	10	6	1	8	0	0	0.77	0.4	0.0	0.0	0.0	0.0	0.0	0.1	0.0	2.57	0.19	0.18	0.26	0.40
$L_{12,19}$	10	4	1	10	0	0	0.80	0.4	0.0	0.0	0.0	0.0	0.0	0.1	0.0	1.33	0.24	0.22	0.30	0.41
$L_{12,20}$	12	8	1	10	0	0	1.24	0.6	0.1	0.0	0.0	0.0	0.0	0.2	0.0	6.27	0.27	0.34	0.52	0.53
$L_{12,21}$	12	5	1	12	0	0	1.37	0.7	0.1	0.0	0.0	0.0	0.0	0.3	0.0	2.55	0.38	0.40	0.60	0.58
$L_{12,22}$	14	10	1	12	0	0	1.81	0.9	0.2	0.0	0.0	0.0	0.0	0.4	0.0	13.62	0.43	0.61	1.02	0.82

Benchmarks L_{12}.

$\mathscr{C}$	d	τ	S	E	V	A	ISOT	IB	IG	IR	IF	IV	IA	IS	IC	INS	TOP	CA0	CA1	C2D
$L_{13,1}$	6	2	1	4	0	0	0.30	0.2	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.38	0.12	0.13	0.14	0.32
$L_{13,2}$	8	3	1	4	0	0	0.38	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.65	0.15	0.20	0.26	0.32
$L_{13,3}$	10	4	1	4	0	0	0.47	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.10	0.18	0.37	0.48	0.36
$L_{13,4}$	12	5	1	4	0	0	0.62	0.4	0.0	0.0	0.0	0.0	0.0	0.1	0.0	5.44	0.20	0.50	0.82	0.34
$L_{13,5}$	14	6	1	4	0	0	0.80	0.5	0.0	0.0	0.0	0.0	0.0	0.1	0.0	10.85	0.21	0.76	1.43	0.50
$L_{13,6}$	16	7	1	4	0	0	0.90	0.6	0.0	0.0	0.0	0.0	0.0	0.1	0.0	20.25	0.24	1.29	2.23	0.36
$L_{13,7}$	8	3	1	8	0	0	0.43	0.2	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.97	0.18	0.32	0.40	0.33
$L_{13,8}$	10	4	1	8	0	0	0.54	0.2	0.0	0.0	0.0	0.0	0.0	0.1	0.0	2.27	0.20	0.54	0.72	0.37
$L_{13,9}$	12	5	1	8	0	0	0.58	0.2	0.0	0.0	0.0	0.0	0.0	0.1	0.0	5.74	0.24	0.82	1.17	0.34
$L_{13,10}$	14	6	1	8	0	0	0.70	0.2	0.0	0.0	0.0	0.0	0.0	0.2	0.0	13.31	0.26	1.28	1.98	0.55
$L_{13,11}$	16	7	1	8	0	0	0.82	0.2	0.1	0.0	0.0	0.0	0.0	0.2	0.0	25.39	0.28	1.50	3.36	0.37
$L_{13,12}$	18	7	1	8	0	0	0.96	0.2	0.1	0.0	0.0	0.0	0.0	0.3	0.0	46.96	0.34	2.89	4.28	1.14
$L_{13,13}$	10	4	1	8	0	0	0.72	0.4	0.0	0.0	0.0	0.0	0.0	0.1	0.0	2.79	0.32	0.60	0.77	0.37
$L_{13,14}$	12	5	1	8	0	0	0.86	0.5	0.1	0.0	0.0	0.0	0.0	0.1	0.0	8.09	0.36	0.94	1.19	0.34
$L_{13,15}$	14	6	1	8	0	0	1.09	0.6	0.1	0.0	0.0	0.0	0.0	0.2	0.0	18.62	0.43	1.41	2.18	0.56
$L_{13,16}$	16	7	1	8	0	0	1.27	0.7	0.1	0.0	0.0	0.0	0.0	0.2	0.0	39.60	0.48	1.97	3.45	0.36
$L_{13,17}$	18	7	1	8	0	0	1.57	0.8	0.1	0.0	0.0	0.0	0.0	0.2	0.0	77.09	0.52	3.73	4.66	1.23
$L_{13,18}$	20	8	1	8	0	0	1.83	1.0	0.1	0.0	0.0	0.0	0.0	0.3	0.0	142.03	0.59	5.81	6.75	0.44
$L_{13,19}$	12	5	1	12	0	0	1.06	0.5	0.1	0.0	0.0	0.0	0.0	0.3	0.0	4.79	0.44	1.12	1.49	0.35
$L_{13,20}$	14	6	1	12	0	0	1.16	0.4	0.1	0.0	0.0	0.0	0.0	0.3	0.0	14.76	0.52	1.98	2.57	0.61
$L_{13,21}$	16	7	1	12	0	0	1.34	0.4	0.1	0.0	0.0	0.0	0.0	0.3	0.0	30.85	0.58	2.98	4.38	0.40
$L_{13,22}$	18	7	1	12	0	0	1.59	0.4	0.2	0.0	0.0	0.0	0.0	0.4	0.0	67.03	0.67	4.46	5.89	1.32
$L_{13,23}$	20	8	1	12	0	0	1.87	0.5	0.3	0.0	0.0	0.0	0.0	0.5	0.0	124.45	0.74	4.32	12.00	0.48
$L_{13,24}$	22	9	1	12	0	0	2.34	0.5	0.4	0.0	0.0	0.0	0.0	0.6	0.0	213.66	0.87	6.52	19.50	3.20
$L_{13,25}$	14	6	1	12	0	0	1.57	0.8	0.1	0.0	0.0	0.0	0.0	0.3	0.0	14.30	0.74	2.00	2.74	0.65
$L_{13,26}$	16	7	1	12	0	0	1.83	0.9	0.2	0.0	0.0	0.0	0.0	0.3	0.0	41.35	0.87	2.88	4.80	0.40
$L_{13,27}$	18	7	1	12	0	0	2.20	1.0	0.2	0.0	0.0	0.0	0.0	0.4	0.0	82.85	0.97	5.97	8.18	1.41
$L_{13,28}$	20	8	1	12	0	0	2.85	1.2	0.3	0.0	0.0	0.0	0.0	0.4	0.0	166.21	1.13	9.14	10.20	0.50
$L_{13,29}$	22	9	1	12	0	0	3.48	1.3	0.5	0.0	0.0	0.0	0.0	0.5	0.0	298.41	1.25	12.80	16.70	3.34

Benchmarks L_{13} (continued on the next page).

$\mathscr{C}$	d	τ	S	E	V	A	ISOT	IB	IG	IR	IF	IV	IA	IS	IC	INS	TOP	CA0	CA1	C2D
$L_{13,30}$	24	10	1	12	0	0	3.98	1.6	0.5	0.0	0.0	0.0	0.0	0.7	0.0	539.39	1.38	16.90	37.30	0.68
$L_{13,31}$	16	7	1	16	0	0	2.17	0.9	0.3	0.0	0.0	0.0	0.0	0.5	0.0	18.74	0.96	4.76	6.30	0.41
$L_{13,32}$	18	7	1	16	0	0	2.38	0.9	0.3	0.0	0.0	0.0	0.0	0.6	0.0	60.64	1.14	7.57	10.80	1.67
$L_{13,33}$	20	8	1	16	0	0	2.70	0.8	0.4	0.0	0.0	0.0	0.0	0.7	0.0	122.79	1.29	6.74	17.10	0.56
$L_{13,34}$	22	9	1	16	0	0	3.26	0.8	0.6	0.0	0.0	0.0	0.0	0.8	0.0	234.38	1.46	9.51	24.50	3.88
$L_{13,35}$	24	10	1	16	0	0	3.90	0.9	0.8	0.0	0.0	0.0	0.1	1.0	0.0	433.36	1.50	24.20	22.90	0.76
$L_{13,36}$	26	11	1	16	0	0	4.93	0.8	1.3	0.0	0.0	0.0	0.0	1.2	0.0	INT	1.70	32.00	31.70	9.46
$L_{13,37}$	18	7	1	16	0	0	3.18	1.5	0.6	0.0	0.0	0.0	0.0	0.5	0.0	51.23	1.57	8.96	12.00	1.88
$L_{13,38}$	20	8	1	16	0	0	3.53	1.5	0.7	0.0	0.0	0.0	0.0	0.6	0.0	146.45	1.75	10.60	24.40	0.55
$L_{13,39}$	22	9	1	16	0	0	4.14	1.7	0.8	0.0	0.0	0.0	0.0	0.7	0.0	262.48	1.98	18.60	30.50	4.30
$L_{13,40}$	24	10	1	16	0	0	5.00	1.9	1.0	0.0	0.0	0.0	0.1	0.9	0.0	517.24	2.25	23.90	39.90	0.78
$L_{13,41}$	26	11	1	16	0	0	6.25	2.1	1.4	0.0	0.0	0.0	0.1	1.2	0.0	INT	2.53	34.50	51.40	9.97
$L_{13,42}$	28	12	1	16	0	0	7.22	2.3	1.8	0.0	0.0	0.0	0.1	1.2	0.0	INT	2.86	55.10	93.20	1.20
$L_{13,43}$	20	8	1	20	0	0	4.61	1.8	1.1	0.0	0.0	0.0	0.0	0.8	0.0	57.96	1.88	14.50	21.90	0.62
$L_{13,44}$	22	9	1	20	0	0	4.98	1.6	1.3	0.0	0.0	0.0	0.0	1.0	0.0	180.50	2.14	13.80	34.80	5.14
$L_{13,45}$	24	10	1	20	0	0	5.42	1.5	1.5	0.0	0.0	0.0	0.0	1.2	0.0	338.30	2.38	30.70	44.10	0.89
$L_{13,46}$	26	11	1	20	0	0	6.54	1.3	1.9	0.0	0.0	0.0	0.0	1.5	0.0	INT	2.68	32.50	63.80	12.45
$L_{13,47}$	28	12	1	20	0	0	7.56	1.4	2.4	0.0	0.0	0.0	0.1	1.7	0.0	580.78	3.00	57.20	121.00	1.48
$L_{13,48}$	30	13	1	20	0	0	9.22	1.4	3.2	0.0	0.0	0.0	0.1	2.0	0.0	INT	3.35	78.60	170.00	24.74
$L_{13,49}$	22	9	1	20	0	0	6.87	2.8	2.1	0.0	0.0	0.0	0.0	1.0	0.0	158.06	3.33	19.10	32.20	6.68
$L_{13,50}$	24	10	1	20	0	0	7.38	2.7	2.3	0.0	0.0	0.0	0.0	1.2	0.0	447.12	3.78	30.90	56.10	0.95
$L_{13,51}$	26	11	1	20	0	0	8.56	2.8	2.7	0.0	0.0	0.0	0.1	1.4	0.0	INT	4.28	40.80	93.30	13.55
$L_{13,52}$	28	12	1	20	0	0	9.89	2.9	3.2	0.0	0.0	0.0	0.1	1.5	0.0	INT	4.86	53.30	101.00	1.49
$L_{13,53}$	30	13	1	20	0	0	11.50	3.1	3.8	0.0	0.0	0.0	0.1	1.7	0.0	INT	5.48	94.60	135.00	26.53
$L_{13,54}$	32	14	1	20	0	0	14.44	3.4	5.2	0.0	0.0	0.0	0.1	2.2	0.0	INT	6.19	111.00	219.00	2.39
$L_{13,55}$	24	10	1	24	0	0	9.85	3.3	3.7	0.0	0.0	0.0	0.0	1.5	0.0	153.21	3.55	31.40	67.00	1.08
$L_{13,56}$	26	11	1	24	0	0	10.13	2.6	4.0	0.0	0.0	0.0	0.0	1.8	0.0	458.02	3.91	46.70	78.70	15.56
$L_{13,57}$	28	12	1	24	0	0	10.98	2.4	4.6	0.0	0.0	0.0	0.1	2.0	0.0	INT	4.52	48.00	91.90	1.69
$L_{13,58}$	30	13	1	24	0	0	12.59	2.3	5.2	0.0	0.0	0.0	0.1	2.4	0.0	INT	5.05	112.00	228.00	30.54
$L_{13,59}$	32	14	1	24	0	0	14.89	2.1	6.5	0.0	0.0	0.0	0.1	2.9	0.0	INT	8.72	164.00	261.00	2.94
$L_{13,60}$	34	15	1	24	0	0	18.65	2.2	8.3	0.0	0.0	0.0	0.1	3.7	0.0	X	6.25	185.00	439.00	54.00
$L_{13,61}$	26	11	1	24	0	0	14.64	4.9	6.3	0.0	0.0	0.0	0.0	1.7	0.0	404.36	6.58	58.20	111.00	17.91
$L_{13,62}$	28	12	1	24	0	0	14.64	4.2	6.8	0.0	0.0	0.0	0.0	1.8	0.0	INT	7.31	90.90	131.00	1.79
$L_{13,63}$	30	13	1	24	0	0	16.14	4.3	7.5	0.0	0.0	0.0	0.1	2.1	0.0	INT	8.14	97.30	280.00	34.32
$L_{13,64}$	32	14	1	24	0	0	18.85	4.4	8.6	0.0	0.0	0.0	0.1	2.6	0.0	INT	13.88	149.00	322.00	3.09
$L_{13,65}$	34	15	1	24	0	0	22.09	4.7	10.0	0.0	0.0	0.0	0.1	3.1	0.0	INT	10.32	233.00	408.00	61.09
$L_{13,66}$	36	16	1	24	0	0	25.18	4.9	11.5	0.0	0.0	0.0	0.2	3.6	0.0	X	11.76	272.00	655.00	4.74
$L_{13,67}$	18	6	1	18	0	0	8.76	4.6	1.8	0.0	0.0	0.0	0.2	0.6	0.0	162.58	13.52	4.14	5.42	3.30
$L_{13,68}$	26	6	1	22	0	0	18.78	3.5	8.8	0.0	0.0	0.0	0.9	1.4	0.0	X	16.62	33.50	46.70	ERR

Benchmarks L_{13} (continued on the next page).

$\mathscr{C}$	d	τ	S	E	V	A	Isot	Ib	Ig	Ir	If	Iv	Ia	Is	Ic	Ins	Top	CA0	CA1	C2d
$L_{13,69}$	34	6	1	18	0	0	380.57	174.2	173.9	0.0	5.1	0.0	2.9	2.2	0.0	X	INT	203.00	316.00	ERR
$L_{13,70}$	42	6	1	22	0	0	540.52	123.3	353.7	0.0	6.4	0.0	6.9	3.4	0.0	X	X	584.00	1120.00	169.23
$L_{13,71}$	50	6	1	-	0	0	ERR	ERR	ERR	ERR	ERR	ERR	ERR	ERR	ERR	X	X	2920.00	4850.00	519.78
$L_{13,72}$	58	6	1	-	0	0	ERR	ERR	ERR	ERR	ERR	ERR	ERR	ERR	ERR	X	X	INT	INT	ERR
$L_{13,73}$	66	6	1	-	0	0	ERR	ERR	ERR	ERR	ERR	ERR	ERR	ERR	ERR	X	X	INT	INT	ERR

Benchmarks L_{13}.

$\mathscr{C}$	d	τ	S	E	V	A	Isot	Ib	Ig	Ir	If	Iv	Ia	Is	Ic	Ins	Top	CA0	CA1	C2d
$L_{14,1}$	7	3	1	1	0	0	0.25	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.14	0.05	0.01	0.02	0.32
$L_{14,2}$	7	4	1	1	0	0	0.25	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.16	0.04	0.01	0.01	0.32
$L_{14,3}$	7	4	1	1	0	0	0.26	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.15	0.04	0.01	0.02	0.32
$L_{14,4}$	8	3	1	2	0	0	0.32	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.24	0.12	0.07	0.08	0.32
$L_{14,5}$	8	4	1	2	0	0	0.32	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.26	0.12	0.06	0.08	0.33
$L_{14,6}$	8	4	1	2	0	0	0.35	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.26	0.12	0.07	0.08	0.32
$L_{14,7}$	9	3	1	1	0	0	0.34	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.25	0.05	0.01	0.02	0.34
$L_{14,8}$	9	4	1	1	0	0	0.36	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.24	0.05	0.01	0.02	0.32
$L_{14,9}$	9	4	1	1	0	0	0.34	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.26	0.04	0.02	0.02	0.33
$L_{14,10}$	10	3	1	2	0	0	0.41	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.45	0.14	0.10	0.12	0.33
$L_{14,11}$	10	4	1	2	0	0	0.42	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.47	0.15	0.10	0.12	0.33
$L_{14,12}$	10	4	1	2	0	0	0.38	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.46	0.16	0.10	0.12	0.32
$L_{14,13}$	11	3	1	1	0	0	0.42	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.50	0.04	0.02	0.02	0.32
$L_{14,14}$	11	4	1	1	0	0	0.42	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.50	0.04	0.02	0.02	0.33
$L_{14,15}$	11	4	1	1	0	0	0.43	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.51	0.05	0.02	0.02	0.33
$L_{14,16}$	12	3	1	2	0	0	0.49	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.89	0.26	0.14	0.16	0.33
$L_{14,17}$	12	4	1	2	0	0	0.47	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.82	0.20	0.12	0.14	0.34
$L_{14,18}$	12	4	1	2	0	0	0.46	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.84	0.22	0.12	0.14	0.32
$L_{14,19}$	13	3	1	1	0	0	0.52	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.02	0.05	0.03	0.04	0.33
$L_{14,20}$	13	4	1	1	0	0	0.51	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.95	0.04	0.02	0.03	0.33
$L_{14,21}$	13	4	1	1	0	0	0.49	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.95	0.05	0.02	0.04	0.33
$L_{14,22}$	14	3	1	2	0	0	0.55	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.51	0.35	0.15	0.18	0.33
$L_{14,23}$	14	4	1	2	0	0	0.58	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.57	0.33	0.16	0.19	0.33
$L_{14,24}$	14	4	1	2	0	0	0.56	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.58	0.34	0.16	0.19	0.32
$L_{14,25}$	15	3	1	1	0	0	0.57	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.81	0.06	0.03	0.04	0.34
$L_{14,26}$	15	4	1	1	0	0	0.56	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.82	0.05	0.03	0.04	0.33
$L_{14,27}$	15	4	1	1	0	0	0.58	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.83	0.04	0.03	0.04	0.34
$L_{14,28}$	16	3	1	2	0	0	0.66	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.58	0.42	0.19	0.23	0.34
$L_{14,29}$	16	4	1	2	0	0	0.65	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.57	0.43	0.17	0.21	0.34
$L_{14,30}$	16	4	1	2	0	0	0.63	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.61	0.46	0.17	0.22	0.34

Benchmarks L_{14} (continued on the next page).

$\mathscr{C}$	d	τ	S	E	V	A	ISOT	IB	IG	IR	IF	IV	IA	IS	IC	INS	TOP	CA0	CA1	C2D
$L_{14,31}$	17	3	1	1	0	0	0.66	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.16	0.06	0.03	0.05	0.34
$L_{14,32}$	17	4	1	1	0	0	0.67	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.22	0.06	0.03	0.05	0.34
$L_{14,33}$	17	4	1	1	0	0	0.65	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.20	0.05	0.04	0.06	0.36
$L_{14,34}$	18	3	1	2	0	0	0.74	0.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.40	0.55	0.28	0.33	0.34
$L_{14,35}$	18	4	1	2	0	0	0.76	0.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.34	0.57	0.24	0.33	0.35
$L_{14,36}$	18	4	1	2	0	0	0.76	0.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.40	0.58	0.27	0.28	0.34
$L_{14,37}$	5	3	1	2	0	0	0.22	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.18	0.04	0.01	0.02	0.32
$L_{14,38}$	5	4	1	2	0	0	0.21	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.18	0.04	0.02	0.02	0.32
$L_{14,39}$	5	4	1	2	0	0	0.22	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.18	0.05	0.01	0.02	0.32
$L_{14,40}$	6	3	1	4	0	0	0.26	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.23	0.07	0.08	0.09	0.33
$L_{14,41}$	6	4	1	4	0	0	0.29	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.22	0.08	0.08	0.08	0.32
$L_{14,42}$	6	4	1	4	0	0	0.27	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.21	0.07	0.08	0.09	0.32
$L_{14,43}$	7	3	1	2	0	0	0.28	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.23	0.06	0.02	0.02	0.32
$L_{14,44}$	7	4	1	2	0	0	0.28	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.22	0.06	0.02	0.02	0.32
$L_{14,45}$	7	4	1	2	0	0	0.29	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.25	0.04	0.02	0.02	0.33
$L_{14,46}$	8	3	1	4	0	0	0.36	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.31	0.08	0.11	0.12	0.32
$L_{14,47}$	8	4	1	4	0	0	0.35	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.32	0.09	0.10	0.12	0.33
$L_{14,48}$	8	4	1	4	0	0	0.35	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.34	0.09	0.11	0.12	0.33
$L_{14,49}$	9	3	1	2	0	0	0.37	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.30	0.06	0.03	0.03	0.32
$L_{14,50}$	9	4	1	2	0	0	0.35	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.30	0.05	0.02	0.03	0.32
$L_{14,51}$	9	4	1	2	0	0	0.36	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.30	0.06	0.02	0.03	0.32
$L_{14,52}$	10	3	1	4	0	0	0.46	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.41	0.10	0.14	0.16	0.33
$L_{14,53}$	10	4	1	4	0	0	0.43	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.42	0.09	0.14	0.15	0.33
$L_{14,54}$	10	4	1	4	0	0	0.44	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.41	0.09	0.13	0.16	0.32
$L_{14,55}$	11	3	1	2	0	0	0.45	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.34	0.06	0.03	0.05	0.33
$L_{14,56}$	11	4	1	2	0	0	0.42	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.33	0.06	0.03	0.04	0.33
$L_{14,57}$	11	4	1	2	0	0	0.46	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.36	0.06	0.03	0.04	0.34
$L_{14,58}$	12	3	1	4	0	0	0.52	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.46	0.10	0.17	0.21	0.33
$L_{14,59}$	12	4	1	4	0	0	0.53	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.46	0.10	0.18	0.20	0.33
$L_{14,60}$	12	4	1	4	0	0	0.52	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.47	0.10	0.17	0.20	0.34
$L_{14,61}$	13	3	1	2	0	0	0.49	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.40	0.07	0.04	0.06	0.34
$L_{14,62}$	13	4	1	2	0	0	0.52	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.44	0.07	0.04	0.06	0.34
$L_{14,63}$	13	4	1	2	0	0	0.51	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.45	0.06	0.04	0.06	0.34
$L_{14,64}$	14	3	1	4	0	0	0.61	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.51	0.11	0.22	0.26	0.35
$L_{14,65}$	14	4	1	4	0	0	0.62	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.55	0.12	0.21	0.26	0.35
$L_{14,66}$	14	4	1	4	0	0	0.60	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.52	0.11	0.22	0.26	0.34
$L_{14,67}$	15	3	1	2	0	0	0.59	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.46	0.07	0.05	0.07	0.34
$L_{14,68}$	15	4	1	2	0	0	0.61	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.50	0.07	0.05	0.06	0.35
$L_{14,69}$	15	4	1	2	0	0	0.58	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.48	0.06	0.05	0.07	0.36

Benchmarks L_{14} (continued on the next page).

$\mathscr{C}$	d	τ	S	E	V	A	Isot	IB	IG	IR	IF	IV	IA	IS	IC	Ins	Top	CA0	CA1	C2D
$L_{14,70}$	16	3	1	4	0	0	0.72	0.5	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.63	0.11	0.26	0.31	0.35
$L_{14,71}$	16	4	1	4	0	0	0.72	0.5	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.67	0.11	0.26	0.31	0.36
$L_{14,72}$	16	4	1	4	0	0	0.73	0.5	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.67	0.12	0.26	0.30	0.35
$L_{14,73}$	5	3	1	1	0	0	0.21	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.17	0.05	0.01	0.02	0.32
$L_{14,74}$	5	4	1	1	0	0	0.22	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.16	0.04	0.01	0.02	0.32
$L_{14,75}$	5	4	1	1	0	0	0.21	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.14	0.03	0.01	0.02	0.32
$L_{14,76}$	6	3	1	2	0	0	0.24	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.30	0.12	0.10	0.12	0.32
$L_{14,77}$	6	4	1	2	0	0	0.25	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.31	0.12	0.09	0.12	0.32
$L_{14,78}$	6	4	1	2	0	0	0.24	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.31	0.13	0.10	0.12	0.31
$L_{14,79}$	7	3	1	1	0	0	0.25	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.30	0.05	0.02	0.02	0.32
$L_{14,80}$	7	4	1	1	0	0	0.28	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.32	0.05	0.01	0.02	0.32
$L_{14,81}$	7	4	1	1	0	0	0.29	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.24	0.04	0.02	0.02	0.32
$L_{14,82}$	8	3	1	2	0	0	0.34	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.51	0.17	0.11	0.14	0.32
$L_{14,83}$	8	4	1	2	0	0	0.34	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.51	0.18	0.11	0.13	0.32
$L_{14,84}$	8	4	1	2	0	0	0.35	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.50	0.18	0.13	0.15	0.33
$L_{14,85}$	9	3	1	1	0	0	0.37	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.65	0.06	0.02	0.03	0.33
$L_{14,86}$	9	4	1	1	0	0	0.37	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.66	0.06	0.02	0.03	0.34
$L_{14,87}$	9	4	1	1	0	0	0.36	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.67	0.06	0.02	0.02	0.34
$L_{14,88}$	10	3	1	2	0	0	0.42	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.05	0.28	0.18	0.22	0.34
$L_{14,89}$	10	4	1	2	0	0	0.43	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.07	0.29	0.16	0.20	0.34
$L_{14,90}$	10	4	1	2	0	0	0.44	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.04	0.28	0.20	0.22	0.33
$L_{14,91}$	11	3	1	1	0	0	0.44	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.40	0.05	0.03	0.04	0.34
$L_{14,92}$	11	4	1	1	0	0	0.44	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.45	0.06	0.03	0.04	0.34
$L_{14,93}$	11	4	1	1	0	0	0.44	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.42	0.05	0.03	0.04	0.34
$L_{14,94}$	12	3	1	2	0	0	0.48	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.09	0.40	0.28	0.28	0.34
$L_{14,95}$	12	4	1	2	0	0	0.49	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.21	0.41	0.23	0.28	0.34
$L_{14,96}$	12	4	1	2	0	0	0.48	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.18	0.41	0.22	0.28	0.34
$L_{14,97}$	6	3	1	0	0	0	0.19	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.04	0.00	0.01	0.32
$L_{14,98}$	6	4	1	0	0	0	0.17	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.20	0.03	0.00	0.01	0.32
$L_{14,99}$	6	4	1	0	0	0	0.20	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.19	0.04	0.00	0.01	0.32
$L_{14,100}$	6	3	1	0	0	0	0.22	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.32	0.10	0.02	0.02	0.33
$L_{14,101}$	6	4	1	0	0	0	0.22	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.24	0.09	0.02	0.02	0.32
$L_{14,102}$	6	4	1	0	0	0	0.24	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.26	0.08	0.02	0.02	0.32
$L_{14,103}$	7	3	1	0	0	0	0.26	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.25	0.02	0.01	0.02	0.33
$L_{14,104}$	7	4	1	0	0	0	0.26	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.26	0.04	0.01	0.02	0.33
$L_{14,105}$	7	4	1	0	0	0	0.26	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.27	0.03	0.01	0.02	0.32
$L_{14,106}$	8	3	1	0	0	0	0.31	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.49	0.04	0.01	0.02	0.32
$L_{14,107}$	8	4	1	0	0	0	0.33	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.48	0.03	0.01	0.02	0.32
$L_{14,108}$	8	4	1	0	0	0	0.30	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.52	0.03	0.01	0.02	0.33

Benchmarks L_{14} (continued on the next page).

$\mathscr{C}$	d	τ	S	E	V	A	ISOT	IB	IG	IR	IF	IV	IA	IS	IC	INS	TOP	CA0	CA1	C2D
$L_{14,109}$	9	3	1	0	0	0	0.36	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.78	0.04	0.02	0.03	0.33
$L_{14,110}$	9	4	1	0	0	0	0.34	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.80	0.03	0.02	0.03	0.33
$L_{14,111}$	9	4	1	0	0	0	0.35	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.81	0.03	0.02	0.02	0.34
$L_{14,112}$	10	3	1	0	0	0	0.42	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.94	0.15	0.04	0.05	0.34
$L_{14,113}$	10	4	1	0	0	0	0.40	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.04	0.18	0.03	0.05	0.33
$L_{14,114}$	10	4	1	0	0	0	0.42	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.08	0.16	0.03	0.05	0.34
$L_{14,115}$	11	3	1	0	0	0	0.44	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.56	0.02	0.03	0.04	0.34
$L_{14,116}$	11	4	1	0	0	0	0.44	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.69	0.04	0.02	0.04	0.34
$L_{14,117}$	11	4	1	0	0	0	0.43	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.70	0.03	0.02	0.04	0.35
$L_{14,118}$	12	3	1	0	0	0	0.50	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.04	0.04	0.02	0.04	0.34
$L_{14,119}$	12	4	1	0	0	0	0.50	0.4	0.1	0.0	0.0	0.0	0.0	0.0	0.0	2.15	0.05	0.02	0.04	0.34
$L_{14,120}$	12	4	1	0	0	0	0.50	0.4	0.1	0.0	0.0	0.0	0.0	0.0	0.0	2.22	0.04	0.02	0.04	0.34
$L_{14,121}$	7	3	1	1	0	0	0.22	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.25	0.04	0.01	0.02	0.32
$L_{14,122}$	7	4	1	1	0	0	0.19	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.35	0.05	0.01	0.02	0.32
$L_{14,123}$	7	4	1	1	0	0	0.20	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.36	0.04	0.01	0.02	0.32
$L_{14,124}$	7	3	1	2	0	0	0.25	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.52	0.20	0.14	0.17	0.32
$L_{14,125}$	7	4	1	2	0	0	0.24	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.52	0.19	0.12	0.16	0.33
$L_{14,126}$	7	4	1	2	0	0	0.26	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.50	0.19	0.12	0.15	0.33
$L_{14,127}$	7	3	1	1	0	0	0.28	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.41	0.05	0.02	0.03	0.34
$L_{14,128}$	7	4	1	1	0	0	0.27	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.42	0.05	0.01	0.03	0.33
$L_{14,129}$	7	4	1	1	0	0	0.28	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.41	0.05	0.02	0.03	0.32
$L_{14,130}$	8	3	1	2	0	0	0.35	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.93	0.28	0.22	0.22	0.32
$L_{14,131}$	8	4	1	2	0	0	0.33	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.90	0.30	0.17	0.22	0.36
$L_{14,132}$	8	4	1	2	0	0	0.34	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.94	0.30	0.21	0.26	0.34
$L_{14,133}$	9	3	1	1	0	0	0.38	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.17	0.05	0.02	0.05	0.35
$L_{14,134}$	9	4	1	1	0	0	0.38	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.24	0.05	0.03	0.04	0.34
$L_{14,135}$	9	4	1	1	0	0	0.36	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.22	0.05	0.03	0.04	0.34
$L_{14,136}$	10	3	1	2	0	0	0.43	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.90	0.21	0.26	0.32	0.34
$L_{14,137}$	10	4	1	2	0	0	0.43	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.00	0.22	0.25	0.35	0.33
$L_{14,138}$	10	4	1	2	0	0	0.43	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.98	0.22	0.25	0.34	0.35
$L_{14,139}$	11	3	1	1	0	0	0.45	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.76	0.07	0.06	0.07	0.35
$L_{14,140}$	11	4	1	1	0	0	0.46	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.77	0.05	0.05	0.07	0.37
$L_{14,141}$	11	4	1	1	0	0	0.46	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.76	0.06	0.05	0.07	0.35
$L_{14,142}$	12	3	1	2	0	0	0.51	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.80	0.85	0.42	0.50	0.35
$L_{14,143}$	12	4	1	2	0	0	0.52	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.93	0.89	0.41	0.46	0.39
$L_{14,144}$	12	4	1	2	0	0	0.51	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.94	0.87	0.35	0.44	0.35
$L_{14,145}$	8	3	1	0	0	0	0.22	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.04	0.01	0.01	0.32
$L_{14,146}$	8	4	1	0	0	0	0.20	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.41	0.03	0.01	0.01	0.32
$L_{14,147}$	8	4	1	0	0	0	0.21	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.39	0.04	0.01	0.02	0.32

Benchmarks L_{14} (continued on the next page).

$\mathscr{C}$	d	τ	S	E	V	A	ISOT	IB	IG	IR	IF	IV	IA	IS	IC	INS	TOP	CA0	CA1	C2D
$L_{14,148}$	8	3	1	0	0	0	0.26	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.65	0.15	0.02	0.03	0.32
$L_{14,149}$	8	4	1	0	0	0	0.25	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.66	0.15	0.02	0.03	0.33
$L_{14,150}$	8	4	1	0	0	0	0.25	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.69	0.16	0.02	0.03	0.33
$L_{14,151}$	8	3	1	0	0	0	0.26	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.70	0.04	0.02	0.02	0.34
$L_{14,152}$	8	4	1	0	0	0	0.28	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.67	0.03	0.01	0.02	0.33
$L_{14,153}$	8	4	1	0	0	0	0.26	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.69	0.04	0.01	0.02	0.34
$L_{14,154}$	8	3	1	0	0	0	0.32	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.82	0.04	0.01	0.03	0.33
$L_{14,155}$	8	4	1	0	0	0	0.34	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.62	0.03	0.01	0.02	0.32
$L_{14,156}$	8	4	1	0	0	0	0.34	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.68	0.03	0.01	0.03	0.33
$L_{14,157}$	9	3	1	0	0	0	0.37	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.35	0.04	0.03	0.04	0.34
$L_{14,158}$	9	4	1	0	0	0	0.36	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.40	0.03	0.02	0.04	0.34
$L_{14,159}$	9	4	1	0	0	0	0.38	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.40	0.04	0.02	0.04	0.35
$L_{14,160}$	10	3	1	0	0	0	0.47	0.3	0.1	0.0	0.0	0.0	0.0	0.0	0.0	1.95	0.30	0.05	0.07	0.34
$L_{14,161}$	10	4	1	0	0	0	0.45	0.3	0.1	0.0	0.0	0.0	0.0	0.0	0.0	1.99	0.30	0.04	0.06	0.36
$L_{14,162}$	10	4	1	0	0	0	0.48	0.3	0.1	0.0	0.0	0.0	0.0	0.0	0.0	1.96	0.28	0.04	0.06	0.34
$L_{14,163}$	11	3	1	0	0	0	0.47	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.38	0.03	0.05	0.08	0.36
$L_{14,164}$	11	4	1	0	0	0	0.45	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.36	0.04	0.03	0.06	0.36
$L_{14,165}$	11	4	1	0	0	0	0.46	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.50	0.03	0.04	0.07	0.36
$L_{14,166}$	12	3	1	0	0	0	0.61	0.4	0.1	0.0	0.0	0.0	0.0	0.0	0.0	4.66	0.04	0.02	0.05	0.36
$L_{14,167}$	12	4	1	0	0	0	0.62	0.4	0.1	0.0	0.0	0.0	0.0	0.0	0.0	4.61	0.04	0.03	0.05	0.36
$L_{14,168}$	12	4	1	0	0	0	0.64	0.4	0.1	0.0	0.0	0.0	0.0	0.0	0.0	4.63	0.04	0.03	0.05	0.35
$L_{14,169}$	9	3	1	1	0	0	0.23	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.75	0.05	0.01	0.02	0.33
$L_{14,170}$	9	4	1	1	0	0	0.22	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.73	0.04	0.01	0.02	0.32
$L_{14,171}$	9	4	1	1	0	0	0.21	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.76	0.04	0.02	0.02	0.33
$L_{14,172}$	9	3	1	2	0	0	0.25	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.06	0.33	0.26	0.27	0.32
$L_{14,173}$	9	4	1	2	0	0	0.27	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.04	0.32	0.18	0.25	0.32
$L_{14,174}$	9	4	1	2	0	0	0.27	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.07	0.34	0.18	0.26	0.34
$L_{14,175}$	9	3	1	1	0	0	0.33	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.18	0.05	0.03	0.04	0.34
$L_{14,176}$	9	4	1	1	0	0	0.29	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.18	0.05	0.03	0.04	0.33
$L_{14,177}$	9	4	1	1	0	0	0.32	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.24	0.05	0.02	0.04	0.33
$L_{14,178}$	9	3	1	2	0	0	0.36	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.67	0.55	0.34	0.43	0.34
$L_{14,179}$	9	4	1	2	0	0	0.36	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.52	0.56	0.26	0.34	0.34
$L_{14,180}$	9	4	1	2	0	0	0.36	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.68	0.57	0.34	0.43	0.33
$L_{14,181}$	9	3	1	1	0	0	0.40	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.68	0.06	0.04	0.07	0.34
$L_{14,182}$	9	4	1	1	0	0	0.38	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.65	0.06	0.04	0.06	0.34
$L_{14,183}$	9	4	1	1	0	0	0.41	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.63	0.06	0.04	0.06	0.34
$L_{14,184}$	10	3	1	2	0	0	0.44	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.96	0.93	0.53	0.64	0.34
$L_{14,185}$	10	4	1	2	0	0	0.47	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.09	0.94	0.48	0.59	0.34
$L_{14,186}$	10	4	1	2	0	0	0.47	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.18	0.94	0.52	0.64	0.36

Benchmarks L_{14} (continued on the next page).

$\mathscr{C}$	d	τ	S	E	V	A	ISOT	IB	IG	IR	IF	IV	IA	IS	IC	INS	TOP	CA0	CA1	C2D
$L_{14,187}$	11	3	1	1	0	0	0.49	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.75	0.05	0.09	0.12	0.38
$L_{14,188}$	11	4	1	1	0	0	0.50	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.03	0.06	0.08	0.10	0.39
$L_{14,189}$	11	4	1	1	0	0	0.50	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.95	0.06	0.07	0.12	0.39
$L_{14,190}$	12	3	1	2	0	0	0.55	0.3	0.1	0.0	0.0	0.0	0.0	0.0	0.0	6.22	1.66	0.64	0.85	0.37
$L_{14,191}$	12	4	1	2	0	0	0.55	0.3	0.1	0.0	0.0	0.0	0.0	0.0	0.0	6.36	1.65	0.68	0.97	0.37
$L_{14,192}$	12	4	1	2	0	0	0.54	0.3	0.1	0.0	0.0	0.0	0.0	0.0	0.0	6.64	1.72	0.78	0.79	0.37
$L_{14,193}$	10	3	1	0	0	0	0.19	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.94	0.02	0.01	0.02	0.32
$L_{14,194}$	10	4	1	0	0	0	0.20	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.92	0.04	0.01	0.02	0.32
$L_{14,195}$	10	4	1	0	0	0	0.19	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.92	0.04	0.00	0.02	0.32
$L_{14,196}$	10	3	1	0	0	0	0.25	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.18	0.19	0.03	0.04	0.32
$L_{14,197}$	10	4	1	0	0	0	0.25	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.19	0.20	0.02	0.04	0.33
$L_{14,198}$	10	4	1	0	0	0	0.24	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.22	0.21	0.02	0.04	0.33
$L_{14,199}$	10	3	1	0	0	0	0.26	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.54	0.03	0.02	0.04	0.33
$L_{14,200}$	10	4	1	0	0	0	0.26	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.56	0.04	0.02	0.04	0.36
$L_{14,201}$	10	4	1	0	0	0	0.27	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.63	0.03	0.02	0.04	0.34
$L_{14,202}$	10	3	1	0	0	0	0.42	0.2	0.1	0.0	0.0	0.0	0.0	0.0	0.0	2.00	0.03	0.02	0.03	0.33
$L_{14,203}$	10	4	1	0	0	0	0.39	0.2	0.1	0.0	0.0	0.0	0.0	0.0	0.0	2.14	0.03	0.01	0.03	0.33
$L_{14,204}$	10	4	1	0	0	0	0.40	0.2	0.1	0.0	0.0	0.0	0.0	0.0	0.0	2.14	0.04	0.02	0.03	0.34
$L_{14,205}$	10	3	1	0	0	0	0.38	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.04	0.04	0.04	0.06	0.37
$L_{14,206}$	10	4	1	0	0	0	0.38	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.24	0.04	0.04	0.05	0.36
$L_{14,207}$	10	4	1	0	0	0	0.38	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.25	0.03	0.04	0.06	0.36
$L_{14,208}$	10	3	1	0	0	0	0.56	0.3	0.2	0.0	0.0	0.0	0.0	0.0	0.0	2.81	0.21	0.06	0.09	0.35
$L_{14,209}$	10	4	1	0	0	0	0.55	0.3	0.2	0.0	0.0	0.0	0.0	0.0	0.0	2.56	0.16	0.06	0.09	0.35
$L_{14,210}$	10	4	1	0	0	0	0.57	0.3	0.2	0.0	0.0	0.0	0.0	0.0	0.0	2.55	0.16	0.06	0.09	0.34
$L_{14,211}$	11	3	1	0	0	0	0.47	0.3	0.1	0.0	0.0	0.0	0.0	0.0	0.0	6.69	0.04	0.08	0.10	0.41
$L_{14,212}$	11	4	1	0	0	0	0.46	0.3	0.1	0.0	0.0	0.0	0.0	0.0	0.0	6.43	0.04	0.08	0.11	0.41
$L_{14,213}$	11	4	1	0	0	0	0.47	0.3	0.1	0.0	0.0	0.0	0.0	0.0	0.0	6.56	0.04	0.06	0.09	0.41
$L_{14,214}$	12	3	1	0	0	0	0.84	0.4	0.4	0.0	0.0	0.0	0.0	0.0	0.0	5.49	0.04	0.04	0.08	0.38
$L_{14,215}$	12	4	1	0	0	0	0.83	0.4	0.3	0.0	0.0	0.0	0.0	0.0	0.0	5.76	0.04	0.03	0.07	0.38
$L_{14,216}$	12	4	1	0	0	0	0.83	0.4	0.4	0.0	0.0	0.0	0.0	0.0	0.0	5.72	0.03	0.03	0.07	0.37
$L_{14,217}$	11	3	1	1	0	0	0.24	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.55	0.04	0.02	0.02	0.34
$L_{14,218}$	11	4	1	1	0	0	0.23	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.53	0.04	0.02	0.03	0.32
$L_{14,219}$	11	4	1	1	0	0	0.23	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.55	0.05	0.02	0.02	0.33
$L_{14,220}$	11	3	1	2	0	0	0.28	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.99	0.50	0.33	0.43	0.32
$L_{14,221}$	11	4	1	2	0	0	0.29	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.00	0.51	0.33	0.47	0.34
$L_{14,222}$	11	4	1	2	0	0	0.25	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.04	0.50	0.28	0.42	0.33
$L_{14,223}$	11	3	1	1	0	0	0.31	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.32	0.05	0.04	0.06	0.36
$L_{14,224}$	11	4	1	1	0	0	0.34	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.42	0.05	0.04	0.05	0.35
$L_{14,225}$	11	4	1	1	0	0	0.32	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.55	0.04	0.03	0.05	0.34

Benchmarks L_{14} (continued on the next page).

$\mathscr{C}$	d	τ	S	E	V	A	Isot	Ib	Ig	Ir	If	Iv	Ia	Is	Ic	Ins	Top	CA0	CA1	C2d
$L_{14,226}$	11	3	1	2	0	0	0.35	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.35	0.95	0.39	0.54	0.34
$L_{14,227}$	11	4	1	2	0	0	0.38	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.49	0.97	0.42	0.60	0.33
$L_{14,228}$	11	4	1	2	0	0	0.40	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.50	0.98	0.46	0.58	0.34
$L_{14,229}$	11	3	1	1	0	0	0.42	0.2	0.1	0.0	0.0	0.0	0.0	0.0	0.0	3.85	0.06	0.06	0.09	0.38
$L_{14,230}$	11	4	1	1	0	0	0.44	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.93	0.06	0.07	0.10	0.39
$L_{14,231}$	11	4	1	1	0	0	0.44	0.3	0.1	0.0	0.0	0.0	0.0	0.0	0.0	3.98	0.05	0.07	0.10	0.38
$L_{14,232}$	11	3	1	2	0	0	0.49	0.3	0.1	0.0	0.0	0.0	0.0	0.0	0.0	5.43	1.72	0.66	0.80	0.36
$L_{14,233}$	11	4	1	2	0	0	0.50	0.3	0.1	0.0	0.0	0.0	0.0	0.0	0.0	5.59	1.70	0.60	0.86	0.36
$L_{14,234}$	11	4	1	2	0	0	0.49	0.3	0.1	0.0	0.0	0.0	0.0	0.0	0.0	5.50	1.73	0.61	0.82	0.37
$L_{14,235}$	11	3	1	1	0	0	0.54	0.3	0.1	0.0	0.0	0.0	0.0	0.0	0.0	5.06	0.07	0.12	0.15	0.40
$L_{14,236}$	11	4	1	1	0	0	0.53	0.3	0.1	0.0	0.0	0.0	0.0	0.0	0.0	4.93	0.07	0.12	0.17	0.42
$L_{14,237}$	11	4	1	1	0	0	0.52	0.3	0.1	0.0	0.0	0.0	0.0	0.0	0.0	4.88	0.07	0.12	0.15	0.41
$L_{14,238}$	12	3	1	2	0	0	0.61	0.3	0.1	0.0	0.0	0.0	0.0	0.0	0.0	7.92	2.68	1.64	1.85	0.39
$L_{14,239}$	12	4	1	2	0	0	0.58	0.3	0.1	0.0	0.0	0.0	0.0	0.0	0.0	8.25	2.72	1.82	1.51	0.39
$L_{14,240}$	12	4	1	2	0	0	0.60	0.3	0.1	0.0	0.0	0.0	0.0	0.0	0.0	8.44	2.75	1.78	1.87	0.39
$L_{14,241}$	12	3	1	0	0	0	0.21	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.95	0.04	0.01	0.02	0.33
$L_{14,242}$	12	4	1	0	0	0	0.20	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.96	0.02	0.01	0.02	0.33
$L_{14,243}$	12	4	1	0	0	0	0.19	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.01	0.02	0.01	0.02	0.33
$L_{14,244}$	12	3	1	0	0	0	0.29	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.49	0.28	0.03	0.05	0.32
$L_{14,245}$	12	4	1	0	0	0	0.27	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.32	0.30	0.03	0.04	0.33
$L_{14,246}$	12	4	1	0	0	0	0.29	0.2	0.1	0.0	0.0	0.0	0.0	0.0	0.0	2.38	0.30	0.03	0.04	0.33
$L_{14,247}$	12	3	1	0	0	0	0.28	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.28	0.03	0.03	0.05	0.35
$L_{14,248}$	12	4	1	0	0	0	0.29	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.20	0.03	0.03	0.05	0.35
$L_{14,249}$	12	4	1	0	0	0	0.26	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.28	0.04	0.04	0.05	0.35
$L_{14,250}$	12	3	1	0	0	0	0.49	0.2	0.1	0.0	0.0	0.0	0.0	0.0	0.0	4.34	0.02	0.02	0.04	0.34
$L_{14,251}$	12	4	1	0	0	0	0.46	0.2	0.1	0.0	0.0	0.0	0.0	0.0	0.0	4.30	0.03	0.02	0.04	0.34
$L_{14,252}$	12	4	1	0	0	0	0.47	0.2	0.1	0.0	0.0	0.0	0.0	0.0	0.0	4.37	0.03	0.02	0.04	0.36
$L_{14,253}$	12	3	1	0	0	0	0.41	0.3	0.1	0.0	0.0	0.0	0.0	0.0	0.0	5.15	0.04	0.06	0.08	0.36
$L_{14,254}$	12	4	1	0	0	0	0.43	0.3	0.1	0.0	0.0	0.0	0.0	0.0	0.0	5.40	0.04	0.06	0.10	0.36
$L_{14,255}$	12	4	1	0	0	0	0.40	0.3	0.1	0.0	0.0	0.0	0.0	0.0	0.0	5.70	0.03	0.06	0.08	0.36
$L_{14,256}$	12	3	1	0	0	0	0.74	0.3	0.3	0.0	0.0	0.0	0.0	0.0	0.0	6.27	0.92	0.08	0.11	0.37
$L_{14,257}$	12	4	1	0	0	0	0.73	0.3	0.3	0.0	0.0	0.0	0.0	0.0	0.0	5.93	0.90	0.07	0.12	0.38
$L_{14,258}$	12	4	1	0	0	0	0.77	0.3	0.4	0.0	0.0	0.0	0.0	0.0	0.0	6.03	0.92	0.08	0.12	0.36
$L_{14,259}$	12	3	1	0	0	0	0.52	0.3	0.1	0.0	0.0	0.0	0.0	0.0	0.0	7.76	0.04	0.11	0.16	0.44
$L_{14,260}$	12	4	1	0	0	0	0.52	0.3	0.1	0.0	0.0	0.0	0.0	0.0	0.0	7.73	0.04	0.12	0.18	0.46
$L_{14,261}$	12	4	1	0	0	0	0.51	0.3	0.1	0.0	0.0	0.0	0.0	0.0	0.0	8.00	0.04	0.10	0.14	0.46
$L_{14,262}$	12	3	1	0	0	0	1.08	0.4	0.6	0.0	0.0	0.0	0.0	0.0	0.0	7.31	0.04	0.04	0.10	0.39
$L_{14,263}$	12	4	1	0	0	0	1.08	0.4	0.6	0.0	0.0	0.0	0.0	0.0	0.0	7.08	0.04	0.04	0.09	0.40
$L_{14,264}$	12	4	1	0	0	0	1.07	0.4	0.6	0.0	0.0	0.0	0.0	0.0	0.0	6.97	0.04	0.05	0.10	0.40

Benchmarks L_{14} (continued on the next page).

$\mathscr{C}$	d	τ	S	E	V	A	ISOT	IB	IG	IR	IF	IV	IA	IS	IC	INS	TOP	CA0	CA1	C2D
$L_{14,265}$	4	4	0	2	0	0	0.08	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.03	0.01	0.01	0.70
$L_{14,266}$	4	4	0	2	0	0	0.08	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.04	0.01	0.01	0.70
$L_{14,267}$	4	4	0	2	0	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.04	0.01	0.01	0.70
$L_{14,268}$	4	4	1	2	0	0	0.17	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.02	0.02	0.02	0.84
$L_{14,269}$	4	4	1	2	0	0	0.18	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.04	0.02	0.02	0.84
$L_{14,270}$	4	4	1	2	0	0	0.16	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.04	0.01	0.01	0.85
$L_{14,271}$	4	4	1	1	0	0	0.18	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.04	0.01	0.01	0.33
$L_{14,272}$	4	4	1	1	0	0	0.17	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.03	0.01	0.01	0.32
$L_{14,273}$	4	4	1	1	0	0	0.16	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.03	0.01	0.01	0.32
$L_{14,274}$	4	4	1	0	0	0	0.15	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.02	0.00	0.01	0.32
$L_{14,275}$	4	4	1	0	0	0	0.16	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.02	0.00	0.00	0.32
$L_{14,276}$	4	4	1	0	0	0	0.16	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.02	0.00	0.01	0.32
$L_{14,277}$	5	4	1	1	0	0	0.18	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.16	0.04	0.00	0.01	0.32
$L_{14,278}$	5	4	1	1	0	0	0.17	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.16	0.03	0.01	0.01	0.33
$L_{14,279}$	5	4	1	1	0	0	0.18	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.15	0.03	0.01	0.01	0.32
$L_{14,280}$	6	4	1	2	0	0	0.18	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.26	0.04	0.02	0.01	0.86
$L_{14,281}$	6	4	1	2	0	0	0.18	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.24	0.04	0.01	0.02	0.84
$L_{14,282}$	6	4	1	2	0	0	0.19	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.24	0.03	0.01	0.02	0.84
$L_{14,283}$	7	4	1	1	0	0	0.18	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.24	0.04	0.00	0.02	0.70
$L_{14,284}$	7	4	1	1	0	0	0.16	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.24	0.05	0.01	0.02	0.71
$L_{14,285}$	7	4	1	1	0	0	0.18	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.23	0.04	0.01	0.01	0.79
$L_{14,286}$	8	4	1	2	0	0	0.19	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.36	0.05	0.01	0.02	1.34
$L_{14,287}$	8	4	1	2	0	0	0.18	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.39	0.04	0.01	0.02	1.32
$L_{14,288}$	8	4	1	2	0	0	0.19	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.38	0.04	0.02	0.02	1.34
$L_{14,289}$	9	4	1	1	0	0	0.18	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.32	0.03	0.01	0.02	0.95
$L_{14,290}$	9	4	1	1	0	0	0.19	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.32	0.05	0.00	0.02	0.96
$L_{14,291}$	9	4	1	1	0	0	0.20	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.32	0.05	0.01	0.02	0.95
$L_{14,292}$	10	4	1	2	0	0	0.21	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.47	0.04	0.02	0.02	1.82
$L_{14,293}$	10	4	1	2	0	0	0.19	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.48	0.04	0.01	0.02	1.81
$L_{14,294}$	10	4	1	2	0	0	0.20	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.48	0.05	0.01	0.02	1.77
$L_{14,295}$	11	4	1	1	0	0	0.20	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.40	0.04	0.01	0.02	1.19
$L_{14,296}$	11	4	1	1	0	0	0.18	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.41	0.05	0.01	0.02	1.34
$L_{14,297}$	11	4	1	1	0	0	0.18	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.44	0.04	0.01	0.02	1.18
$L_{14,298}$	12	4	1	2	0	0	0.19	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.59	0.05	0.02	0.02	2.63
$L_{14,299}$	12	4	1	2	0	0	0.20	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.60	0.04	0.01	0.02	2.20
$L_{14,300}$	12	4	1	2	0	0	0.19	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.61	0.03	0.02	0.02	2.26

Benchmarks L_{14}.

$\mathscr{C}$	d	τ	S	E	V	A	ISOT	IB	IG	IR	IF	IV	IA	IS	IC	INS	TOP	CA0	CA1	C2D
$L_{15,1}$	90	176	16	0	0	1	ERR	ERR	ERR	ERR	ERR	ERR	ERR	ERR	ERR	X	X	INT	INT	ERR

Benchmarks L_{15}.

$\mathscr{C}$	d	τ	S	E	V	A	ISOT	IB	IG	IR	IF	IV	IA	IS	IC	INS	TOP	CA0	CA1	C2D
$L_{16,1}$	9	3	4	0	0	0	0.31	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.0	2.31	0.86	0.31	0.38	0.36
$L_{16,2}$	17	7	4	0	0	0	1.87	0.2	1.1	0.0	0.0	0.0	0.1	0.1	0.0	106.49	X	6.04	11.70	1.28
$L_{16,3}$	25	10	4	0	0	0	15.80	0.5	13.6	0.0	0.0	0.0	0.3	0.4	0.0	INT	X	156.00	106.00	10.66
$L_{16,4}$	33	14	4	0	0	0	89.29	1.9	83.8	0.0	0.0	0.0	0.8	0.7	0.0	INT	X	2120.00	1320.00	57.88
$L_{16,5}$	41	18	4	0	0	0	351.01	3.9	339.0	0.0	0.0	0.0	1.9	1.2	0.0	X	X	11200.00	12300.00	227.09
$L_{16,6}$	49	22	4	0	0	0	1143.58	7.9	1115.5	0.0	0.0	0.0	4.4	2.4	0.0	X	X	INT	INT	603.54
$L_{16,7}$	17	3	4	0	0	0	1.79	0.1	0.9	0.0	0.0	0.0	0.0	0.2	0.0	76.34	X	11.80	10.20	1.23
$L_{16,8}$	33	7	4	0	0	0	62.29	0.3	58.0	0.0	0.0	0.0	0.4	0.9	0.0	X	INT	2000.00	2170.00	45.76
$L_{16,9}$	49	10	4	0	0	0	803.51	0.8	788.4	0.0	0.0	0.0	2.4	2.1	0.0	X	X	INT	INT	469.25
$L_{16,10}$	65	14	4	0	0	0	ERR	ERR	ERR	ERR	ERR	ERR	ERR	ERR	ERR	X	595.21	INT	INT	2505.38
$L_{16,11}$	81	18	4	0	0	0	INT	INT	INT	INT	INT	INT	INT	INT	INT	INT	X	INT	INT	ERR
$L_{16,12}$	97	22	4	0	0	0	INT	INT	INT	INT	INT	INT	INT	INT	INT	X	X	INT	INT	ERR
$L_{16,13}$	25	3	4	0	0	0	12.01	0.1	9.6	0.0	0.0	0.0	0.1	1.0	0.0	X	X	INT	INT	ERR

Benchmarks L_{16}.

$\mathscr{C}$	d	τ	S	E	V	A	ISOT	IB	IG	IR	IF	IV	IA	IS	IC	INS	TOP	CA0	CA1	C2D
$L_{17,1}$	9	16	2	0	0	0	0.28	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.08	0.15	0.06	0.10	0.42
$L_{17,2}$	9	30	2	0	0	0	0.31	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.51	0.20	0.06	0.12	0.46
$L_{17,3}$	9	43	2	0	0	0	0.34	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.18	0.24	0.08	0.14	0.51
$L_{17,4}$	9	56	2	0	0	0	0.28	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.66	0.26	0.09	0.16	0.57
$L_{17,5}$	9	70	2	0	0	0	0.31	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.23	0.28	0.11	0.18	0.66
$L_{17,6}$	9	16	6	0	0	0	0.50	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.0	2.44	1.04	0.38	0.48	0.42
$L_{17,7}$	9	30	6	0	0	0	0.50	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.0	2.87	1.33	0.51	0.63	0.46
$L_{17,8}$	9	43	6	0	0	0	0.57	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.0	3.51	1.57	0.54	0.81	0.52
$L_{17,9}$	9	56	6	0	0	0	0.59	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.0	3.94	1.85	0.74	0.90	0.59
$L_{17,10}$	9	70	6	0	0	0	0.62	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.0	4.60	2.12	0.82	1.04	0.64
$L_{17,11}$	17	33	2	0	0	0	1.98	0.2	1.0	0.0	0.1	0.0	0.2	0.1	0.0	102.57	12.55	1.69	2.90	3.83
$L_{17,12}$	17	60	2	0	0	0	2.37	0.2	1.1	0.0	0.0	0.0	0.3	0.1	0.0	151.90	20.77	3.43	5.43	7.37
$L_{17,13}$	17	86	2	0	0	0	3.04	0.2	1.3	0.0	0.2	0.0	0.5	0.1	0.0	206.64	29.20	5.68	8.49	10.23
$L_{17,14}$	17	113	2	0	0	0	INT	INT	INT	INT	INT	INT	INT	INT	INT	260.16	38.17	8.68	12.50	14.27
$L_{17,15}$	17	140	2	0	0	0	INT	INT	INT	INT	INT	INT	INT	INT	INT	314.01	47.25	11.80	16.80	17.71
$L_{17,16}$	17	33	6	0	0	0	2.45	0.2	1.0	0.0	0.1	0.0	0.2	0.2	0.0	108.97	132.81	20.00	18.90	3.75

Benchmarks L_{17} (continued on the next page).

$\mathscr{C}$	d	τ	S	E	V	A	Isot	Ib	Ig	Ir	If	Iv	Ia	Is	Ic	Ins	Top	CA0	CA1	C2d
$L_{17,17}$	17	60	6	0	0	0	2.85	0.2	1.1	0.0	0.1	0.0	0.4	0.2	0.0	159.06	213.51	34.20	35.50	7.36
$L_{17,18}$	17	86	6	0	0	0	3.61	0.2	1.3	0.0	0.2	0.0	0.5	0.3	0.0	211.97	301.29	59.70	64.50	10.23
$L_{17,19}$	17	113	6	0	0	0	INT	INT	INT	INT	INT	INT	INT	INT	INT	260.70	387.83	80.40	93.80	14.23
$L_{17,20}$	17	140	6	0	0	0	INT	INT	INT	INT	INT	INT	INT	INT	INT	318.76	468.95	106.00	108.00	17.74
$L_{17,21}$	17	16	2	0	0	0	1.52	0.1	0.8	0.0	0.0	0.0	0.1	0.1	0.0	51.78	1.01	0.60	1.26	2.33
$L_{17,22}$	17	30	2	0	0	0	1.67	0.1	0.9	0.0	0.0	0.0	0.1	0.1	0.0	61.90	1.51	0.72	1.53	3.20
$L_{17,23}$	17	43	2	0	0	0	1.87	0.1	0.9	0.0	0.1	0.0	0.1	0.1	0.0	76.47	2.00	0.92	1.89	4.40
$L_{17,24}$	17	56	2	0	0	0	INT	INT	INT	INT	INT	INT	INT	INT	INT	89.03	2.46	1.06	2.16	6.47
$L_{17,25}$	17	70	2	0	0	0	INT	INT	INT	INT	INT	INT	INT	INT	INT	100.44	2.94	1.26	2.60	7.60
$L_{17,26}$	17	16	6	0	0	0	2.23	0.1	0.8	0.0	0.1	0.0	0.1	0.2	0.0	63.53	X	11.00	21.00	2.40
$L_{17,27}$	17	30	6	0	0	0	2.30	0.1	0.8	0.0	0.0	0.0	0.1	0.2	0.0	66.58	595.12	22.20	25.40	3.50
$L_{17,28}$	17	43	6	0	0	0	2.59	0.1	0.9	0.0	0.1	0.0	0.1	0.3	0.0	79.66	X	31.10	31.90	4.60
$L_{17,29}$	17	56	6	0	0	0	INT	INT	INT	INT	INT	INT	INT	INT	INT	91.25	X	36.40	43.80	6.67
$L_{17,30}$	17	70	6	0	0	0	INT	INT	INT	INT	INT	INT	INT	INT	INT	106.24	X	46.40	42.70	7.50
$L_{17,31}$	33	33	2	0	0	0	54.44	0.2	48.6	0.0	1.0	0.0	1.0	0.3	0.0	INT	154.33	28.50	68.50	111.29
$L_{17,32}$	33	60	2	0	0	0	63.70	0.3	54.4	0.0	1.9	0.0	1.2	0.3	0.0	INT	282.62	62.30	120.00	181.95
$L_{17,33}$	33	86	2	0	0	0	74.50	0.3	61.0	0.0	3.5	0.0	1.5	0.4	0.0	X	409.53	106.00	185.00	251.79
$L_{17,34}$	33	113	2	0	0	0	INT	INT	INT	INT	INT	INT	INT	INT	INT	X	543.52	154.00	256.00	326.43
$L_{17,35}$	33	140	2	0	0	0	INT	INT	INT	INT	INT	INT	INT	INT	INT	X	X	217.00	344.00	404.08
$L_{17,36}$	33	33	6	0	0	0	58.05	0.2	48.8	0.0	1.1	0.0	1.0	0.9	0.0	X	X	3370.00	INT	121.17
$L_{17,37}$	33	60	6	0	0	0	66.29	0.3	54.5	0.0	2.0	0.0	1.2	1.0	0.0	X	X	INT	INT	180.65
$L_{17,38}$	33	86	6	0	0	0	77.90	0.3	61.5	0.0	3.5	0.0	1.6	1.0	0.0	INT	X	INT	INT	252.22
$L_{17,39}$	33	113	6	0	0	0	INT	INT	INT	INT	INT	INT	INT	INT	INT	X	X	INT	INT	326.28
$L_{17,40}$	33	140	6	0	0	0	INT	INT	INT	INT	INT	INT	INT	INT	INT	INT	X	INT	INT	407.72
$L_{17,41}$	25	16	2	0	0	0	9.73	0.1	7.8	0.0	0.2	0.0	0.2	0.1	0.0	X	3.52	4.63	8.61	18.22
$L_{17,42}$	25	30	2	0	0	0	10.33	0.1	8.4	0.0	0.2	0.0	0.3	0.1	0.0	X	5.38	5.02	10.10	25.90
$L_{17,43}$	25	43	2	0	0	0	11.87	0.1	9.0	0.0	0.4	0.0	0.3	0.2	0.0	X	7.13	5.74	11.60	33.89
$L_{17,44}$	25	56	2	0	0	0	11.87	0.1	9.4	0.0	0.4	0.0	0.3	0.2	0.0	X	9.09	6.55	13.70	42.67
$L_{17,45}$	25	70	2	0	0	0	INT	INT	INT	INT	INT	INT	INT	INT	INT	X	11.36	7.46	15.90	48.77
$L_{17,46}$	25	16	6	0	0	0	11.46	0.1	7.9	0.0	0.2	0.0	0.2	0.5	0.0	X	INT	323.00	353.00	18.77
$L_{17,47}$	25	30	6	0	0	0	12.11	0.1	8.3	0.0	0.2	0.0	0.2	0.5	0.0	X	INT	499.00	439.00	24.80
$L_{17,48}$	25	43	6	0	0	0	13.93	0.1	8.8	0.0	0.4	0.0	0.3	0.5	0.0	X	X	605.00	733.00	34.26
$L_{17,49}$	25	56	6	0	0	0	14.16	0.1	9.2	0.0	0.4	0.0	0.3	0.7	0.0	X	411.80	784.00	908.00	42.54
$L_{17,50}$	25	70	6	0	0	0	INT	INT	INT	INT	INT	INT	INT	INT	INT	X	X	923.00	892.00	50.81
$L_{17,51}$	49	33	2	0	0	0	596.97	0.3	564.1	0.0	7.3	0.0	5.1	0.5	0.0	X	X	167.00	495.00	918.56
$L_{17,52}$	49	60	2	0	0	0	662.72	0.3	613.1	0.0	15.8	0.0	4.8	0.5	0.0	X	X	325.00	762.00	1384.76
$L_{17,53}$	49	7	2	0	0	0	ERR	ERR	ERR	ERR	ERR	ERR	ERR	ERR	ERR	X	X	538.00	1100.00	1861.54
$L_{17,54}$	49	113	2	0	0	0	ERR	ERR	ERR	ERR	ERR	ERR	ERR	ERR	ERR	INT	X	786.00	1510.00	2355.65
$L_{17,55}$	49	7	6	0	0	0	ERR	ERR	ERR	ERR	ERR	ERR	ERR	ERR	ERR	X	X	1040.00	1920.00	2849.34

Benchmarks L_{17} (continued on the next page).

$\mathscr{C}$	d	τ	S	E	V	A	Isot	Ib	Ig	Ir	If	Iv	Ia	Is	Ic	Ins	Top	CA0	CA1	C2D
$L_{17,56}$	49	33	6	0	0	0	606.97	0.3	557.5	0.0	8.3	0.0	5.0	1.3	0.0	X	X	INT	INT	917.28
$L_{17,57}$	49	60	6	0	0	0	683.17	0.3	617.2	0.0	15.8	0.0	4.7	1.5	0.0	X	X	INT	INT	1374.69
$L_{17,58}$	49	86	6	0	0	0	ERR	ERR	ERR	ERR	ERR	ERR	ERR	ERR	ERR	X	X	INT	INT	1850.53
$L_{17,59}$	49	113	6	0	0	0	ERR	ERR	ERR	ERR	ERR	ERR	ERR	ERR	ERR	INT	X	INT	INT	2347.76
$L_{17,60}$	49	7	6	0	0	0	ERR	ERR	ERR	ERR	ERR	ERR	ERR	ERR	ERR	X	X	INT	INT	2861.47
$L_{17,61}$	33	16	2	0	0	0	48.73	0.1	43.9	0.0	0.5	0.0	0.7	0.2	0.0	X	9.44	22.50	39.40	81.97
$L_{17,62}$	33	30	2	0	0	0	51.82	0.1	46.4	0.0	0.7	0.0	0.8	0.2	0.0	X	14.96	23.90	46.20	120.22
$L_{17,63}$	33	43	2	0	0	0	52.26	0.1	46.1	0.0	1.0	0.0	0.9	0.4	0.0	X	20.64	26.50	49.90	145.41
$L_{17,64}$	33	56	2	0	0	0	INT	INT	INT	INT	INT	INT	INT	INT	INT	X	26.62	29.80	57.40	182.59
$L_{17,65}$	33	70	2	0	0	0	INT	INT	INT	INT	INT	INT	INT	INT	INT	X	X	X	X	X

Benchmarks L_{17}.

A.1.3 Benchmarks on resultants

The curves $R_{i,j}$ are the resultants in z of two surfaces of degree i, with coefficients of bitsize 32 and 50% of nonzero coefficients. Each experiment is repeated five times, and j reflects the experiment number.

$\mathscr{C}$	d	τ	S	E	V	A	Isot	Ib	Ig	Ir	If	Iv	Ia	Is	Ic	Ins	Top	CA0	CA1	C2D
$R_{4,1}$	16	56	4	6	0	0	30.10	23.8	2.3	0.0	0.0	0.0	2.6	0.4	0.0	INT	57.55	16.00	18.50	PRIM
$R_{4,2}$	16	53	14	12	0	0	32.01	23.1	2.3	0.0	0.3	0.0	2.6	1.1	0.0	X	62.83	17.00	19.00	PRIM
$R_{4,3}$	16	44	4	6	0	0	27.86	21.9	2.3	0.0	0.1	0.0	2.1	0.5	0.0	X	50.53	12.80	14.70	PRIM
$R_{4,4}$	16	47	4	4	0	0	28.02	22.3	2.0	0.0	0.1	0.0	2.3	0.3	0.0	X	49.33	12.70	14.80	PRIM
$R_{4,5}$	16	49	12	6	0	0	29.14	21.7	2.2	0.0	0.2	0.0	2.1	0.8	0.0	INT	52.78	13.80	16.10	PRIM
$R_{5,1}$	24	54	11	12	0	2	542.14	498.5	19.3	0.0	1.8	0.0	14.2	2.3	0.0	X	X	245.00	266.00	PRIM
$R_{5,2}$	25	66	14	12	0	0	698.42	633.6	32.8	0.0	2.3	0.0	18.6	3.3	0.0	X	X	553.00	597.00	PRIM
$R_{5,3}$	23	53	5	5	0	1	255.86	230.0	13.9	0.0	0.0	0.0	8.6	0.8	0.0	X	X	154.00	168.00	PRIM
$R_{5,4}$	25	72	8	10	0	0	736.30	662.8	43.4	0.0	0.7	0.0	22.2	2.1	0.0	X	X	656.00	702.00	PRIM
$R_{5,5}$	25	60	10	12	0	1	727.56	665.2	32.5	0.0	2.0	0.0	19.1	2.3	0.0	X	X	456.00	492.00	PRIM

Benchmarks of resultants.

A.1.4 Benchmarks on translated curves

The curves shown in this section are obtained as a product of one curve and one or more of its vertical translates. The curves $M_{i,j,k}$ are the product of a random curve of degree

i, with 32-bit coefficients and 50% of nonzero coefficients, and j of its vertical translates (note that j may also be equal to zero). Each experiment is repeated five times, and the experiment number is specified by k.

The curves $L_{i,j,k}$ are the product of the curve $L_{i,j}$ shown above and k of its vertical translates.

The curves $R_{i,j,k,l}$ are the product of a resultant in z of two surfaces of degree i and j respectively and k of its vertical translates. Each experiment is repeated five times, and l specifies the experiment number.

$\mathscr{C}$	d	τ	S	E	V	A	ISOT	IB	IG	IR	IF	IV	IA	IS	IC	INS	TOP	CA0	CA1	C2D
$M_{4,0,1}$	4	31	0	7	0	1	0.18	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.13	0.08	0.13	0.13	MULT
$M_{4,0,2}$	4	32	0	4	0	0	0.13	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.10	0.06	0.02	0.02	0.32
$M_{4,0,3}$	4	32	0	5	0	1	0.18	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.17	0.07	0.03	0.03	MULT
$M_{4,0,4}$	4	32	1	2	0	0	0.16	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.16	0.06	0.02	0.02	0.33
$M_{4,0,5}$	4	32	0	0	0	0	0.10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.04	0.00	0.01	0.33
$M_{4,1,1}$	8	64	2	4	0	1	0.86	0.3	0.0	0.0	0.0	0.0	0.3	0.1	0.0	17.18	0.41	0.58	0.67	MULT
$M_{4,1,2}$	8	65	2	4	0	0	0.79	0.3	0.0	0.0	0.0	0.0	0.2	0.1	0.0	14.86	0.61	0.48	0.61	0.43
$M_{4,1,3}$	8	65	0	0	0	0	0.68	0.3	0.1	0.0	0.0	0.0	0.2	0.0	0.0	39.83	0.33	0.11	0.12	0.33
$M_{4,1,4}$	8	64	8	4	0	0	0.86	0.2	0.0	0.0	0.1	0.0	0.0	0.1	0.0	7.46	1.06	0.76	0.90	0.42
$M_{4,1,5}$	8	64	2	4	0	0	1.14	0.5	0.1	0.0	0.0	0.0	0.2	0.1	0.0	44.64	0.92	0.65	0.83	1.03
$M_{4,2,1}$	12	99	6	9	0	1	4.95	1.9	0.5	0.0	0.2	0.0	1.2	0.3	0.0	138.40	X	7.25	8.20	MULT
$M_{4,2,2}$	12	94	5	3	0	0	3.05	1.2	0.2	0.0	0.1	0.0	0.8	0.3	0.0	249.84	X	6.26	8.02	0.94
$M_{4,2,3}$	12	96	12	6	0	1	4.94	1.8	0.2	0.0	0.1	0.0	1.4	0.4	0.0	163.16	10.34	6.63	8.49	0.61
$M_{4,2,4}$	12	96	8	6	0	0	3.58	1.5	0.2	0.0	0.1	0.0	0.9	0.3	0.0	155.86	9.71	5.46	6.58	0.55
$M_{4,2,5}$	12	96	10	6	0	0	3.95	1.2	0.2	0.0	0.3	0.0	0.7	0.5	0.0	234.62	14.60	6.64	7.60	1.16
$M_{4,3,1}$	16	129	6	0	0	0	15.74	9.1	1.4	0.0	0.2	0.0	3.9	0.3	0.0	X	X	88.10	97.10	0.44
$M_{4,3,2}$	16	128	16	8	0	0	21.81	10.2	1.6	0.0	0.5	0.0	5.9	1.1	0.0	X	X	94.60	106.00	1.54
$M_{4,3,3}$	16	130	6	8	0	0	19.00	10.2	1.5	0.0	0.3	0.0	5.1	0.7	0.0	X	X	93.20	106.00	2.19
$M_{4,3,4}$	16	127	12	8	0	1	20.89	10.8	1.0	0.0	0.6	0.0	5.0	1.1	0.0	X	X	67.80	71.60	MULT
$M_{4,3,5}$	16	129	26	16	0	0	26.36	10.1	3.3	0.0	0.9	0.0	5.1	2.4	0.0	X	X	81.50	97.40	1.44
$M_{5,0,1}$	5	32	0	5	0	1	0.20	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	ERR	0.07	0.13	0.14	MULT
$M_{5,0,2}$	5	32	0	4	0	2	0.27	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.26	0.16	0.04	0.04	MULT
$M_{5,0,3}$	5	32	0	2	0	0	0.17	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.16	0.06	0.02	0.03	0.33
$M_{5,0,4}$	5	32	0	2	0	0	0.16	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.30	0.06	0.02	0.02	0.34
$M_{5,0,5}$	5	32	0	4	0	1	0.19	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.25	0.09	0.03	0.04	MULT
$M_{5,1,1}$	10	65	4	4	0	1	2.41	1.2	0.1	0.0	0.0	0.0	0.6	0.1	0.0	X	X	1.82	1.89	MULT
$M_{5,1,2}$	10	65	6	8	0	0	2.52	1.2	0.2	0.0	0.0	0.0	0.5	0.2	0.0	X	X	2.13	2.47	2.88
$M_{5,1,3}$	10	64	6	12	0	0	3.05	1.4	0.2	0.0	0.1	0.0	0.4	0.4	0.0	X	X	2.71	2.82	11.55

Benchmarks random with traslations (continued on the next page).

$\mathscr{C}$	d	τ	S	E	V	A	Isot	IB	IG	IR	IF	IV	IA	IS	IC	Ins	Top	CA0	CA1	C2D
$M_{5,1,4}$	10	65	4	8	0	0	2.94	1.5	0.2	0.0	0.0	0.0	0.5	0.3	0.0	X	X	2.26	2.84	8.43
$M_{5,1,5}$	10	62	2	8	0	0	2.21	1.1	0.2	0.0	0.0	0.0	0.4	0.2	0.0	X	X	1.97	2.25	8.49
$M_{5,2,1}$	15	95	23	6	0	0	9.75	3.1	1.3	0.0	0.6	0.0	0.9	1.4	0.0	X	X	34.20	38.70	1.82
$M_{5,2,2}$	19	95	23	6	0	0	ERR	ERR	ERR	ERR	ERR	ERR	ERR	ERR	ERR	X	X	40.80	48.00	18.56
$M_{5,2,3}$	15	95	20	12	0	1	14.90	7.8	0.6	0.0	0.3	0.0	2.7	1.1	0.0	X	X	32.30	37.00	MULT
$M_{5,2,4}$	15	96	10	18	0	0	13.98	7.3	0.7	0.0	0.3	0.0	2.8	0.9	0.0	X	X	33.60	33.20	4.37
$M_{5,2,5}$	15	97	16	18	0	0	20.12	11.2	0.9	0.0	0.5	0.0	3.2	1.5	0.0	X	X	46.50	53.20	11.34
$M_{5,3,1}$	20	130	26	8	0	1	86.11	56.2	5.2	0.0	2.3	0.0	13.7	2.3	0.0	X	X	444.00	475.00	MULT
$M_{5,3,2}$	20	128	18	12	0	1	54.16	35.8	3.9	0.0	0.7	0.0	7.8	1.8	0.0	X	X	307.00	307.00	MULT
$M_{5,3,3}$	20	132	18	12	0	0	70.88	40.3	14.7	0.0	1.3	0.0	8.0	2.0	0.0	X	X	291.00	318.00	3.53
$M_{5,3,4}$	20	129	16	12	0	0	77.35	53.6	6.1	0.0	1.4	0.0	8.7	2.7	0.0	X	X	432.00	515.00	22.79
$M_{5,3,5}$	20	130	16	12	0	0	ERR	ERR	ERR	ERR	ERR	ERR	ERR	ERR	ERR	X	X	562.00	594.00	12.78
$M_{6,0,1}$	6	32	0	3	0	1	0.30	0.1	0.0	0.0	0.0	0.0	0.1	0.0	0.0	X	X	0.04	0.05	MULT
$M_{6,0,2}$	6	32	0	2	0	1	0.40	0.2	0.0	0.0	0.0	0.0	0.1	0.0	0.0	X	X	0.13	0.14	MULT
$M_{6,0,3}$	6	32	0	10	0	1	0.44	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.0	X	X	0.12	0.13	MULT
$M_{6,0,4}$	6	32	1	3	0	0	0.36	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	X	X	0.06	0.07	0.36
$M_{6,0,5}$	6	32	0	2	0	1	0.29	0.1	0.0	0.0	0.0	0.0	0.1	0.0	0.0	X	X	0.05	0.05	MULT
$M_{6,1,1}$	12	65	6	12	0	0	6.86	4.3	0.5	0.0	0.1	0.0	0.8	0.4	0.0	X	X	6.42	6.17	34.92
$M_{6,1,2}$	12	65	2	4	0	1	7.53	5.2	0.5	0.0	0.2	0.0	1.0	0.1	0.0	X	X	5.95	6.79	MULT
$M_{6,1,3}$	12	64	0	8	0	1	6.16	4.2	0.5	0.0	0.0	0.0	1.0	0.2	0.0	X	X	5.46	4.68	MULT
$M_{6,1,4}$	12	66	3	6	0	1	4.27	2.7	0.3	0.0	0.0	0.0	0.7	0.2	0.0	X	X	4.05	4.50	MULT
$M_{6,1,5}$	12	65	3	6	0	1	5.61	3.9	0.3	0.0	0.0	0.0	0.9	0.2	0.0	X	X	4.69	5.67	6.25
$M_{6,2,1}$	18	97	18	18	0	0	27.97	17.0	1.5	0.0	0.9	0.0	3.3	1.8	0.0	X	X	87.50	98.60	21.59
$M_{6,2,2}$	18	100	6	12	0	0	65.75	49.7	4.8	0.0	0.8	0.0	7.1	1.1	0.0	X	X	161.00	186.00	158.76
$M_{6,2,3}$	18	98	16	18	0	1	62.36	45.1	4.0	0.0	1.1	0.0	6.4	1.8	0.0	X	X	150.00	166.00	68.86
$M_{6,2,4}$	18	96	9	15	0	1	51.05	39.6	2.1	0.0	1.0	0.0	4.7	1.2	0.0	X	X	123.00	88.30	MULT
$M_{6,2,5}$	18	99	13	9	0	1	40.42	31.0	1.9	0.0	0.4	0.0	4.0	0.9	0.0	X	X	75.60	90.40	10.73
$M_{6,3,1}$	24	128	24	12	0	1	ERR	ERR	ERR	ERR	ERR	ERR	ERR	ERR	ERR	X	X	1560.00	1610.00	63.34
$M_{6,3,2}$	24	128	24	12	0	1	276.69	219.8	20.7	0.0	2.7	0.0	20.9	3.6	0.0	X	X	1810.00	1780.00	MULT
$M_{6,3,3}$	24	130	30	32	0	1	224.57	160.2	17.0	0.0	4.3	0.0	23.3	6.6	0.0	X	X	1460.00	1780.00	69.07
$M_{6,3,4}$	24	129	20	24	0	0	189.43	130.6	18.5	0.0	6.1	0.0	19.4	4.6	0.0	X	X	1350.00	1320.00	53.84
$M_{6,3,5}$	24	130	24	8	0	1	139.14	102.8	9.1	0.0	1.9	0.0	16.2	2.2	0.0	X	X	1230.00	1330.00	MULT
$M_{7,0,1}$	7	32	0	8	0	0	0.40	0.1	0.1	0.0	0.0	0.0	0.0	0.1	0.0	X	X	0.12	0.14	0.39
$M_{7,0,2}$	7	32	0	1	0	1	0.86	0.2	0.5	0.0	0.0	0.0	0.1	0.0	0.0	X	X	0.23	0.22	MULT
$M_{7,0,3}$	7	32	0	8	0	1	0.82	0.3	0.1	0.0	0.0	0.0	0.1	0.1	0.0	X	X	0.14	0.16	MULT
$M_{7,0,4}$	7	32	0	5	0	0	0.44	0.1	0.1	0.0	0.0	0.0	0.1	0.1	0.0	X	X	0.11	0.11	0.39
$M_{7,0,5}$	7	32	0	7	0	0	0.65	0.3	0.1	0.0	0.0	0.0	0.1	0.1	0.0	X	X	0.12	0.12	0.38
$M_{7,1,1}$	14	65	6	8	0	0	14.76	10.6	1.1	0.0	0.1	0.0	1.6	0.4	0.0	X	X	16.60	18.20	234.45
$M_{7,1,2}$	14	64	2	4	0	0	16.13	12.3	1.3	0.0	0.1	0.0	1.7	0.2	0.0	X	X	18.20	21.70	530.40

Benchmarks random with traslations (continued on the next page).

$\mathscr{C}$	d	τ	S	E	V	A	ISOT	IB	IG	IR	IF	IV	IA	IS	IC	INS	TOP	CA0	CA1	C2D
$M_{7,1,3}$	14	64	4	8	0	1	17.50	13.1	1.1	0.0	0.2	0.0	1.8	0.4	0.0	X	X	14.90	20.40	MULT
$M_{7,1,4}$	14	63	8	6	0	1	16.33	11.8	0.9	0.0	0.1	0.0	2.0	0.4	0.0	X	X	17.80	19.90	MULT
$M_{7,1,5}$	14	64	4	8	0	2	14.11	9.7	0.9	0.0	0.6	0.0	1.6	0.3	0.0	X	X	15.10	16.50	MULT
$M_{7,2,1}$	21	97	14	12	0	0	153.34	120.0	11.6	0.0	1.7	0.0	13.0	2.2	0.0	X	X	601.00	543.00	610.35
$M_{7,2,2}$	21	97	18	18	0	0	137.65	105.3	10.0	0.0	2.2	0.0	11.3	2.5	0.0	X	X	434.00	428.00	356.27
$M_{7,2,3}$	21	98	8	12	0	1	154.27	124.4	10.3	0.0	1.5	0.0	12.7	1.4	0.0	X	X	559.00	599.00	411.75
$M_{7,2,4}$	21	96	14	9	0	0	120.64	97.0	8.0	0.0	1.0	0.0	8.5	2.0	0.0	X	X	531.00	501.00	647.14
$M_{7,2,5}$	21	97	12	18	0	1	149.70	117.8	10.2	0.0	2.2	0.0	12.7	2.0	0.0	X	X	500.00	550.00	MULT
$M_{7,3,1}$	28	127	30	16	0	0	826.84	687.8	62.6	0.0	6.7	0.0	43.4	7.2	0.0	X	X	INT	INT	452.55
$M_{7,3,2}$	28	131	24	20	0	1	732.11	582.9	62.8	0.0	11.5	0.0	46.1	7.5	0.0	X	X	X	X	X

Benchmarks random with traslations.

$\mathscr{C}$	d	τ	S	E	V	A	ISOT	IB	IG	IR	IF	IV	IA	IS	IC	INS	TOP	CA0	CA1	C2D
$L_{1,1,1}$	8	7	7	2	0	0	0.47	0.1	0.0	0.0	0.0	0.0	0.1	0.1	0.0	X	X	0.33	0.41	0.33
$L_{1,2,1}$	10	8	12	2	0	0	1.14	0.1	0.1	0.0	0.0	0.0	0.1	0.3	0.0	X	X	0.86	1.36	0.35
$L_{1,3,1}$	12	10	21	2	0	0	2.54	0.4	0.1	0.0	0.0	0.0	0.3	0.6	0.0	X	X	3.88	3.93	0.40
$L_{1,4,1}$	14	12	30	2	0	0	6.34	0.4	0.8	0.0	0.4	0.0	0.8	1.2	0.0	X	X	9.01	9.27	0.55
$L_{1,5,1}$	16	14	43	2	0	0	13.33	0.6	2.6	0.0	1.1	0.0	1.6	2.3	0.0	X	X	27.00	33.70	0.80
$L_{1,6,1}$	18	16	56	2	0	0	28.88	1.2	8.0	0.0	3.6	0.0	3.3	3.9	0.0	X	X	76.10	79.80	1.41
$L_{1,7,1}$	20	18	73	2	0	0	ERR	ERR	ERR	ERR	ERR	ERR	ERR	ERR	ERR	X	X	290.00	303.00	2.41

Benchmarks L_1 with one traslation.

$\mathscr{C}$	d	τ	S	E	V	A	ISOT	IB	IG	IR	IF	IV	IA	IS	IC	INS	TOP	CA0	CA1	C2D
$L_{3,1,1}$	8	10	1	16	0	0	0.74	0.1	0.0	0.0	0.0	0.0	0.1	0.2	0.0	4.42	1.46	0.92	0.81	0.35
$L_{3,2,1}$	8	21	1	18	0	0	0.93	0.2	0.0	0.0	0.0	0.0	0.1	0.3	0.0	6.47	2.48	0.88	0.72	0.39
$L_{3,3,1}$	8	41	1	18	0	0	1.05	0.2	0.0	0.0	0.0	0.0	0.2	0.3	0.0	10.76	4.07	1.24	0.80	0.49
$L_{3,4,1}$	8	67	1	18	0	0	1.18	0.2	0.0	0.0	0.0	0.0	0.2	0.3	0.0	16.64	6.21	0.76	0.86	0.61
$L_{3,5,1}$	8	101	1	18	0	0	1.38	0.3	0.1	0.0	0.0	0.0	0.3	0.3	0.0	24.94	9.62	1.22	1.44	0.91
$L_{3,6,1}$	10	8	6	27	0	0	2.47	0.4	0.1	0.0	0.1	0.0	0.2	0.7	0.0	32.08	6.19	2.42	3.21	0.50
$L_{3,7,1}$	10	22	6	30	0	0	2.86	0.4	0.1	0.0	0.3	0.0	0.2	0.8	0.0	74.08	10.46	2.78	2.90	0.90
$L_{3,8,1}$	10	42	6	32	0	0	3.46	0.5	0.1	0.0	0.3	0.0	0.4	0.9	0.0	166.52	22.38	3.30	3.66	2.16
$L_{3,9,1}$	10	68	6	32	0	0	4.03	0.8	0.1	0.0	0.4	0.0	0.5	0.9	0.0	355.29	37.63	3.79	4.06	5.64
$L_{3,10,1}$	10	101	6	32	0	0	4.47	1.0	0.1	0.0	0.4	0.0	0.7	0.9	0.0	X	59.02	4.26	4.97	11.69
$L_{3,11,1}$	12	12	9	44	0	0	6.04	1.0	0.3	0.0	0.4	0.0	0.4	1.6	0.0	X	33.44	8.64	9.29	1.88
$L_{3,12,1}$	12	25	9	46	0	0	7.10	1.2	0.4	0.0	0.5	0.0	0.6	1.8	0.0	X	X	X	X	X
$L_{3,13,1}$	12	45	9	48	0	0	9.05	2.0	0.7	0.0	0.6	0.0	0.9	1.9	0.0	X	108.43	10.80	12.20	22.40

Benchmarks L_3 with one traslation (continued on the next page).

$\mathscr{C}$	d	τ	S	E	V	A	Isot	Ib	Ig	Ir	If	Iv	Ia	Is	Ic	Ins	Top	CA0	CA1	C2d
$L_{3,14,1}$	12	72	9	50	0	0	10.47	2.4	0.4	0.0	1.3	0.0	1.3	2.0	0.0	X	193.85	13.40	14.30	70.37
$L_{3,15,1}$	12	105	9	50	0	0	12.61	4.5	0.6	0.0	0.8	0.0	1.6	2.0	0.0	X	297.87	15.80	17.20	568.69
$L_{3,16,1}$	14	15	16	64	0	0	13.60	2.4	0.7	0.0	1.4	0.0	1.0	3.3	0.0	X	X	X	X	X
$L_{3,17,1}$	14	28	16	66	0	0	17.09	4.3	1.4	0.0	1.6	0.0	1.3	3.5	0.0	X	X	23.40	24.00	261.16
$L_{3,18,1}$	14	48	16	68	0	0	20.73	5.1	2.6	0.0	2.0	0.0	1.9	3.6	0.0	X	X	27.00	30.70	877.19
$L_{3,19,1}$	14	75	16	70	0	0	29.77	9.6	5.0	0.0	2.6	0.0	2.7	3.9	0.0	X	X	34.00	35.70	ERR
$L_{3,20,1}$	14	108	16	72	0	0	30.23	11.3	1.6	0.0	3.2	0.0	3.7	4.1	0.0	X	X	45.50	47.20	PRIM
$L_{3,21,1}$	16	14	25	88	0	0	33.95	7.4	3.7	0.0	3.4	0.0	2.0	6.6	0.0	X	X	39.90	69.60	ERR
$L_{3,22,1}$	16	23	25	90	0	0	45.67	14.1	6.0	0.0	4.7	0.0	2.5	6.6	0.0	X	X	55.10	66.40	ERR
$L_{3,23,1}$	16	43	25	92	0	0	55.88	16.1	11.2	0.0	5.5	0.0	3.5	6.8	0.0	X	X	67.10	78.40	ERR
$L_{3,24,1}$	16	70	25	94	0	0	82.29	29.9	20.0	0.0	7.1	0.0	4.9	7.1	0.1	X	X	84.50	103.00	PRIM
$L_{3,25,1}$	16	103	25	96	0	0	111.25	39.0	34.7	0.0	8.6	0.0	6.7	7.4	0.0	X	X	116.00	95.90	PRIM
$L_{3,26,1}$	18	-	0	116	0	0	93.35	28.4	14.1	0.0	12.4	0.0	4.1	12.5	0.1	X	X	109.00	126.00	ERR
$L_{3,27,1}$	18	-	0	118	0	0	109.60	31.1	21.8	0.0	14.8	0.0	5.2	12.8	0.0	X	X	138.00	145.00	PRIM
$L_{3,28,1}$	18	-	0	120	0	0	161.31	61.4	36.9	0.0	17.5	0.0	6.9	13.2	0.0	X	X	168.00	179.00	PRIM
$L_{3,29,1}$	18	-	0	122	0	0	207.54	71.3	65.6	0.0	21.8	0.0	9.2	13.8	0.0	X	X	223.00	241.00	PRIM
$L_{3,30,1}$	18	-	0	124	0	0	304.00	83.0	107.3	0.0	54.3	0.0	12.4	15.0	0.0	X	X	294.00	335.00	PRIM
$L_{3,31,1}$	18	-	0	130	0	0	203.04	64.9	65.4	0.0	20.3	0.0	5.4	18.2	0.1	X	X	177.00	171.00	C2d

Benchmarks L_3 with one traslation.

$\mathscr{C}$	d	τ	S	E	V	A	Isot	Ib	Ig	Ir	If	Iv	Ia	Is	Ic	Ins	Top	CA0	CA1	C2d
$L_{4,1,1}$	8	23	0	8	0	0	0.38	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.0	X	X	0.41	0.43	0.34
$L_{4,2,1}$	8	43	0	8	0	0	0.46	0.1	0.0	0.0	0.0	0.0	0.1	0.1	0.0	X	X	0.43	0.51	0.34
$L_{4,3,1}$	8	23	0	8	0	0	0.40	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.0	X	X	0.39	0.44	0.34
$L_{4,4,1}$	8	43	0	8	0	0	0.43	0.0	0.0	0.0	0.0	0.0	0.1	0.1	0.0	X	X	0.42	0.48	0.35
$L_{4,5,1}$	8	30	0	8	0	0	0.43	0.1	0.0	0.0	0.0	0.0	0.1	0.1	0.0	X	X	0.40	0.47	0.37
$L_{4,6,1}$	8	43	0	8	0	0	0.48	0.1	0.0	0.0	0.0	0.0	0.1	0.1	0.0	X	X	0.44	0.50	0.34
$L_{4,7,1}$	12	45	0	8	0	0	1.03	0.2	0.0	0.0	0.0	0.0	0.3	0.2	0.0	X	X	1.45	2.03	0.53
$L_{4,8,1}$	12	85	0	8	0	0	1.46	0.4	0.0	0.0	0.0	0.0	0.5	0.2	0.0	X	X	2.83	2.94	0.80
$L_{4,9,1}$	12	45	0	8	0	0	1.02	0.2	0.0	0.0	0.0	0.0	0.3	0.2	0.0	X	X	1.63	2.22	0.49
$L_{4,10,1}$	12	85	0	8	0	0	1.46	0.4	0.0	0.0	0.0	0.0	0.5	0.2	0.0	X	X	2.75	2.98	0.79
$L_{4,11,1}$	12	45	0	12	0	0	1.24	0.2	0.0	0.0	0.0	0.0	0.3	0.3	0.0	X	X	2.08	2.56	0.49
$L_{4,12,1}$	12	85	0	8	0	0	1.44	0.4	0.1	0.0	0.0	0.0	0.5	0.2	0.0	X	X	2.54	3.36	0.76
$L_{4,13,1}$	16	73	0	4	0	0	3.09	1.4	0.2	0.0	0.0	0.0	1.1	0.2	0.0	X	X	10.30	11.80	3.98
$L_{4,14,1}$	16	140	0	8	0	0	6.03	2.9	0.2	0.0	0.0	0.0	2.2	0.3	0.0	X	X	26.20	26.30	13.15
$L_{4,15,1}$	16	73	0	8	0	0	3.36	1.4	0.1	0.0	0.0	0.0	1.1	0.4	0.0	X	X	10.80	12.30	3.40
$L_{4,16,1}$	16	140	0	8	0	0	4.83	1.7	0.2	0.0	0.0	0.0	2.2	0.3	0.0	X	X	26.10	25.70	12.21
$L_{4,17,1}$	16	73	0	8	0	0	2.81	0.8	0.1	0.0	0.0	0.0	1.1	0.3	0.0	X	X	12.30	12.50	3.12

Benchmarks L_4 with one traslation (continued on the next page).

$\mathscr{C}$	d	τ	s	e	v	a	Isot	Ib	Ig	Ir	If	Iv	Ia	Is	Ic	Ins	Top	CA0	CA1	C2d
$L_{4,18,1}$	16	140	0	8	0	0	4.86	1.7	0.2	0.0	0.0	0.0	2.2	0.4	0.0	X	X	25.40	26.40	11.41
$L_{4,19,1}$	20	108	0	4	0	0	11.42	6.8	0.4	0.0	0.0	0.0	3.6	0.3	0.0	X	X	73.60	68.80	43.54
$L_{4,20,1}$	20	208	0	8	0	0	23.87	14.7	0.7	0.0	0.0	0.0	7.3	0.4	0.0	X	X	189.00	172.00	196.87
$L_{4,21,1}$	20	108	0	8	0	0	11.76	6.8	0.4	0.0	0.0	0.0	3.5	0.5	0.0	X	X	83.50	72.80	38.38
$L_{4,22,1}$	20	208	0	8	0	0	23.68	14.6	0.7	0.0	0.0	0.0	7.2	0.5	0.0	X	X	197.00	160.00	184.90
$L_{4,23,1}$	20	108	0	8	0	0	11.62	6.7	0.4	0.0	0.0	0.0	3.5	0.4	0.0	X	X	79.80	70.70	34.88
$L_{4,24,1}$	20	208	0	8	0	0	23.58	14.5	0.7	0.0	0.0	0.0	7.2	0.5	0.0	X	X	197.00	171.00	173.92
$L_{4,25,1}$	24	150	0	4	0	0	41.03	28.9	1.4	0.0	0.0	0.0	9.8	0.4	0.0	X	X	459.00	310.00	356.39
$L_{4,26,1}$	24	289	0	8	0	0	91.27	65.6	2.7	0.0	0.0	0.0	21.1	0.8	0.0	X	X	1060.00	871.00	PRIM
$L_{4,27,1}$	24	150	0	8	0	0	41.41	28.6	1.4	0.0	0.0	0.0	9.7	0.9	0.0	X	X	433.00	336.00	320.12
$L_{4,28,1}$	24	289	0	8	0	0	90.70	65.2	2.4	0.0	0.0	0.0	21.0	0.8	0.0	X	X	1080.00	868.00	PRIM
$L_{4,29,1}$	24	150	0	8	0	0	41.26	28.5	1.4	0.0	0.0	0.0	9.6	0.9	0.0	X	X	417.00	340.00	293.25
$L_{4,30,1}$	24	289	0	8	0	0	89.72	64.7	2.4	0.0	0.0	0.0	20.7	0.8	0.0	X	X	1140.00	870.00	PRIM

Benchmarks L_4 with one traslation.

$\mathscr{C}$	d	τ	s	e	v	a	Isot	Ib	Ig	Ir	If	Iv	Ia	Is	Ic	Ins	Top	CA0	CA1	C2d
$L_{7,1,1}$	8	2	2	4	0	0	0.35	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.0	X	X	0.29	0.38	0.32
$L_{7,2,1}$	12	2	2	4	0	0	0.67	0.1	0.1	0.0	0.0	0.0	0.0	0.2	0.0	X	X	1.37	1.84	0.33
$L_{7,3,1}$	16	2	2	4	0	0	1.83	0.2	0.7	0.0	0.0	0.0	0.0	0.3	0.0	X	X	7.42	9.40	0.33
$L_{7,4,1}$	20	2	2	4	0	0	3.93	0.2	2.1	0.0	0.0	0.0	0.1	0.7	0.0	X	X	31.20	29.30	0.35
$L_{7,5,1}$	24	2	2	4	0	0	9.79	0.3	6.6	0.0	0.0	0.0	0.2	1.1	0.0	X	X	174.00	216.00	0.37
$L_{7,6,1}$	28	2	2	4	0	0	25.92	0.4	20.6	0.0	0.0	0.0	0.4	2.1	0.0	X	X	719.00	740.00	0.46
$L_{7,7,1}$	32	2	2	4	0	0	52.73	0.7	43.1	0.0	0.0	0.0	0.6	4.1	0.0	X	X	1360.00	1210.00	0.54
$L_{7,8,1}$	36	2	2	4	0	0	115.31	0.5	98.2	0.0	0.0	0.0	1.4	6.6	0.0	X	X	INT	INT	0.77
$L_{7,9,1}$	40	2	2	4	0	0	253.46	0.6	223.3	0.0	0.0	0.0	2.7	11.4	0.0	X	X	X	X	1.07
$L_{7,10,1}$	44	2	2	4	0	0	431.71	1.0	386.3	0.0	3.3	0.0	4.6	18.5	0.0	X	X	X	X	1.78
$L_{7,11,1}$	48	2	2	4	0	0	812.63	1.6	744.3	0.0	0.0	0.0	6.4	31.5	0.0	X	X	X	X	2.26

Benchmarks L_7 with one traslation.

$\mathscr{C}$	d	τ	s	e	v	a	Isot	Ib	Ig	Ir	If	Iv	Ia	Is	Ic	Ins	Top	CA0	CA1	C2d
$L_{8,1,1}$	4	0	0	0	1	0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	X	X	0.00	0.00	ERR
$L_{8,2,1}$	6	0	0	0	1	0	0.08	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	X	X	0.00	0.00	0.32
$L_{8,3,1}$	3	0	0	0	1	0	0.08	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	X	X	0.00	0.00	0.36
$L_{8,4,1}$	3	0	0	0	1	0	0.09	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	X	X	0.00	0.00	0.32
$L_{8,5,1}$	3	0	0	0	1	0	0.08	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	X	X	0.00	0.00	0.32
$L_{8,6,1}$	3	0	0	0	1	0	0.08	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	X	X	0.00	0.00	0.32

Benchmarks L_8 with one traslation (continued on the next page).

$\mathscr{C}$	d	τ	S	E	V	A	ISOT	IB	IG	IR	IF	IV	IA	IS	IC	INS	TOP	CA0	CA1	C2D
$L_{8,7,1}$	3	0	0	0	1	0	0.08	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	X	X	0.00	0.00	0.32
$L_{8,8,1}$	3	0	0	0	1	0	0.08	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	X	X	0.00	0.00	0.32
$L_{8,9,1}$	3	0	0	0	1	0	0.08	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	X	X	0.00	0.00	0.33
$L_{8,10,1}$	3	0	0	0	1	0	0.08	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	X	X	0.00	0.00	0.32
$L_{8,11,1}$	3	0	0	0	1	0	0.08	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	X	X	0.00	0.00	0.32
$L_{8,12,1}$	3	0	0	0	1	0	0.08	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	X	X	0.00	0.00	0.32
$L_{8,13,1}$	3	0	0	0	1	0	0.08	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	X	X	0.00	0.00	0.32
$L_{8,14,1}$	9	2	2	4	1	0	0.44	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.0	X	X	0.15	0.41	0.33
$L_{8,15,1}$	13	2	2	4	1	0	0.85	0.2	0.1	0.0	0.0	0.1	0.0	0.3	0.0	X	X	1.46	1.60	0.34
$L_{8,16,1}$	17	2	2	4	1	0	2.53	0.2	0.7	0.0	0.0	0.1	0.0	1.0	0.0	X	X	7.16	7.68	0.34
$L_{8,17,1}$	21	2	2	4	1	0	5.75	0.2	2.1	0.0	0.0	0.1	0.1	2.4	0.0	X	X	30.30	28.60	0.35
$L_{8,18,1}$	25	2	2	4	1	0	14.81	0.3	6.6	0.0	0.0	0.1	0.1	6.2	0.0	X	X	136.00	138.00	0.39
$L_{8,19,1}$	29	2	2	4	1	0	38.10	0.3	20.8	0.0	0.0	0.2	0.1	14.2	0.0	X	X	584.00	590.00	0.47
$L_{8,20,1}$	33	2	2	4	1	0	79.33	0.8	43.5	0.0	0.0	0.2	0.1	30.6	0.0	X	X	ERR	ERR	X
$L_{8,21,1}$	9	2	2	4	1	0	0.45	0.2	0.0	0.0	0.0	0.1	0.0	0.1	0.0	X	X	0.30	0.36	0.32
$L_{8,22,1}$	13	2	2	4	1	0	0.88	0.2	0.1	0.0	0.0	0.1	0.1	0.3	0.0	X	X	1.46	1.83	0.34
$L_{8,23,1}$	17	2	2	4	1	0	2.53	0.2	0.7	0.0	0.0	0.1	0.1	0.9	0.0	X	X	7.17	9.74	0.34
$L_{8,24,1}$	21	2	2	4	1	0	5.86	0.3	2.1	0.0	0.0	0.1	0.1	2.4	0.0	X	X	21.70	41.70	0.36
$L_{8,25,1}$	25	2	2	4	1	0	15.01	0.3	6.6	0.0	0.0	0.1	0.2	6.2	0.0	X	X	86.30	162.00	0.41
$L_{8,26,1}$	29	2	2	4	1	0	37.92	0.4	20.6	0.0	0.0	0.1	0.2	14.2	0.0	X	X	784.00	735.00	0.48
$L_{8,27,1}$	33	2	2	4	1	0	79.17	0.8	43.5	0.0	0.0	0.2	0.3	30.3	0.0	X	X	ERR	ERR	0.57
$L_{8,28,1}$	9	2	2	4	1	0	0.48	0.1	0.0	0.0	0.0	0.0	0.1	0.1	0.0	X	X	0.29	0.37	0.33
$L_{8,29,1}$	13	2	2	4	1	0	1.01	0.2	0.1	0.0	0.0	0.1	0.1	0.3	0.0	X	X	1.76	1.86	0.34
$L_{8,30,1}$	17	2	2	4	1	0	2.68	0.3	0.7	0.0	0.0	0.1	0.2	0.9	0.0	X	X	7.18	8.31	0.35
$L_{8,31,1}$	21	2	2	4	1	0	6.11	0.3	2.1	0.0	0.0	0.1	0.3	2.4	0.0	X	X	30.80	46.00	0.38
$L_{8,32,1}$	25	2	2	4	1	0	15.09	0.3	6.6	0.0	0.0	0.1	0.5	6.1	0.0	X	X	135.00	183.00	0.42
$L_{8,33,1}$	29	2	2	4	1	0	38.64	0.4	20.7	0.0	0.0	0.2	0.6	14.5	0.0	X	X	791.00	595.00	0.54
$L_{8,34,1}$	33	2	2	4	1	0	79.31	0.8	43.1	0.0	0.0	0.2	0.8	30.4	0.0	X	X	ERR	ERR	0.65
$L_{8,35,1}$	9	2	2	4	1	0	0.45	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.0	X	X	0.46	0.38	0.32
$L_{8,36,1}$	13	2	2	4	1	0	0.86	0.2	0.1	0.0	0.0	0.1	0.1	0.2	0.0	X	X	1.34	1.59	0.34
$L_{8,37,1}$	17	2	2	4	1	0	2.52	0.2	0.7	0.0	0.0	0.1	0.0	1.0	0.0	X	X	6.78	7.92	0.34
$L_{8,38,1}$	21	2	2	4	1	0	5.82	0.2	2.1	0.0	0.0	0.1	0.0	2.4	0.0	X	X	33.80	42.00	0.35
$L_{8,39,1}$	25	2	2	4	1	0	14.84	0.3	6.6	0.0	0.0	0.1	0.1	6.2	0.0	X	X	198.00	182.00	0.58
$L_{8,40,1}$	29	2	2	4	1	0	37.88	0.3	20.7	0.0	0.0	0.2	0.1	14.1	0.0	X	X	784.00	670.00	0.46
$L_{8,41,1}$	33	2	2	4	1	0	78.54	0.8	43.2	0.0	0.0	0.2	0.1	30.2	0.0	X	X	ERR	ERR	0.56
$L_{8,42,1}$	9	2	2	4	1	0	0.46	0.2	0.0	0.0	0.0	0.1	0.0	0.1	0.0	X	X	0.30	0.40	0.33
$L_{8,43,1}$	13	2	2	4	1	0	0.92	0.2	0.1	0.0	0.0	0.1	0.1	0.3	0.0	X	X	1.43	1.83	0.33
$L_{8,44,1}$	17	2	2	4	1	0	2.55	0.2	0.7	0.0	0.0	0.1	0.1	0.9	0.0	X	X	4.92	7.42	0.34
$L_{8,45,1}$	21	2	2	4	1	0	5.89	0.3	2.1	0.0	0.0	0.1	0.1	2.4	0.0	X	X	43.00	38.10	0.37

Benchmarks L_8 with one traslation (continued on the next page).

$\mathscr{C}$	d	τ	s	e	v	a	Isot	Ib	Ig	Ir	If	Iv	Ia	Is	Ic	Ins	Top	CA0	CA1	C2d
$L_{8,46,1}$	25	2	2	4	1	0	15.06	0.3	6.6	0.0	0.0	0.1	0.2	6.2	0.0	X	X	116.00	182.00	0.40
$L_{8,47,1}$	29	2	2	4	1	0	37.84	0.4	20.6	0.0	0.0	0.2	0.3	14.0	0.0	X	X	300.00	373.00	0.51
$L_{8,48,1}$	33	2	2	4	1	0	80.37	0.8	44.3	0.0	0.0	0.2	0.3	30.6	0.0	X	X	ERR	1150.00	0.60
$L_{8,49,1}$	9	2	2	4	1	0	0.48	0.1	0.0	0.0	0.0	0.0	0.1	0.1	0.0	X	X	0.47	0.38	0.33
$L_{8,50,1}$	13	2	2	4	1	0	0.98	0.2	0.1	0.0	0.0	0.1	0.2	0.3	0.0	X	X	1.46	1.61	0.35
$L_{8,51,1}$	17	2	2	4	1	0	2.66	0.2	0.7	0.0	0.0	0.1	0.3	0.9	0.0	X	X	6.76	8.90	0.35
$L_{8,52,1}$	21	2	2	4	1	0	6.04	0.2	2.1	0.0	0.0	0.1	0.4	2.4	0.0	X	X	44.50	42.40	0.40
$L_{8,53,1}$	25	2	2	4	1	0	15.16	0.3	6.6	0.0	0.0	0.1	0.5	6.1	0.0	X	X	135.00	138.00	0.41
$L_{8,54,1}$	29	2	2	4	1	0	38.14	0.3	20.6	0.0	0.0	0.2	0.6	14.1	0.0	X	X	617.00	672.00	0.54
$L_{8,55,1}$	33	2	2	4	1	0	80.14	0.8	43.1	0.0	0.0	0.2	0.8	31.3	0.0	X	X	ERR	ERR	0.62
$L_{8,56,1}$	9	2	2	4	1	0	0.46	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.0	X	X	0.34	0.36	0.32
$L_{8,57,1}$	13	2	2	4	1	0	0.84	0.2	0.1	0.0	0.0	0.1	0.0	0.2	0.0	X	X	1.20	1.82	0.33
$L_{8,58,1}$	17	2	2	4	1	0	2.49	0.2	0.7	0.0	0.0	0.1	0.0	0.9	0.0	X	X	7.27	9.21	0.34
$L_{8,59,1}$	21	2	2	4	1	0	5.86	0.2	2.1	0.0	0.0	0.1	0.0	2.5	0.0	X	X	30.70	28.80	0.35
$L_{8,60,1}$	25	2	2	4	1	0	14.85	0.3	6.6	0.0	0.0	0.1	0.1	6.2	0.0	X	X	135.00	111.00	0.38
$L_{8,61,1}$	29	2	2	4	1	0	37.77	0.3	20.7	0.0	0.0	0.2	0.1	14.0	0.0	X	X	776.00	812.00	0.47
$L_{8,62,1}$	33	2	2	4	1	0	78.96	0.8	43.4	0.0	0.0	0.2	0.1	30.4	0.0	X	X	1350.00	ERR	0.55
$L_{8,63,1}$	9	2	2	4	1	0	0.47	0.2	0.0	0.0	0.0	0.1	0.0	0.1	0.0	X	X	0.33	0.35	0.33
$L_{8,64,1}$	13	2	2	4	1	0	0.90	0.2	0.1	0.0	0.0	0.1	0.1	0.3	0.0	X	X	1.36	1.76	0.33
$L_{8,65,1}$	17	2	2	4	1	0	2.52	0.2	0.7	0.0	0.0	0.1	0.1	0.9	0.0	X	X	7.55	8.82	0.34
$L_{8,66,1}$	21	2	2	4	1	0	5.85	0.3	2.1	0.0	0.0	0.1	0.1	2.4	0.0	X	X	33.90	52.30	0.36
$L_{8,67,1}$	25	2	2	4	1	0	15.09	0.3	6.6	0.0	0.0	0.1	0.2	6.2	0.0	X	X	156.00	228.00	0.40
$L_{8,68,1}$	29	2	2	4	1	0	38.11	0.4	20.7	0.0	0.0	0.2	0.3	14.2	0.0	X	X	591.00	914.00	0.48
$L_{8,69,1}$	33	2	2	4	1	0	79.39	0.8	43.4	0.0	0.0	0.2	0.3	30.4	0.0	X	X	ERR	ERR	0.58
$L_{8,70,1}$	9	2	2	4	1	0	0.48	0.1	0.0	0.0	0.0	0.0	0.1	0.1	0.0	X	X	0.16	0.21	0.33
$L_{8,71,1}$	13	2	2	4	1	0	1.02	0.2	0.1	0.0	0.0	0.1	0.1	0.3	0.0	X	X	1.20	1.95	0.35
$L_{8,72,1}$	17	2	2	4	1	0	2.72	0.2	0.7	0.0	0.0	0.1	0.3	1.0	0.0	X	X	6.17	9.78	0.35
$L_{8,73,1}$	21	2	2	4	1	0	6.12	0.3	2.1	0.0	0.0	0.1	0.4	2.5	0.0	X	X	30.80	38.40	0.39
$L_{8,74,1}$	25	2	2	4	1	0	15.32	0.4	6.6	0.0	0.0	0.1	0.5	6.2	0.0	X	X	111.00	244.00	0.44
$L_{8,75,1}$	29	2	2	4	1	0	38.24	0.4	20.6	0.0	0.0	0.2	0.6	14.2	0.0	X	X	731.00	377.00	0.52
$L_{8,76,1}$	33	2	2	4	1	0	79.52	0.7	43.0	0.0	0.0	0.2	0.8	30.7	0.0	X	X	ERR	ERR	0.64
$L_{8,77,1}$	10	2	4	4	0	0	0.66	0.2	0.0	0.0	0.0	0.0	0.0	0.1	0.0	X	X	0.70	0.80	0.34
$L_{8,78,1}$	14	2	4	4	0	0	1.22	0.2	0.3	0.0	0.0	0.0	0.0	0.3	0.0	X	X	2.31	3.13	0.35
$L_{8,79,1}$	18	2	4	4	0	0	2.71	0.3	1.2	0.0	0.0	0.0	0.0	0.4	0.0	X	X	9.66	13.20	0.36
$L_{8,80,1}$	22	2	4	4	0	0	6.05	0.4	3.3	0.0	0.3	0.0	0.0	0.8	0.0	X	X	88.40	91.00	0.37
$L_{8,81,1}$	26	2	4	4	0	0	13.81	0.5	9.7	0.0	0.0	0.0	0.0	1.3	0.0	X	X	347.00	400.00	0.41
$L_{8,82,1}$	10	2	4	4	0	0	0.64	0.2	0.1	0.0	0.0	0.0	0.0	0.1	0.0	X	X	0.70	0.90	0.34
$L_{8,83,1}$	14	2	4	4	0	0	1.26	0.2	0.2	0.0	0.0	0.0	0.1	0.3	0.0	X	X	3.15	3.76	0.34
$L_{8,84,1}$	18	2	4	4	0	0	2.82	0.3	1.2	0.0	0.0	0.0	0.1	0.4	0.0	X	X	9.86	20.60	0.37

Benchmarks L_8 with one traslation (continued on the next page).

$\mathscr{C}$	d	τ	S	E	V	A	Isot	Ib	Ig	Ir	If	Iv	Ia	Is	Ic	Ins	Top	CA0	CA1	C2d
$L_{8,85,1}$	22	2	4	4	0	0	6.26	0.5	3.4	0.0	0.3	0.0	0.1	0.8	0.0	X	X	52.10	98.00	0.36
$L_{8,86,1}$	26	2	4	4	0	0	13.92	0.5	9.7	0.0	0.0	0.0	0.2	1.3	0.0	X	X	157.00	370.00	0.41
$L_{8,87,1}$	10	2	4	4	0	0	0.66	0.2	0.0	0.0	0.0	0.0	0.1	0.1	0.0	X	X	0.39	0.84	0.33
$L_{8,88,1}$	14	2	4	4	0	0	1.36	0.2	0.2	0.0	0.0	0.0	0.1	0.3	0.0	X	X	2.99	3.53	0.36
$L_{8,89,1}$	18	2	4	4	0	0	2.94	0.3	1.2	0.0	0.0	0.0	0.2	0.4	0.0	X	X	9.67	13.10	0.36
$L_{8,90,1}$	22	2	4	4	0	0	6.43	0.5	3.4	0.0	0.3	0.0	0.3	0.8	0.0	X	X	100.00	98.10	0.40
$L_{8,91,1}$	26	2	4	4	0	0	14.19	0.5	9.7	0.0	0.0	0.0	0.5	1.4	0.0	X	X	217.00	338.00	0.44
$L_{8,92,1}$	10	2	4	4	0	0	0.64	0.3	0.1	0.0	0.0	0.0	0.0	0.1	0.0	X	X	0.60	0.85	0.33
$L_{8,93,1}$	14	2	4	4	0	0	1.23	0.2	0.2	0.0	0.0	0.0	0.0	0.3	0.0	X	X	2.99	4.29	0.34
$L_{8,94,1}$	18	2	4	4	0	0	2.73	0.3	1.2	0.0	0.0	0.0	0.0	0.4	0.0	X	X	13.70	16.40	0.34
$L_{8,95,1}$	22	2	4	4	0	0	6.12	0.4	3.4	0.0	0.3	0.0	0.0	0.8	0.0	X	X	87.30	91.10	0.37
$L_{8,96,1}$	26	2	4	4	0	0	13.93	0.5	9.8	0.0	0.0	0.0	0.0	1.3	0.0	X	X	431.00	428.00	0.40
$L_{8,97,1}$	10	2	4	4	0	0	0.66	0.2	0.1	0.0	0.0	0.0	0.0	0.1	0.0	X	X	0.72	0.88	0.33
$L_{8,98,1}$	14	2	4	4	0	0	1.24	0.2	0.2	0.0	0.0	0.0	0.1	0.3	0.0	X	X	3.76	3.69	0.35
$L_{8,99,1}$	18	2	4	4	0	0	2.82	0.3	1.2	0.0	0.0	0.0	0.1	0.4	0.0	X	X	13.70	22.50	0.36
$L_{8,100,1}$	22	2	4	4	0	0	6.26	0.4	3.4	0.0	0.3	0.0	0.1	0.8	0.0	X	X	76.80	91.60	0.38
$L_{8,101,1}$	26	2	4	4	0	0	13.94	0.5	9.7	0.0	0.0	0.0	0.2	1.3	0.0	X	X	324.00	398.00	0.42
$L_{8,102,1}$	10	2	4	4	0	0	0.69	0.2	0.1	0.0	0.0	0.0	0.1	0.1	0.0	X	X	0.70	1.03	0.34
$L_{8,103,1}$	14	2	4	4	0	0	1.38	0.2	0.2	0.0	0.0	0.0	0.1	0.3	0.0	X	X	2.81	3.53	0.36
$L_{8,104,1}$	18	2	4	4	0	0	2.96	0.3	1.2	0.0	0.0	0.0	0.2	0.4	0.0	X	X	9.71	13.20	0.38
$L_{8,105,1}$	22	2	4	4	0	0	6.40	0.5	3.3	0.0	0.3	0.0	0.4	0.8	0.0	X	X	87.90	95.20	0.41
$L_{8,106,1}$	26	2	4	4	0	0	14.17	0.5	9.7	0.0	0.0	0.0	0.5	1.4	0.0	X	X	334.00	200.00	0.46
$L_{8,107,1}$	10	2	4	4	0	0	0.60	0.2	0.1	0.0	0.0	0.0	0.0	0.1	0.0	X	X	1.02	0.83	0.33
$L_{8,108,1}$	14	2	4	4	0	0	1.20	0.2	0.2	0.0	0.0	0.0	0.0	0.3	0.0	X	X	3.05	3.75	0.34
$L_{8,109,1}$	18	2	4	4	0	0	2.69	0.3	1.2	0.0	0.0	0.0	0.0	0.4	0.0	X	X	11.00	17.30	0.35
$L_{8,110,1}$	22	2	4	4	0	0	6.09	0.5	3.3	0.0	0.3	0.0	0.0	0.8	0.0	X	X	81.70	74.80	0.37
$L_{8,111,1}$	26	2	4	4	0	0	13.87	0.5	9.8	0.0	0.0	0.0	0.0	1.3	0.0	X	X	285.00	335.00	0.41
$L_{8,112,1}$	10	2	4	4	0	0	0.67	0.2	0.1	0.0	0.0	0.0	0.0	0.1	0.0	X	X	0.66	1.06	0.34
$L_{8,113,1}$	14	2	4	4	0	0	1.30	0.2	0.2	0.0	0.0	0.0	0.1	0.3	0.0	X	X	2.69	3.51	0.34
$L_{8,114,1}$	18	2	4	4	0	0	2.81	0.3	1.2	0.0	0.0	0.0	0.1	0.4	0.0	X	X	14.70	13.10	0.36
$L_{8,115,1}$	22	2	4	4	0	0	6.23	0.5	3.4	0.0	0.3	0.0	0.1	0.8	0.0	X	X	60.20	65.40	0.39
$L_{8,116,1}$	26	2	4	4	0	0	13.91	0.5	9.7	0.0	0.0	0.0	0.2	1.3	0.0	X	X	208.00	336.00	0.43
$L_{8,117,1}$	10	2	4	4	0	0	0.72	0.2	0.1	0.0	0.0	0.0	0.1	0.1	0.0	X	X	0.72	0.52	0.34
$L_{8,118,1}$	14	2	4	4	0	0	1.40	0.3	0.2	0.0	0.0	0.0	0.2	0.3	0.0	X	X	3.06	4.00	0.35
$L_{8,119,1}$	18	2	4	4	0	0	2.90	0.3	1.2	0.0	0.0	0.0	0.2	0.4	0.0	X	X	12.90	17.50	0.38
$L_{8,120,1}$	22	2	4	4	0	0	6.40	0.4	3.4	0.0	0.3	0.0	0.4	0.8	0.0	X	X	67.90	97.60	0.41
$L_{8,121,1}$	26	2	4	4	0	0	14.29	0.5	9.8	0.0	0.0	0.0	0.5	1.4	0.0	X	X	349.00	325.00	0.47

Benchmarks L_8 with one traslation.

$\mathscr{C}$	d	τ	S	E	V	A	Isot	Ib	Ig	Ir	If	Iv	Ia	Is	Ic	Ins	Top	CA0	CA1	C2d
$L_{9,1,1}$	8	3	2	4	0	0	ERR	ERR	ERR	ERR	ERR	ERR	ERR	ERR	ERR	X	X	X	X	0.33
$L_{9,2,1}$	12	5	6	8	0	0	1.00	0.1	0.0	0.0	0.0	0.0	0.0	0.3	0.0	X	X	X	X	0.37
$L_{9,3,1}$	16	7	16	8	0	0	2.90	0.3	0.1	0.0	0.1	0.0	0.1	1.0	0.0	X	X	X	X	0.40
$L_{9,4,1}$	20	8	10	8	0	0	3.66	0.5	0.4	0.0	0.1	0.0	0.1	1.2	0.0	X	X	X	X	0.59
$L_{9,5,1}$	24	17	20	24	0	0	16.42	0.6	1.5	0.0	1.0	0.0	0.4	6.1	0.0	X	X	X	X	0.63
$L_{9,6,1}$	28	19	14	16	0	0	18.79	0.9	3.5	0.0	1.1	0.0	0.8	5.2	0.0	X	X	X	X	3.67
$L_{9,7,1}$	32	14	40	24	0	0	70.78	4.3	8.7	0.0	4.5	0.0	0.9	22.8	0.0	X	X	X	X	1.46
$L_{9,8,1}$	36	16	18	16	0	0	74.09	6.5	16.4	0.0	3.9	0.0	1.4	23.1	0.0	X	X	X	X	21.27
$L_{9,9,1}$	40	27	44	40	0	0	223.68	13.3	37.6	0.0	20.2	0.0	4.3	62.9	0.0	X	X	X	X	ERR
$L_{9,10,1}$	44	29	22	24	0	0	241.63	17.0	72.8	0.0	19.3	0.0	6.2	56.5	0.0	X	X	X	X	ERR
$L_{9,11,1}$	44	29	20	24	0	0	787.08	173.9	332.8	0.0	33.3	0.0	9.3	102.5	0.0	X	X	X	X	27.34

Benchmarks L_9 with one traslation.

$\mathscr{C}$	d	τ	S	E	V	A	Isot	Ib	Ig	Ir	If	Iv	Ia	Is	Ic	Ins	Top	CA0	CA1	C2d
$L_{10,1,1}$	8	6	6	4	0	0	0.52	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.0	2.24	0.56	0.36	0.43	0.32
$L_{10,2,1}$	12	6	4	4	0	0	0.78	0.2	0.1	0.0	0.0	0.0	0.0	0.2	0.0	9.70	3.00	0.98	1.20	0.33
$L_{10,3,1}$	12	10	10	4	0	0	1.88	0.3	0.2	0.0	0.1	0.0	0.1	0.5	0.0	25.17	8.46	2.32	2.72	0.36
$L_{10,4,1}$	16	12	4	4	0	0	3.60	0.4	1.1	0.0	0.0	0.0	0.2	0.9	0.0	85.22	52.14	7.56	10.50	0.34
$L_{10,5,1}$	16	10	6	4	0	0	2.45	0.2	0.5	0.0	0.0	0.0	0.1	0.7	0.0	64.64	36.47	5.48	6.35	0.34
$L_{10,6,1}$	16	14	8	16	0	0	5.09	0.4	1.1	0.0	0.0	0.0	0.2	1.5	0.0	145.31	84.01	11.50	14.50	0.36
$L_{10,7,1}$	20	12	4	4	0	0	5.47	0.6	1.8	0.0	0.0	0.0	0.3	1.6	0.0	373.73	X	25.80	20.90	0.36
$L_{10,8,1}$	20	10	6	4	0	0	3.76	0.5	0.9	0.0	0.0	0.0	0.2	1.1	0.0	236.77	X	X	X	0.36
$L_{10,9,1}$	20	14	8	16	0	0	9.42	0.5	2.1	0.0	0.2	0.0	0.3	2.8	0.0	0.00	0.00	27.20	32.20	0.40
$L_{10,10,1}$	20	19	10	20	0	0	15.90	0.7	4.4	0.0	0.2	0.0	0.5	4.5	0.0	0.00	0.00	86.90	92.90	0.72
$L_{10,11,1}$	24	19	4	4	0	0	29.86	1.0	10.1	0.0	0.0	0.0	1.0	13.6	0.0	X	X	136.00	322.00	0.49
$L_{10,12,1}$	24	21	6	4	0	0	30.34	1.3	11.5	0.0	0.0	0.0	1.0	10.5	0.0	X	X	290.00	295.00	0.53
$L_{10,13,1}$	24	14	8	12	0	0	13.48	0.9	3.8	0.0	0.2	0.0	0.5	3.9	0.0	X	X	62.80	116.00	0.55
$L_{10,14,1}$	24	19	10	20	0	0	26.12	0.8	8.1	0.0	0.3	0.0	0.8	7.2	0.0	X	X	139.00	143.00	0.92
$L_{10,15,1}$	24	25	12	24	0	0	39.49	1.3	15.0	0.0	0.5	0.0	1.1	9.4	0.0	X	X	336.00	295.00	1.09
$L_{10,16,1}$	28	19	4	4	0	0	50.74	2.7	19.1	0.0	0.0	0.0	1.3	22.5	0.0	X	X	459.00	506.00	0.59
$L_{10,17,1}$	28	21	6	4	0	0	51.27	2.0	22.3	0.0	0.0	0.0	1.4	18.2	0.0	X	X	424.00	630.00	0.68
$L_{10,18,1}$	28	14	8	12	0	0	21.01	1.9	7.1	0.0	0.2	0.0	0.6	5.7	0.0	X	X	115.00	180.00	0.70
$L_{10,19,1}$	28	19	10	20	0	0	41.22	1.8	15.9	0.0	0.4	0.0	1.0	11.4	0.0	X	X	365.00	704.00	1.05
$L_{10,20,1}$	28	25	12	24	0	0	69.20	1.9	28.7	0.0	0.5	0.0	1.5	15.4	0.0	X	X	553.00	942.00	1.36
$L_{10,21,1}$	28	31	14	28	0	0	118.96	3.0	48.8	0.0	1.6	0.0	2.3	23.5	0.0	X	X	2080.00	2330.00	1.72
$L_{10,22,1}$	32	26	4	4	0	0	185.45	3.9	62.8	0.0	0.7	0.0	3.6	103.1	0.0	X	X	INT	1780.00	1.02
$L_{10,23,1}$	32	21	6	4	0	0	74.50	2.9	31.8	0.0	0.5	0.0	1.8	28.5	0.0	X	X	1460.00	1070.00	0.78
$L_{10,24,1}$	32	31	8	12	0	0	183.35	4.5	84.1	0.0	0.9	0.0	8.5	49.7	0.0	X	X	X	X	1.52

Benchmarks L_{10} with one traslation (continued on the next page).

$\mathscr{C}$	d	τ	S	E	V	A	ISOT	IB	IG	IR	IF	IV	IA	IS	IC	INS	TOP	CA0	CA1	C2D
$L_{10,25,1}$	32	19	10	20	0	0	60.24	2.3	22.5	0.0	0.4	0.0	2.6	19.2	0.0	X	X	X	X	1.20
$L_{10,26,1}$	32	25	12	24	0	0	104.23	3.5	42.3	0.0	0.6	0.0	4.3	23.1	0.0	X	X	X	X	1.61
$L_{10,27,1}$	32	31	14	28	0	0	184.68	2.6	73.4	0.0	2.1	0.0	6.7	33.4	0.0	X	X	X	X	2.34
$L_{10,28,1}$	32	31	8	32	0	0	336.27	4.2	119.9	0.0	3.6	0.0	9.8	75.6	0.0	X	X	X	X	3.74
$L_{10,29,1}$	36	26	4	4	0	0	292.76	6.8	102.0	0.0	1.1	0.0	9.8	159.3	0.0	X	X	X	X	1.10
$L_{10,30,1}$	36	33	6	4	0	0	360.93	5.3	170.1	0.0	1.5	0.0	14.7	125.5	0.0	X	X	X	X	1.52
$L_{10,31,1}$	36	33	8	12	0	0	282.76	7.7	132.9	0.0	1.2	0.0	11.1	75.1	0.0	X	X	X	X	1.63
$L_{10,32,1}$	36	19	10	20	0	0	87.83	2.8	35.0	0.0	0.5	0.0	3.4	29.1	0.0	X	X	X	X	1.11
$L_{10,33,1}$	36	25	12	24	0	0	159.76	6.2	67.4	0.0	0.9	0.0	5.7	34.9	0.0	X	X	X	X	1.68

Benchmarks L_{10} with one traslation.

$\mathscr{C}$	d	τ	S	E	V	A	ISOT	IB	IG	IR	IF	IV	IA	IS	IC	INS	TOP	CA0	CA1	C2D
$R_{3,3,1,1}$	14	37	14	16	0	1	9.52	4.7	0.6	0.0	0.5	0.0	0.5	1.0	0.0	X	X	17.60	14.50	MULT
$R_{3,3,1,2}$	18	73	12	16	0	0	32.23	22.1	2.7	0.0	0.3	0.0	2.3	1.7	0.0	X	X	129.00	142.00	178.38
$R_{3,3,1,3}$	18	65	14	12	0	1	26.02	17.9	1.7	0.0	0.9	0.0	1.8	1.0	0.0	X	X	69.50	92.20	MULT
$R_{3,3,1,4}$	18	78	10	12	0	0	36.23	25.2	4.3	0.0	0.3	0.0	2.4	1.3	0.0	X	X	143.00	163.00	172.57
$R_{3,3,1,5}$	18	-	-	-	0	0	ERR	ERR	ERR	ERR	ERR	ERR	ERR	ERR	ERR	X	X	108.00	120.00	134.50

Benchmarks of resultants of degree-3 surfaces with one translation.

$\mathscr{C}$	d	τ	S	E	V	A	ISOT	IB	IG	IR	IF	IV	IA	IS	IC	INS	TOP	CA0	CA1	C2D
$R_{3,4,1,1}$	24	91	10	12	0	0	268.30	221.8	25.0	0.0	3.9	0.0	9.0	2.5	0.0	X	X	INT	INT	PRIM
$R_{3,4,1,2}$	24	96	14	8	0	0	268.16	223.7	25.0	0.0	1.7	0.0	9.1	2.4	0.0	X	X	INT	INT	PRIM
$R_{3,4,1,3}$	24	100	18	20	0	0	280.86	224.0	25.0	0.0	6.2	0.0	9.3	4.9	0.0	X	X	INT	INT	PRIM
$R_{3,4,1,4}$	24	87	14	16	0	0	263.06	217.4	24.8	0.0	1.8	0.0	8.5	3.1	0.0	X	X	INT	INT	PRIM
$R_{3,4,1,5}$	22	66	20	12	0	0	163.91	134.0	13.8	0.0	1.4	0.0	4.5	3.0	0.0	X	X	613.00	508.00	PRIM

Benchmarks of resultants of degree-3 and -4 surfaces with one translation.

A.1.5 Benchmarks on symmetrized random polynomials

The curves on this section are obtained as $S_{i,j}(x,y) = f^2(x,y) + f^2(x,-y)$, where f is a random polynomial of degree i.

$\mathscr{C}$	d	τ	S	E	V	A	ISOT	IB	IG	IR	IF	IV	IA	IS	IC	INS	TOP	CA0	CA1	C2D
$S_{10,1}$	20	65	2	0	0	0	35.15	24.5	9.0	0.0	0.0	0.0	1.0	0.1	0.0	X	X	163.00	222.00	24.59
$S_{10,2}$	20	64	6	0	0	0	48.15	22.8	22.8	0.0	0.0	0.0	1.2	0.3	0.0	X	X	187.00	223.00	PRIM
$S_{10,3}$	20	65	8	0	0	0	34.76	23.4	8.6	0.0	0.0	0.0	1.1	0.5	0.0	X	X	177.00	203.00	PRIM
$S_{12,1}$	24	65	4	0	0	0	ERR	ERR	ERR	ERR	ERR	ERR	ERR	ERR	ERR	X	X	84.50	108.00	PRIM
$S_{12,2}$	24	66	4	0	0	0	244.29	100.9	138.5	0.0	0.0	0.0	3.1	0.3	0.0	X	X	821.00	940.00	PRIM
$S_{12,3}$	24	65	4	0	0	0	ERR	ERR	ERR	ERR	ERR	ERR	ERR	ERR	ERR	X	X	88.80	113.00	PRIM
$S_{14,1}$	28	65	4	0	0	0	386.36	306.2	72.3	0.0	0.0	0.0	4.7	0.4	0.0	X	X	INT	INT	PRIM
$S_{14,2}$	28	65	4	0	0	1	ERR	ERR	ERR	ERR	ERR	ERR	ERR	ERR	ERR	X	X	INT	INT	PRIM
$S_{14,3}$	28	65	4	0	0	1	318.30	237.3	73.3	0.0	0.0	0.0	5.2	0.4	0.0	X	X	INT	INT	PRIM

Benchmarks of symmetrized random polynomials.

A.2 Tested curves

We present in this section the equations of curves that are referred to in Section A.1. For reasons of space, some curves are not shown. Labs' curves can be found on the paper [109]. Translated curves can be easily computed: the polynomial resulting from k translations of a given polynomial P equals $\prod_{i=0}^{k} P(x, y+i)$.

A.2.1 ACS and F polynomials

This section shows the polynomials defining the curves whose benchmarks are shown in Section A.1.1.

$A_1 = y^5 - xy^4 - y^4 - 2y^3x + y^3 + 2y^2x + y^2 + 2xy - y - x - 1$

$A_2 = -26886144y^2x^2 + 45313024x^3 + 16384x^5 + 631918592x^2 + 372736x^4 - 676666512y^2 + 928499520y^3 - 420114176y^4 + 77229056y^5 - 405504y^6 - 16384y^7 + 1561393008x - 264033244y + 975939072y^2x + 22183936y^3x^2 - 245073920y^3x + 66400256xy^4 + 4374528y^2x^3 - 1697953344xy - 2695168yx^3 - 344064xy^5 + 409600yx^4 + 130729984yx^2 + 503455579$

$A_3 = 2 - 14x^3 + 7x^5 - x^7 - 35x^4 - 16y^2 + 14y^3 + 20y^4 - 7y^5 - 8y^6 + y^7 + y^8 + 7x - 7y - 42y^2x - 70y^3x^2 + 35xy^4 + 70y^2x^3 - 35x^3y^4 + 7x^6y - 21x^5y^2 + 21x^2y^5 + 35y^3x^4 + 42yx^2 - 7xy^6$

$A_4 = -3 + 12y^2 + 2y^4 - 12y^6 + y^8 + 12x^2 - 28y^2x^2 + 12y^4x^2 + 4y^6x^2 - 18x^4 + 20y^2x^4 + 2y^4x^4 + 12x^6 - 4x^6y^2 - 3x^8$

$A_5 = 102y^4 + 150y^3x - 150y^3 + 43y^2 + 48y^2x^2 - 144y^2x + 52xy - 48yx^2 + 8x - 4y - 2$

$A_6 = 16x^5 - 20x^3 + 5x - 4y^3 + 3y$

$A_7 = 105y^2x^4 - 80y^3 + 140y^3x^3 - 140y^3\text{x} + 35y^4 - 105y^4x^2 + 48y^5 + 42xy^5 - 42x^2 + 35x^4 - 7x^6 + 32\text{y} + 84\text{xy} - 140yx^3 + 42yx^5 + 210y^2x^2 - 42y^2 - 7y^6 - 8y^7 + 7$

$A_8 = 503745158400000y^2x^2 + 1071172241025921 + 5190521760000000x^3 + 616320000000000x^5 + 6145719528600000x^2 + 2453784000000000x^4 + 64000000000000x^6 + 1222596581848560y^2 - 865873043485440y^3 + 185942031773440y^4 - 461482272768y^5 + 481890304y^6 + 3867042171121200\text{x} - 964421897260968\text{y} + 941052996288000y^2\text{x} + 368793600000y^4x^2 + 18816000000000y^2x^4 - 118029260800000y^3x^2 - 419460195584000y^3\text{x} + 15000761856000xy^4 + 144862720000000y^2x^3 - 1744221527208000\text{xy} - 3512320000000y^3x^3 - 1267367040000000yx^3 - 20652441600xy^5 - 53760000000000yx^5 - 405824000000000yx^4 - 2025865699200000yx^2$

$A_9 = 27975600\text{x} + 27993600xy^4 - 55969200y^2\text{x} - 1558300x^3 + 21700x^5 + 2604360x^2 - 230390x^4 + 3590x^6 + 37065600y^4 - 74528640y^2 - 7277400y^2x^2 + 258940y^2x^4 + 129600y^6 + 4672800y^4x^2 + 1558800y^2x^3 + 37333439$

$A_{10} = 176 - 28y^2x^2 - 32x^3 + 24x^2 + 27x^4 + 32y^2 - 1552y^3 - 752y^4 + 1200y^5 + 568y^6 - 80y^7 - 28y^8 - 384\text{x} + 576\text{y} + 608y^2\text{x} - 108y^4x^2 + 162y^6x^2 - 244y^3x^2 + 192y^3\text{x} + 64xy^4 - 144\text{xy} + 108y^3x^3 - 36yx^3 - 56xy^5 + 144yx^2 - 392xy^6 - 180y^9 - 36y^{10} + 27y^{12} - 108y^7\text{x} + 108xy^9$

$F_1 = 2 - 42y^2\text{x} - 14x^3 + 7x^5 - x^7 - 16y^2 + 14y^3 + 20y^4 - 7y^5 - 8y^6 + y^7 + y^8 + 7\text{x} - 7\text{y} - 70y^3x^2 + 35xy^4 + 42yx^2 + 70y^2x^3 - 35x^3y^4 - 21x^5y^2 - 35yx^4 + 21x^2y^5 + 7x^6\text{y} + 35y^3x^4 - 7xy^6$

$F_2 = y^5 - xy^4 - y^4 - 2y^3\text{x} + y^3 + 2y^2\text{x} + y^2 + 2\text{xy} - \text{y} - 2\text{x} - 1$

$F_3 = y^5 - xy^4 - y^4 - 2y^3\text{x} + y^3 + 2y^2\text{x} + y^2 + 2\text{xy} - \text{y} - \text{x} - 1$

$F_4 = y^4 - 6y^2\text{x} + x^2 - 4y^2x^2 + 24x^3$

$F_5 = x^4 + 2y^2x^2 - x^2 + y^4 - 4y^2$

$F_6 = -3 + 12y^2 + 2y^4 - 12y^6 + y^8 + 12x^2 - 28y^2x^2 + 12y^4x^2 + 4y^6x^2 - 18x^4 + 20y^2x^4 + 2y^4x^4 + 12x^6 - 4x^6y^2 - 3x^8$

$F_7 = x^5 - y^3x^2 + xy^4 - x^4 + y^3\text{x} - y^4 + x^3 + yx^2 + y^2\text{x} - 2x^2 - y^2 + x$

$F_8 = 27975600\text{x} + 27993600xy^4 - 55969200y^2\text{x} - 1558300x^3 + 21700x^5 + 2604360x^2 - 230390x^4 + 3590x^6 + 37065600y^4 - 74528640y^2 - 7277400y^2x^2 + 258940y^2x^4 + 129600y^6 + 4672800y^4x^2 + 1558800y^2x^3 + 37333439$

$F_9 = x^8 + 4x^6y^2 + 6y^4x^4 + 4y^6x^2 + y^8 - 4x^6 - 12y^2x^4 - 12y^4x^2 - 4y^6 + 16y^2x^2$

$F_{10} = x^6 + y^2x^4 - y^4x^2 - 2x^4 - y^6 + 2y^4 + x^2 - y^2 + xy$

$F_{11} = -2 - 2y^2\text{x} + 2y^2x^2 + 6x^3 - x^5 - 6x^2 - 6x^4 - 7x^6 + 3y^2 + 4y^3 + 4y^4 + 5y^5 + y^6 - 5\text{x} - 3\text{y} - 3y^4x^2 - 7y^2x^4 + y^4x^4 - 3x^6y^2 - 3y^3x^2 - y^3\text{x} - 6xy^4 + 4yx^2 - 6y^2x^3 + 3\text{xy} - y^3x^3 + 7x^3y^4 - 5yx^3 - 7x^5y^3 - 7x^5y^2 - 4xy^5 - 6yx^5 + 5yx^4 - 3x^6y^3 + 5x^6y^4 - 3x^2y^5 + 4x^5y^4 - 6x^6y^5 - 4x^5y^5 + 2x^4y^5 - 2x^6\text{y} + 5x^3y^5 + 2y^3x^4$

$F_{12} = 10000000y^8 + 250000000y^4x^4 + 610000000y^6 + 620000000x^5y^4 + 10000000x^6 - 50000000x^2y^5 - 110000000\text{xy} + y$

$F_{13} = y^8 - \text{xy} + x^2$

$F_{14} = -2x^6y^3 + x^5y^4 + x^7y + y^4x^4 + y^6x^2 - 2y^3x^4 - 2x^2y^5 - xy^6 + y^2x^4 - y^4x^2 - y^2x^3 - yx^3 + y^4 - yx^2$

$F_{15} = -x^3 + yx^2 - x^2 - y^2x + x + y^3 - y^2 - y + 1$

$F_{16} = x^6 + 3y^2x^4 + 3y^4x^2 + y^6 - 4y^2x^2$

$F_{17} = y^4 - 6y^2x + x^2 - 4y^2x^2 + 24x^3$

$F_{18} = 1250000000000000000000000000000y^4 - 127499999999000000000000000000000y^2x^2 + 18999999999600000000000000000000y^2x - 1099999999995250000000005000000000y^2 + 18012499999991000000000000000000x^4 - 48099999999928000000000000000000x^3 + 722999999997837500000000900000000000x^2 - 5799999999971100000000036000000000000x + 1939999999855000000000361249999997 - 5000000000000000000000000000000y^2x^4 + 4000000000000000000000000000000y^2x^3 + 30000000000000000000000000000000x^6 - 3600000000000000000000000000000000x^5$

$F_{19} = 3889197 + 35414400y^2x - 24085600y^2x^2 - 33036800x^3 - 2880000x^5 + 40724680x^2 + 13729600x^4 + 240000x^6 - 3074760y^2 - 10545600y^3 + 10118200y^4 - 4432000y^5 + 1116000y^6 - 160000y^7 + 10000y^8 - 21484320x + 16791840y - 1012000y^4x^2 - 952000y^2x^4 - 40000y^4x^4 + 8096000y^3x^2 - 11904000y^3x + 1488000xy^4 + 31574400yx^2 + 7616000y^2x^3 - 46425600xy - 2560000y^3x^3 + 320000x^3y^4 - 9984000yx^3 + 1248000yx^4 + 320000y^3x^4$

$F_{20} = 9 - 116y^2x^2 - 52x^2 - 2x^4 - 4x^6 + 4y^2 - 40y^3 - 34y^4 + 8y^5 + 20y^6 + 8y^7 + y^8 + 24y + 36y^4x^2 + 4y^6x^2 + 12y^2x^4 + 6y^4x^4 + 4x^6y^2 - 16y^3x^2 - 136yx^2 - 24yx^4 + 24x^2y^5 + x^8 + 8x^6y + 24y^3x^4$

$F_{21} = 8999999 - 116000000y^2x^2 - 52000000x^2 - 2000000x^4 - 4000000x^6 + 4000000y^2 - 40000000y^3 - 34000000y^4 + 8000000y^5 + 20000000y^6 + 8000000y^7 + 1000000y^8 + 24000000y + 36000000y^4x^2 + 4000000y^6x^2 + 12000000y^2x^4 + 6000000y^4x^4 + 4000000x^6y^2 - 16000000y^3x^2 - 136000000yx^2 - 24000000yx^4 + 24000000x^2y^5 + 1000000x^8 + 8000000x^6y + 24000000y^3x^4$

$F_{22} = 9000001 - 116000000y^2x^2 - 52000000x^2 - 2000000x^4 - 4000000x^6 + 4000000y^2 - 40000000y^3 - 34000000y^4 + 8000000y^5 + 20000000y^6 + 8000000y^7 + 1000000y^8 + 24000000y + 36000000y^4x^2 + 4000000y^6x^2 + 12000000y^2x^4 + 6000000y^4x^4 + 4000000x^6y^2 - 16000000y^3x^2 - 136000000yx^2 - 24000000yx^4 + 24000000x^2y^5 + 1000000x^8 + 8000000x^6y + 24000000y^3x^4$

$F_{23} = 899997 - 4y^2x - 11599998y^2x^2 - 12x^3 + 2x^5 - 5199983x^2 - 199997x^4 - 400001x^6 + 400005y^2 - 3999992y^3 - 3400001y^4 + 799996y^5 + 1999999y^6 + 800000y^7 + 100000y^8 + 2x + 2399996y + 3599997y^4x^2 + 400000y^6x^2 + 1199997y^2x^4 + 600000y^4x^4 + 400000x^6y^2 - 1600008y^3x^2 + 2xy^4 - 13599976yx^2 + 4y^2x^3 - 2400004yx^4 + 2400000x^2y^5 + 100000x^8 + 800000x^6y + 2400000y^3x^4$

$F_{24} = -64y^2x - 51199808y^2x^2 + 64x^3 + 48x^5 + 25600096x^4 - 12800008x^6 + 64y^3 + 25600096y^4 + 48y^5 - 12800008y^6 +$

$14400020y^8 - 448000440y^4x^2 - 70400120y^6x^2 - 448000440y^2x^4 + 35199848y^4x^4 - 70400120x^6y^2 - 352y^3x^2 - 192y^3\mathrm{x} - 272xy^4 - 64yx^2 - 352y^2x^3 - 384y^3x^3 - 320x^3y^4 - 192yx^3 - 224x^5y^3 - 336x^5y^2 - 224xy^5 - 224yx^5 - 272yx^4 - 112x^6y^3 - 32000020x^6y^4 - 336x^2y^5 - 56x^5y^4 - 56x^4y^5 + 14400020x^8 - 112x^6\mathrm{y} - 224x^3y^5 - 320y^3x^4 - 32x^7\mathrm{y} - 112xy^6 + 12y^9 - 6400002y^{10} + 2399999y^{12} - 32y^7\mathrm{x} - 112x^3y^6 - 4x^3y^8 - 32000020x^4y^6 - 48y^7x^2 - 8x^4y^7 - 48x^7y^2 - 4y^8\mathrm{x} - 32x^7y^3 + 12x^9 - 4x^{11} - 6400002x^{10} - 8x^7y^4 + 7199997y^8x^4 - 51200018y^8x^2 - 51200018y^2x^8 + 7199997y^4x^8 - 41600000y^6x^6 + 4800000y^{10}x^2 - 4y^{11} + 2399999x^{12} - 32x^3y^7 - 4x^8y^3 - 4x^8\mathrm{y} + 4800000x^{10}y^2 + 800000x^{10}y^4 + 4000000x^6y^8 - 800000y^{14} + 100000y^{16} - 800000x^{14} + 100000x^{16} + 800000y^{10}x^4 + 400000y^{12}x^4 + 2400000y^{12}x^2 + 4000000x^8y^6 + 600000y^8x^8 + 2400000y^2x^{12} + 400000y^4x^{12}$

$F_{25} = 1000000000y^8 + 25000000000y^4x^4 + 61000000000y^6 + 62000000000x^5y^4 + 1000000000x^6 - 5000000000x^2y^5 - 11000000000\mathrm{xy} + y$

A.2.2 Resultants

This section shows the polynomials defining the curves whose benchmarks are shown in Section A.1.3.

$R_{4,1} = -3204585761197660x^3\mathrm{y} + 3350823374458840\mathrm{x} - 601079220575120\mathrm{y} + 14896847244459988x^3 - 6570955599802608x^2\mathrm{y} - 4123557051985672x^2 - 607154654235978x^4 - 5855834724117062x^2y^2 - 1950992951697496y^2 - 13227572133843177y^4 - 1853981814550348\mathrm{xy} + 12232930427590842xy^2 - 1497297529313628y^3 + 2521118398372771y^3\mathrm{x} - 16722336708205328x^5 + 22545720388740052x^4\mathrm{y} - 8802113188185197y^4\mathrm{x} + 4429206015042170x^3y^2 - 10464271751664979x^2y^3 + 1452680074935499y^5 + 10655990441588686x^4y^2 + 8163777596263678x^3y^3 + 5724366091291728y^6 - 23953746403992260x^6 + 9570075082788107y^5\mathrm{x} + 9448812122819188x^2y^4 + 14159018049536329x^5\mathrm{y} + 3703974977594532y^7 + 24041624609433485x^4y^3 - 33897438972748944xy^6 - 50243653489157731x^3y^4 - 32376430246859608x^5y^2 + 14738367332180176x^5y^3 + 6146698809743264x^2y^6 + 4460223444447946x^4y^4 - 33053572472214274x^6y^2 - 4809569347914943y^5x^2 + 6118692285215861y^7\mathrm{x} + 20884383236961522y^5x^3 + 32014481161340009x^6\mathrm{y} - 1152698873647610x^7\mathrm{y} + 15851936053184863x^7 - 13939370113685234x^8 + 19351265413842757y^8 - 5948371401652143x^8\mathrm{y} + 27314264788261409x^7y^2 - 11958928823532724x^4y^5 + 11121299366243278x^6y^3 + 7562625154452461x^5y^4 - 18942057499465854x^3y^6 - 3016440980676604x^7y^3 + 15578899618033140x^9 - 1482300027776904y^9 + 7652946202603089x^{10} + 17609236106072672x^{11} - 14840765556352028x^{12} - 5543193267121072y^{10} - 2550998246625043y^{11} - 11705304186847352y^{12} + 93619025313470386x^6y^4 - 41461115419797176x^9\mathrm{y} -$

$4942576361143444x^4y^6 - 84024798595860056x^5y^5 + 53030828038550732x^8y^2 + 58761488499050823x^3y^8 + 86911102022456140x^7y^4 - 22997770286955248x^4y^7 - 143008067725717402x^6y^5 - 58630588989646212x^8y^3 + 35057558781919095x^8y^4 + 4917757705092632x^5y^7 + 58009010403293945x^7y^5 + 33135591009703542x^9y^3 + 51457794272760623x^5y^6 - 25238805994202881x^4y^8 - 8047420334432430x^6y^6 + 53832957297228488x^9y^2 + 18135202945677879x^{10}y^2 - 24922102066171632x^{10}\mathrm{y} - 4403965690408945x^{11}\mathrm{y} + 1198051142127797x^3y^9 + 19859030890630697y^7x^2 + 10890343015474746xy^8 - 12301543866602639x^3y^7 - 15446610725516880x^2y^8 + 22643259045973428x^2y^9 - 11055890680348140x^2y^{10} - 16638698844636419y^9\mathrm{x} + 33758936971837665y^{10}\mathrm{x} - 16371342752542071y^{11}\mathrm{x} + 37940033069933746x^{10}y^3 - 7087404306097790x^7y^6 - 41920098613409297x^9y^4 - 5148225146868163x^{11}y^2 - 56886556379208093x^6y^7 + 41322569925167274x^8y^5 + 22195582555795487x^{12}\mathrm{y} + 26324428092170870x^5y^8 - 33580589938352678x^4y^9 + 37265452722004300y^{10}x^3 - 17605687793056415y^{11}x^2 - 4350928210527568y^{12}\mathrm{x} + 2435484446581215x^{13} + 422341671045264y^{13} - 5840160462081208x^{14} + 7572486275783469x^{13}\mathrm{y} - 30373202004655651x^{12}y^2 + 55132887056768808x^9y^5 + 25287026619201071x^{11}y^3 - 4316941210294396x^{10}y^4 + 27739654289770972x^8y^6 + 13215997495110479x^5y^9 - 52585622607193077x^7y^7 - 9347309957882964x^6y^8 + 21518245365702287x^4y^{10} + 8200765929070391y^{13}\mathrm{x} - 5672512424104302y^{11}x^3 + 13221021293198900y^{12}x^2 + 1810378275992631y^{14} + 493695992259180y^{15} - 6557656914226565x^{12}y^3 - 41376679699176146x^5y^{10} + 67085119887470427x^{11}y^4 + 3356447941220162x^{14}\mathrm{y} - 29393493784999622x^9y^6 + 57889912012347067x^{10}y^5 - 7450976905574736x^{13}y^2 + 67004605467912814x^8y^7 - 91688077592098684x^7y^8 + 103407888397653340x^6y^9 - 16638638369399773x^4y^{11} - 12225013847882460y^{14}\mathrm{x} - 8861260070168542y^{12}x^3 - 16256438590123730y^{13}x^2 + 2614839706342848x^{15} - 4278017769789080x^{14}y^2 + 13530423471811460x^2y^{14} - 19507417075360849x^{10}y^6 - 21086578018578532x^7y^9 + 3066040854582449x^9y^7 + 44635061379472618x^{11}y^5 + 27892282644480562x^6y^{10} - 27876096647154334x^8y^8 + 4668747112518957x^{12}y^4 + 10765460297698126x^{13}y^3 - 47862137079809516x^5y^{11} + 41558978952484878x^4y^{12} - 27719492345214638x^3y^{13} + 7743143193151416y^{15}\mathrm{x} - 541904871768853x^{16} + 2519398052814249y^{16} - 1694107169167618x^{15}\mathrm{y} + 3200802986149776$

$R_{4,2} = -1252519552523009x^3\mathrm{y} + 87480866200\mathrm{x} + 5464761993699806x^3 + 335186119189118x^2\mathrm{y} - 57754839322096x^2 - 925330127352055x^4 - 6325692879190430x^2y^2 - 57719334400y^2 + 78358697268920y^4 - 7178681270460\mathrm{xy} - 38534907354680xy^2 + 4344553068800y^3 + 612266217684479y^3\mathrm{x} - 7676699403607645x^5 + 515958600229993x^4\mathrm{y} + 7162499850893997y^4\mathrm{x} - 16579327682607195x^3y^2 + 7217798390740975x^2y^3 + 286998365653220y^5 + 3930149646013408x^4y^2 - 6847385462634277x^3y^3 + 1218823436739616y^6 + 21463121284393096x^6 + 1524270368247309y^5\mathrm{x} - 3494384601146459x^2y^4 + 10494108469026404x^5\mathrm{y} + 5786130665555067y^7 - 52487335106542400x^4y^3 + 25054353871166177xy^6 + 17504450292378565x^3y^4 - 19774121043437362x^5y^2 + 12077078787021437x^5y^3 +$

$43709146063393902x^2y^6 - 13892110745009792x^4y^4 - 18213256599626301x^6y^2 + 3036423807294708y^5x^2 - 1023698793088196y^7x + 14633597360905368y^5x^3 + 205823793882702x^6y - 1954580263576866x^7y + 5518356183441985x^7 - 23075440488281093x^8 - 3368056300122225y^8 + 25783838083215113x^8y + 22599147538671702x^7y^2 + 9703960597504587x^4y^5 - 24567082366198024x^6y^3 - 49738389272164238x^5y^4 - 17365589701224646x^3y^6 - 30000278054022664x^7y^3 + 21804491460710289x^9 + 1724831267930027y^9 + 15147231588228627x^{10} - 18277541902790778x^{11} + 4225146381478719x^{12} - 15559110443838163y^{10} + 509872549992063y^{11} + 671495281494010y^{12} + 7657944631538768x^6y^4 + 2162675060877299x^9y + 66791909207356862x^4y^6 - 65273659233201741x^5y^5 - 23090510134897100x^8y^2 + 70717602578328893x^3y^8 - 44490706390563270x^7y^4 - 1832398603365336x^4y^7 + 43269787656739311x^6y^5 + 8060884951758296x^8y^3 + 15041808857249478x^8y^4 + 62879724223139551x^5y^7 - 35893119911539706x^7y^5 - 29137896356049638x^9y^3 - 1487351770200597x^5y^6 - 14344906876357247x^4y^8 - 37812529201404247x^6y^6 + 4483061516292066x^9y^2 + 14633150554966822x^{10}y^2 - 9410791964505743x^{10}y + 16602017082877857x^{11}y + 35970472935668327x^3y^9 + 53687307499603725y^7x^2 + 3997958118778854xy^8 + 45116587439936634x^3y^7 + 4025181112192826x^2y^8 - 29160359634874227x^2y^9 - 3966249286215688x^2y^{10} + 10694781662496228y^9x - 4878544130414203y^{10}x - 13408824369790895y^{11}x + 20773743758426766x^{10}y^3 + 43293843391447527x^7y^6 - 29121453312637460x^9y^4 - 8685465963334675x^{11}y^2 - 45107746838723850x^6y^7 - 32371685612796608x^8y^5 + 3551211902636932x^{12}y + 31477166777875667x^5y^8 + 33434194143041390x^4y^9 - 32230954156273889y^{10}x^3 + 19454651863651074y^{11}x^2 - 17768789519172605y^{12}x + 8380085535129748x^{13} + 5321398955054510y^{13} - 3306711552113289x^{14} - 5096373490908693x^{13}y + 4592190978169000x^{12}y^2 + 19721508867025550x^9y^5 - 3741528650477253x^{11}y^3 - 2161364782477452x^{10}y^4 - 20388655607077757x^8y^6 - 28841796946332551x^5y^9 + 7818509250639541x^7y^7 + 17320182642472046x^6y^8 + 19695341029187257x^4y^{10} + 9909648068770399y^{13}x - 3446384150516634y^{11}x^3 - 12801756015970726y^{12}x^2 - 1794803584492235y^{14} - 1066744662200312y^{15} - 1886442416485437x^{12}y^3 - 2857179203405034x^5y^{10} + 5828862828777893x^{11}y^4 + 269922889603937x^{14}y - 2478246589825058x^9y^6 + 1615813908662x^{10}y^5 + 686258005865128x^{13}y^2 + 3397311874551436x^8y^7 - 5791208186417172x^7y^8 + 1869713790431494x^6y^9 + 93578527652784x^4y^{11} + 3578104985371476y^{14}x + 2455349718701531y^{12}x^3 - 3826513212601492y^{13}x^2 - 1313592160474464x^{15} - 38592331122218x^{14}y^2 + 1861991210989451x^2y^{14} + 2616375002235284x^{10}y^6 + 2308345172029518x^7y^9 - 5260356813731240x^9y^7 - 473928769394134x^{11}y^5 - 4715098982563709x^6y^{10} + 1763182819837315x^8y^8 - 1827616236505141x^{12}y^4 + 1825071895830566x^{13}y^3 + 4174655993754714x^5y^{11} - 543499674411917x^4y^{12} - 1649226399200234x^3y^{13} - 1106234628246384y^{15}x + 516546210888409x^{16} + 261352618291216y^{16}$

$R_{4,3} = - 50628272220306x^3y - 9761486570685x - 217919062800y - 27029843555085x^3 + 29801644220979x^2y$

$+ 21303666504486x^2 + 6732052650252x^4 + 23546305320678x^2y^2 + 33000252525y^2 + 10635449799375y^4 -$
$11123698425660xy - 1649235122100xy^2 - 5206280358150y^3 + 17205021503820y^3x + 12668204437749x^5 +$
$62765922305106x^4y - 6475278321405y^4x - 42068040619527x^3y^2 - 44301632768406x^2y^3 - 4782458700y^5 +$
$71098088900589x^4y^2 + 30607395667272x^3y^3 - 2010008864250y^6 - 19074725099820x^6 - 27613946340720y^5x +$
$3685168822761x^2y^4 - 40814500616271x^5y - 19074075736500y^7 + 36942953975109x^4y^3 - 10100070492840xy^6 +$
$17693705707317x^3y^4 - 61315016344533x^5y^2 - 130907346138135x^5y^3 + 71497556827029x^2y^6 - 56216040719148x^4y^4$
$+ 60376516450227x^6y^2 + 32053701594114y^5x^2 + 35770560846570y^7x - 27457020318186y^5x^3 - 9737863893504x^6y$
$+ 45616543648830x^7y + 36381516279003x^7 - 13735113304572x^8 + 19447466624550y^8 - 111622097559438x^8y +$
$61144075005228x^7y^2 + 70813056679722x^4y^5 + 55400139468009x^6y^3 + 20440731939336x^5y^4 - 127199793156669x^3y^6$
$- 49427362693665x^7y^3 - 3404865890268x^9 + 6709424679450y^9 + 6423041127078x^{10} - 26023158115686x^{11} +$
$8619138872640x^{12} - 3538415060550y^{10} - 20343043493100y^{11} + 19273282981200y^{12} - 21342344609925x^6y^4 +$
$37278781354611x^9y + 67034724651792x^4y^6 + 25329076752231x^5y^5 - 41615158527243x^8y^2 + 10404729389673x^3y^8 +$
$56177395471215x^7y^4 + 125732832691248x^4y^7 - 64297698220647x^6y^5 - 107712002043891x^8y^3 + 153414766931562x^8y^4$
$- 87867408789909x^5y^7 + 104353152235110x^7y^5 - 33644147306805x^9y^3 + 19952356452408x^5y^6 - 35750973052512x^4y^8$
$- 134279448160068x^6y^6 + 61210042283664x^9y^2 - 10788593838831x^{10}y^2 + 30256269796647x^{10}y - 21405830915448x^{11}y$
$+ 24190939704159x^3y^9 - 54099400116069y^7x^2 + 23118538273560xy^8 - 15479736323661x^3y^7 - 2085649330833x^2y^8 +$
$12131772689979x^2y^9 + 69585347285505x^2y^{10} - 49413879046530y^9x - 25474174220700y^{10}x + 16706205090360y^{11}x -$
$8892998925021x^{10}y^3 + 17658821784411x^7y^6 + 9638028671346x^9y^4 - 19215188776872x^{11}y^2 + 60843549313719x^6y^7 -$
$90262778469534x^8y^5 + 30775127433534x^{12}y + 16038496640268x^5y^8 - 41289358170915x^4y^9 - 65905422517458y^{10}x^3$
$- 7161119973369y^{11}x^2 + 21987279925110y^{12}x + 8119339478532x^{13} + 7858383933600y^{13} - 7742927461788x^{14} +$
$6058502963100x^{13}y - 4015307838123x^{12}y^2 - 109517128601067x^9y^5 + 30256578538686x^{11}y^3 + 19539800971695x^{10}y^4$
$+ 40493380100148x^8y^6 - 5973491475234x^5y^9 + 75407154263145x^7y^7 - 70247517809112x^6y^8 + 3377832457320x^4y^{10}$
$- 30946848498405y^{13}x - 1629251464608y^{11}x^3 - 9866213952975y^{12}x^2 - 546669913275y^{14} + 2714241141900 -$
$9048816060525y^{15} + 4444065854847x^{12}y^3 + 127038111121575x^5y^{10} + 19667056850334x^{11}y^4 - 3707230386420x^{14}y +$
$40736838396627x^9y^6 - 3758588361846x^{10}y^5 - 14234187334050x^{13}y^2 - 6648509216844x^8y^7 - 131494228033914x^7y^8 +$
$22771298636919x^6y^9 - 2520944107605x^4y^{11} - 13138313694300y^{14}x - 14881629404664y^{12}x^3 + 15776120253510y^{13}x^2$
$- 206406391752x^{15} - 2215202328495x^{14}y^2 - 4237944713100x^2y^{14} + 8523205348878x^{10}y^6 + 57704235378606x^7y^9 +$
$5858738539686x^9y^7 - 11079414073845x^{11}y^5 + 29392174282104x^6y^{10} - 6925538847852x^8y^8 - 6149990171169x^{12}y^4 -$
$5163545568438x^{13}y^3 - 82436065157772x^5y^{11} - 16113533931972x^4y^{12} + 50337082630845x^3y^{13} - 3127339485450y^{15}x +$
$1709204583060x^{16} + 11142493558875y^{16} + 1603783512396x^{15}y$

$R_{4,4} = 196897788421950x^3y + 151829206534275x - 332015090506272y + 124473585991105x^3 - 232088723205499x^2y$
$- 73774806459468x^2 + 59324853301446x^4 + 341729798670915x^2y^2 + 129365466777197y^2 - 225946538614454y^4 -$
$275491124376510xy - 130842411750291xy^2 + 213081007565657y^3 + 34546090541132y^3x - 160470033674126x^5 +$
$360058904781529x^4y + 542394475808547y^4x - 589387487783419x^3y^2 - 550458894113092x^2y^3 + 46408594081812y^5 -$
$12610138830494x^4y^2 + 567699638736747x^3y^3 + 227318510868472y^6 + 184219830788180x^6 - 390042849998696y^5x +$
$22176077743218x^2y^4 - 1055045070916752x^5y - 174898921138550y^7 - 591999194325753x^4y^3 + 110239016133040xy^6 -$
$184724311144734x^3y^4 - 170092640248967x^5y^2 + 751132915268950x^5y^3 - 411572001177938x^2y^6 + 1330630952856418x^4y^4$
$- 827500087553234x^6y^2 + 424763540651024y^5x^2 + 210245117799540y^7x - 997776720169472y^5x^3 + 458456204173133x^6y$
$+ 602171945906509x^7y - 63504634836013x^7 - 314759946960411x^8 - 23931214928330y^8 - 960314104662102x^8y +$
$1287092721506594x^7y^2 + 506702364203608x^4y^5 - 1001545332567511x^6y^3 - 948887447151166x^5y^4 + 940965045666332x^3y^6$
$+ 528549165249361x^7y^3 + 282374819140793x^9 + 73553657813138y^9 - 81423948597105x^{10} - 229754646390742x^{11} +$
$218823612046685x^{12} - 43391705241216y^{10} - 19362743885187y^{11} + 30570194767339y^{12} + 544519831557451x^6y^4 +$
$423146922757315x^9y - 880062340798424x^4y^6 + 329332497968699x^5y^5 - 1149313804959055x^8y^2 - 223606546877767x^3y^8 -$
$211758421635220x^7y^4 + 514119124612780x^4y^7 - 1080246054272950x^6y^5 + 425162513480513x^8y^3 + 786824847273076x^8y^4$
$- 811324844424169x^5y^7 + 1423617911763566x^7y^5 - 1495984284397554x^9y^3 + 811795650556235x^5y^6 -$
$93772086232062x^4y^8 - 450951492294450x^6y^6 - 16771581852695x^9y^2 + 556226541315178x^{10}y^2 + 190360844099136x^{10}y -$
$434151854533330x^{11}y + 256000724376479x^3y^9 - 161788815935612y^7x^2 - 160677871202846xy^8 - 232969246829903x^3y^7 +$
$547716080745527x^2y^8 - 319288156315780x^2y^9 + 126983085173047x^2y^{10} - 78510144199170y^9x + 108899216614205y^{10}x -$
$79326875407556y^{11}x + 1549178278105283x^{10}y^3 + 57846634481106x^7y^6 - 1104655952953747x^9y^4 - 497633315881100x^{11}y^2$
$+ 767040214174901x^6y^7 - 979377712865003x^8y^5 + 316727675371860x^{12}y + 377037003155497x^5y^8 - 279139068743200x^4y^9$
$- 90039979209170y^{10}x^3 - 60121989455820y^{11}x^2 + 61990439059206y^{12}x - 130803658647480x^{13} - 10980070787250y^{13} +$
$22264101709200x^{14} - 127956292937460x^{13}y + 180046665513708x^{12}y^2 + 473393201373028x^9y^5 - 931595416688487x^{11}y^3 +$
$971326780145183x^{10}y^4 + 85723644094323x^8y^6 + 63658126681360x^5y^9 - 653212895514161x^7y^7 - 189388182258329x^6y^8 +$
$203218890763974x^4y^{10} - 45308869725780y^{13}x - 16543339794308y^{11}x^3 + 17981972123513y^{12}x^2 + 5154191900775y^{14} -$
$4086377059500y^{15} + 278018588008620x^{12}y^3 - 155024118363547x^5y^{10} - 469928136600795x^{11}y^4 + 18397207436400x^{14}y -$
$17467169316348x^9y^6 - 102534649071870x^{10}y^5 - 55506806086680x^{13}y^2 + 371559229966891x^8y^7 + 95320344053798x^7y^8$
$+ 11867676988206x^6y^9 - 58743361445960x^4y^{11} + 22920755481900y^{14}x + 36794278714761y^{12}x^3 - 8262873361845y^{13}x^2$
$+ 745212733200x^{15} + 4753475134200x^{14}y^2 + 6098780162250x^2y^{14} + 11797165860840x^{10}y^6 - 65917675067550x^7y^9 -$
$138102078292635x^9y^7 + 34031275587000x^{11}y^5 + 65803764102695x^6y^{10} + 2469691253970x^8y^8 + 129327730319025x^{12}y^4 -$
$33659394490800x^{13}y^3 + 6155107532780x^5y^{11} + 23540401403870x^4y^{12} - 19931098635765x^3y^{13} - 6190830567000y^{15}x -$

$5010913206000x^{16} + 1406514375000y^{16} + 150276253187993$

$R_{4,5} = -1935474426212x^3\mathrm{y} - 2859080328 + 39427770564\mathrm{x} + 62368431840\mathrm{y} - 22153348038x^3 - 492106648896x^2\mathrm{y} + 136645992888x^2 - 518409772772x^4 - 9390183674664x^2y^2 + 557247361952y^2 + 51174637035648y^4 + 283835203680\mathrm{xy} - 10313558030240xy^2 - 14713819773312y^3 + 13408916780760y^3\mathrm{x} - 848824624652x^5 - 310487661320x^4\mathrm{y} + 81969774704626y^4\mathrm{x} + 27958513996800x^3y^2 + 61320435378610x^2y^3 - 34050893295056y^5 + 48521959517612x^4y^2 + 36620300039792x^3y^3 - 43531277545136y^6 - 667115037848x^6 - 91473320261792y^5\mathrm{x} - 52313258605752x^2y^4 + 2806158824068x^5\mathrm{y} - 64098365015504y^7 - 53672668223754x^4y^3 - 64591710404600xy^6 - 228892899207946x^3y^4 + 60553748104416x^5y^2 - 204765796353128x^5y^3 - 89031800586866x^2y^6 - 96336857569744x^4y^4 + 7108785176968x^6y^2 - 128846165947712y^5x^2 - 142075059692220y^7\mathrm{x} + 120286177479868y^5x^3 + 5989134099880x^6\mathrm{y} + 5897235290468x^7\mathrm{y} + 4304147652x^7 + 528617709926x^8 + 197052492598722y^8 + 2184594877888x^8\mathrm{y} - 23098772971480x^7y^2 + 392258113940648x^4y^5 - 234098524966674x^6y^3 - 110275542908218x^5y^4 - 225093063494216x^3y^6 - 195678704289540x^7y^3 + 572923540668x^9 + 21881975516736y^9 + 397437682816x^{10} + 147682609030x^{11} + 69536580296x^{12} - 147851175636672y^{10} - 109874326188032y^{11} + 150073389559808y^{12} - 46323006147356x^6y^4 + 549196608044x^9\mathrm{y} - 236783977050676x^4y^6 - 277690097249644x^5y^5 - 42878958940992x^8y^2 - 72829172387072x^3y^8 - 250882117073702x^7y^4 + 551465246992944x^4y^7 - 517082105506448x^6y^5 - 128722957035054x^8y^3 - 43974693125608x^8y^4 - 148017349662648x^5y^7 - 218222367000276x^7y^5 - 43810728742008x^9y^3 - 538287725291728x^5y^6 - 340896084358052x^4y^8 + 305532134339848x^6y^6 - 33589249023848x^9y^2 - 17001864604112x^{10}y^2 - 1410375497176x^{10}\mathrm{y} - 329827179056x^{11}\mathrm{y} + 440458042118080x^3y^9 - 36389902375616y^7x^2 + 412785036754992xy^8 - 20965571018160x^3y^7 + 375635988110760x^2y^8 - 108894069210816x^2y^9 + 167249641675456x^2y^{10} + 79375694624128y^9\mathrm{x} - 22405294171904y^{10}\mathrm{x} - 410640487620480y^{11}\mathrm{x} - 26350028893512x^{10}y^3 + 514873494563160x^7y^6 - 53899396841536x^9y^4 - 8994545307672x^{11}y^2 + 4721778142624x^6y^7 + 151976584521192x^8y^5 - 544434386232x^{12}\mathrm{y} - 502806442257552x^5y^8 + 1105332873076224x^4y^9 + 154594014532992y^{10}x^3 - 479309003582464y^{11}x^2 + 134284510753792y^{12}\mathrm{x} + 10297300696x^{13} + 92210791170048y^{13} + 7927241300x^{14} - 4316529504x^{13}\mathrm{y} - 1607011848036x^{12}y^2 + 274595797008828x^9y^5 + 7503097660260x^{11}y^3 + 78770183316820x^{10}y^4 + 588425937481012x^8y^6 + 495728824575552x^5y^9 + 61259975023152x^7y^7 - 36546828873752x^6y^8 + 71875055423296x^4y^{10} + 156526816951296y^{13}\mathrm{x} - 518917266314880y^{11}x^3 + 179868346417152y^{12}x^2 - 95162339082240y^{14} + 4026853712x^{15}\mathrm{y} - 32378419298304y^{15} - 4767240256960x^{12}y^3 - 340616424474368x^5y^{10} - 18041215103704x^{11}y^4 - 86835488328x^{14}\mathrm{y} - 35394858719176x^9y^6 - 27383174918096x^{10}y^5 - 1220120214424x^{13}y^2 + 35101089007680x^8y^7 + 340489018291872x^7y^8 + 213552449718784x^6y^9 + 386954831594496x^4y^{11} + 47559622680576y^{14}\mathrm{x} + 64944593927168y^{12}x^3 - 109854982010880y^{13}x^2 - 1852897680x^{15} + 26050468280x^{14}y^2 + 137841236407296x^2y^{14} + 349039047921082x^{10}y^6 + 12083591652352x^7y^9 + 565926571617300x^9y^7 + 107664967914888x^{11}y^5 + 256545135601088x^6y^{10} + 408421843242322x^8y^8 + 21318102511124x^{12}y^4 + 2493150460724x^{13}y^3 + 340099639135680x^5y^{11}$

$- 203741265749568x^4y^{12} + 21731486739456x^3y^{13} - 96026134818816y^{15}x + 73407194x^{16} + 36164095930368y^{16}$

$R_{5,1} = - 6870692411499298x^3y + 3017034827173375yx^{22} - 1100923349343750x + 159458462656665y^{17}x - 58126750918072673y^{13}x^8 + 10191139488881700y^{13}x^7 - 6589468967495y^{18} + 17949326444796895x^3 - 8371721947184965x^2y - 2689748395796975x^2 - 3430804899033701x^4 - 6186477867955526x^2y^2 + 224364892257900y^2 + 727241076159600y^4 - 327968244013425xy + 2299130542914445xy^2 - 1231054637752050y^3 - 876114196976212x^{23} + 8696168889079538y^{14}x^7 + 25992787326344063y^{14}x^6 + 563131902139615x^{21} - 5648196008638490y^3x - 77406743485637075x^5 + 69314519245752941x^4y - 4312724885372415y^4x - 19164066444608451x^3y^2 + 15362939143195052x^2y^3 + 2676337964777580y^5 + 8081074957676278y^{15}x^6 + 84818830101213009x^4y^2 - 12938141668601406x^3y^3 - 1838352172876400y^6 + 279794439831466332x^6 + 3762581723882522y^5x - 25271175258950810x^2y^4 - 106997244897382847x^5y - 858756588471760y^7 - 55388824044453236x^4y^3 + 7968049992512125xy^6 + 25944832468559733x^3y^4 - 229136225091793329x^5y^2 - 16144872614671205y^{15}x^5 + 252687674355002359x^5y^3 + 1115247459658964x^2y^6 + 33316704793368375x^4y^4 + 464151842655637017x^6y^2 - 16421540271352784y^5x^2 - 4315248427984544y^7x + 61981731552864162y^5x^3 + 98310641562519054x^6y + 129969993767847417x^7y - 563509653205390548x^7 + 759197691455610761x^8 - 781685704855060y^8 - 640690345212238550x^8y - 545976099830935945x^7y^2 - 220544843544155903x^4y^5 - 627810999427308995x^6y^3 - 246872334185938140x^5y^4 - 27840316598518642x^3y^6 + 1255710297636783821x^7y^3 - 754179476778709420x^9 + 1281706551435645y^9 + 547521969911239686x^{10} - 263844504053474408x^{11} + 90336478290702731x^{12} + 1972577566785840y^{10} - 1926645280265355y^{11} - 556862870716105y^{12} + 755710603908893055x^6y^4 + 997587634485032052x^9y + 164338666588625090x^4y^6 + 419819324579523589x^5y^5 + 547489007492768590x^8y^2 - 100283376908250500x^3y^8 - 1132288745171833560x^7y^4 - 300393614802085413x^4y^7 - 660897219469799306x^6y^5 - 1706309897851312626x^8y^3 + 1301504696676691220x^8y^4 + 492343281623809169x^5y^7 + 1294129617589459752x^7y^5 + 1691186944642143211x^9y^3 - 640625100071938530x^5y^6 + 196183033516171071x^4y^8 + 1108919361433803998x^6y^6 - 673144327388420962x^9y^2 + 696289125708976774x^{10}y^2 - 1032215968381335073x^{10}y + 572455757163824533x^{11}y + 142160481908150876x^3y^9 - 13869024105824630y^7x^2 - 2900795745472464xy^8 + 62173642894943848x^3y^7 + 40027956889555419x^2y^8 + 1729705383268871x^2y^9 + 50776810384902148x^2y^{10} + 3187766100336878y^9x - 3614700188504704y^{10}x + 4949679641059253y^{11}x + 4186126846051560x^{20} - 1280325473524809720x^{10}y^3 - 1026554562612167172x^7y^6 - 1365149326340492552x^9y^4 - 780652942636870878x^{11}y^2 - 627682473375084019x^6y^7 - 1636696559491814519x^8y^5 - 6273039103772887y^{16}x^5 + 221785592490921627x^{12}y - 535418846782367306x^5y^8 - 321929638143715849x^4y^9 - 109783199651477093y^{10}x^3 - 11539905090870632y^{11}x^2 - 10110659974453447y^{12}x + 6896219946943638x^{13} - 642190248646295y^{13} - 97203542599555583y^4x^{18} + 6895010967776072y^4x^{19} + 90727895874037741x^{18}y^2 -$

$27085914500848984x^{14} - 635111521374306563x^{13}y + 560026029110762974x^{12}y^{2} + 1312405217495110042x^{9}y^{5} + 424907733130623515x^{11}y^{3} + 971035389434047293x^{10}y^{4} + 896955769244855104x^{8}y^{6} + 1791779980836836 10x^{5}y^{9} + 964016138863569597x^{7}y^{7} + 714793326192046087x^{6}y^{8} + 86575096955596648x^{4}y^{10} + 10300694475507257y^{13}x + 57388398321236190y^{11}x^{3} + 41805635495421801y^{12}x^{2} + 1348424292386380y^{14} + 3027160146740259y^{17}x^{3} - 601460074726097116x^{15}y + 3841530137262093y^{16}x^{4} - 618705403389340y^{15} + 31663181195948483y^{13}x^{9} + 2033811051734489y^{13}x^{10} - 21065660876671820y^{14}x^{8} + 4815959435531237y^{14}x^{9} - 13153264327192045x^{12}y^{3} - 214526599679022784x^{5}y^{10} - 536777983869369653x^{11}y^{4} + 731915157191853005x^{14}y - 614188022979332719x^{9}y^{6} - 711975088950469061x^{10}y^{5} + 163226154530552354x^{13}y^{2} - 896133714981860791x^{8}y^{7} - 277331515062464225x^{7}y^{8} - 73077360412192320x^{6}y^{9} - 125525161614715578x^{4}y^{11} + 1781213866419437y^{14}x - 8479021027722536y^{12}x^{3} + 2283561626957703y^{13}x^{2} + 2245615271651090x^{22} - 280627458862015362x^{11}y^{7} + 23125423108289262yx^{21} + 61020220740996899y^{2}x^{20} - 3126271014299188x^{15} - 407770857082232330x^{14}y^{2} + 2423046198557622x^{2}y^{14} + 367847343662485404x^{10}y^{6} + 292137342278427854x^{7}y^{9} + 405080206775160482x^{9}y^{7} - 201320889162183730x^{11}y^{5} + 287120732037681722x^{6}y^{10} - 44040940440563358x^{8}y^{8} - 149351879131321851x^{12}y^{4} + 76686122150224960x^{13}y^{3} - 4254068313468808x^{5}y^{11} - 31789337932336495x^{4}y^{12} + 3653855716073766x^{3}y^{13} + 1484871253042159y^{15}x - 103805584123471466y^{2}x^{19} - 6430445260075343x^{16} - 92032330001170y^{16} + 8063880124423884y^{12}x^{10} - 14948072477471776y^{12}x^{11} + 6287217547698386y^{8}x^{16} + 198583666145531y^{9}x^{15} - 2138994681332779y^{10}x^{14} - 31922880697666809y^{5}x^{18} + 95618147261381249y^{5}x^{17} - 122288288263831383x^{17}y^{2} - 135101822703822576y^{8}x^{14} - 2302163686686240y^{8}x^{15} - 96912904254446580y^{7}x^{15} + 44841508283162929y^{7}x^{16} - 5034566841378762yx^{23} - 3720364532266257y^{13}x^{11} + 1252129631889896y^{5}x^{19} - 280152382002228y^{17}x^{4} + 190177976291952y^{18}x - 1550277193305323y^{18}x^{2} - 4760568108046085x^{18} + 11532201964082045x^{19} + 8859173339513703y^{6}x^{18} - 31069499110425y^{17} - 40261365259513936x^{17}y^{3} + 6872385804875752x^{17} - 42107369254041557y^{11}x^{11} + 10612927838637862y^{11}x^{12} + 261475279373037xy^{16} - 4604096033160161x^{2}y^{15} - 2365020277506515x^{2}y^{16} - 2185421097769873x^{3}y^{14} + 2344355821490326x^{3}y^{15} - 11754291107316705y^{7}x^{17} - 1983048478398902y^{11}x^{13} - 36985614621661770x^{4}y^{13} - 2328308990989038x^{4}y^{14} - 70519371377914169x^{5}y^{12} - 2190854074283717x^{5}y^{13} + 49148702206698144x^{6}y^{11} + 85765342209163943x^{6}y^{12} - 24184185157564249x^{7}y^{10} + 138296562215020994x^{7}y^{11} - 233285849967596663x^{8}y^{9} - 47649608702266187x^{8}y^{10} + 51266260632360792x^{9}y^{8} + 116896119515837029x^{9}y^{9} - 10988119631405948x^{10}y^{7} - 86656979688517598x^{10}y^{8} - 453180058355041609x^{11}y^{6} + 13969091121032923y^{2}x^{21} + 500897499442842642x^{12}y^{5} + 135012555683315124x^{12}y^{6} + 665120950434470687x^{13}y^{4} - 194680278504200514x^{13}y^{5} - 519598424610545667x^{14}y^{4} + 126519159026279613x^{14}y^{3} + 374563568460408734x^{15}y^{2} - 129381245810496308x^{15}y^{3} - 173893389484084158x^{16}y^{2} + 6664890405193051 0y^{5}x^{16} - 330220678325316900y^{6}x^{15} - 109090748241240420y^{7}x^{14} + 260680186591380326yx^{16}$

$- 61655734931014428yx^{18} - 75997432008654979yx^{17} - 5935614093693100y^2x^{22} + 3561558811921917y^3x^{21} + 269087093727643y^4x^{20} + 82581007241051426x^{19}\mathrm{y} - 84804868554880974y^3x^{18} + 132069583441539728y^4x^{17} + 20890261678355y^{19} + 434981130344841y^{15}x^7 - 1260519064412529y^{15}x^8 + 4070867084428716y^{16}x^6 - 1564255546516172y^{16}x^7 - 2521884943330045y^{17}x^5 + 1965108832221897y^{17}x^6 - 41478170692582y^{18}x^4 + 1757005327453851y^{18}x^3 + 113170827456124673y^6x^{16} - 24433215156030480y^6x^{17} - 820241338085557y^{18}x^5 + 307906229094850y^{18}x^6 + 274434529085702y^{19}\mathrm{x} + 363114753000798y^{19}x^3 - 711669401086286y^{19}x^2 + 153461686784507y^{19}x^4 - 124501908929081y^{19}x^5 - 168129689419482y^{20}x^2 + 12304145665548y^{20}x^3 + 160226304911336y^{20}\mathrm{x} + 29123232139758y^{20}x^4 - 13802365906200y^{21}x^2 + 32369902321400y^{21}\mathrm{x} - 2860467004800y^{21}x^3 + 1882970865000y^{22}\mathrm{x} - 14658655000y^{22}x^2 + 20926641461579254y^3x^{20} - 15760186390119493y^3x^{19} + 1022581752223055y^{14}x^{10} - 64428902410484y^{15}x^9 + 226716633977313y^{16}x^8 - 428710310346929y^{17}x^7 - 18709542879310y^{20} - 9236827575800y^{21} - 1070081815000y^{22} + 5046344117753307y^{12}x^{12} - 625375062981016x^{24} - 1979903786732470x^2y^{17} + 2520729832774173x^3y^{16} + 11106263590513719x^4y^{15} - 21826074158742727x^5y^{14} + 39866322314569119x^6y^{13} + 29416089477118172y^{10}x^{12} + 16278182238159183y^{10}x^{13} + 30503667506162118x^7y^{12} - 139595650809055379x^8y^{11} - 66248622800079299x^8y^{12} - 42529557947387557x^9y^{10} + 29565191980457x^9y^{11} + 24855963841919623x^{10}y^9 + 140070582711931141x^{10}y^{10} - 118244654798522414x^{11}y^8 - 266158894408087531x^{11}y^9 - 28115104903001269x^{12}y^7 - 68205012742862180x^{12}y^8 + 271008569538543439x^{13}y^6 + 417557002402460193x^{13}y^7 + 84807587242772170x^{14}y^6 + 208082796478522532x^{14}y^5 - 332515158526285937x^{15}y^5 + 154024668994662912x^{15}y^4 + 10568661145982929x^{16}y^4 + 152712934557608583x^{16}y^3 - 268011383971 49424yx^{20} + 90421389253460539y^9x^{13} - 32267178399506761y^9x^{14} + 33220446288 5057124y^8x^{13} + 82711722293940939y^9x^{12} - 159831146739948067y^{10}x^{11} + 64335533136868867y^{11}x^{10} + 32811273919089642y^{12}x^9$

$R_{5,2} = 1334459008222371 4784y^{22}x^3 - 106193116189461879x^3\mathrm{y} - 50923254470218920yx^{22} + 4714292053898550\mathrm{x} + 1406541592250586\mathrm{y} + 3464467340919480097y^{17}\mathrm{x} - 3261910869161 1666765y^{13}x^8 + 17117368183197946 2722y^{13}x^7 - 4969246804436653594y^{18} - 29510158848356628x^3 - 26307613427243184x^2\mathrm{y} - 6941113070239800x^2 + 139551031610823123x^4 + 809489530748858478x^2y^2 + 85446047259604503y^2 + 250933910689126800y^4 + 3624937035216129\mathrm{xy} - 453276973471239522xy^2 - 138017255771428956y^3 - 54727059056400x^{23} - 7344709094914247 4521y^{14}x^7 + 14230025975765934 5207y^{14}x^6 - 85361207992414538x^{21} + 864890788738955664y^3\mathrm{x} - 300603850924261824x^5 + 322216796243066538x^4\mathrm{y} + 767401607085558606y^4\mathrm{x} + 1066745266187700 51x^3y^2 + 673169352568658151x^2y^3 + 433052590522724280y^5 + 6714382107240763 3752y^{15}x^6 - 535374357825593817x^4y^2 - 1404821138403870918x^3y^3 + 2966237669928334161y^6 + 250763331591331083x^6 + 9905303116120962458y^5\mathrm{x} + 7284098036467365008x^2y^4 - 706002177188987865x^5\mathrm{y} + 19169836859062431 83y^7 + 753105355308542 36x^4y^3 - 3883731391681220477xy^6 - 10592050346822981928x^3y^4 - 4091752282795817907x^5y^2 - 21158294500269290 3333y^{15}x^5 -$

$12586299121316871886x^5y^3 + 26413968875429832266x^2y^6 - 20850063929328571891x^4y^4 + 5661854321888329994x^6y^2$
$- 20032906044005914288y^5x^2 + 12177325900012318375y^7x + 16865496005390108916y^5x^3 + 904337791526673315x^6y$
$- 1775609863229109927x^7y + 40465355720988546x^7 - 194714925695720676x^8 + 3178457531225605631y^8 +$
$2582810389958743325x^8y - 6028593999401743841x^7y^2 + 13061759593335032480x^4y^5 + 15834002288745121900x^6y^3$
$+ 5026448575421359831 3x^5y^4 - 4102976168840480418 0x^3y^6 + 5341257472009637066x^7y^3 - 316291090017968389x^9$
$+ 3626449381852284517y^9 + 1429393090805146015x^{10} - 2352190872971754235x^{11} + 3072892545814304555x^{12} -$
$9618099723760330 72y^{10} + 8756726013002291319y^{11} + 3440532741953157616y^{12} - 47659593346449769956x^6y^4 -$
$4011667561577289269x^9y - 19257130733199489319x^4y^6 - 7760829399131932735 9x^5y^5 + 7451285267407234787x^8y^2 -$
$52072152015692492329x^3y^8 + 24482461751156498x^7y^4 - 30957012716156081481x^4y^7 + 72208046184257033882x^6y^5 -$
$23501012244554667066x^8y^3 + 54608646390974087333x^8y^4 - 20184522301799952805x^5y^7 - 10289740201628985882x^7y^5 +$
$44712606592723063306x^9y^3 + 66169121732915872122x^5y^6 + 23509172871502734240x^4y^8 - 59033427020218439217x^6y^6 +$
$4943158723827201353x^9y^2 + 1303651772191169536x^{10}y^2 + 8429864204152192209x^{10}y - 4243778709275230931x^{11}y -$
$45177199393622307922x^3y^9 - 28301606075997008855y^7x^2 + 1835196692163251327xy^8 + 22758007600386759549x^3y^7 +$
$1252489647455000860x^2y^8 + 22414389964821244446x^2y^9 - 65892567492545067552x^2y^{10} - 7595158452411746446y^9x +$
$36035551550213118964y^{10}x - 11850713267785964397y^{11}x + 1426355554860913917x^{20} - 44581976910673314486x^{10}y^3 +$
$46604342726297750541x^7y^6 - 48154372409839952652x^9y^4 - 21004595051181953801x^{11}y^2 + 78610892903842977787x^6y^7 -$
$22431462209519881637x^8y^5 + 31007807527273603037y^{16}x^5 - 2464441861171059694x^{12}y - 32325501042443900159x^5y^8 -$
$11936624688922058835 4x^4y^9 - 24180025524478778235y^{10}x^3 + 50820209271177670267y^{11}x^2 + 29152096026641446415y^{12}x$
$- 2457724805608249770x^{13} + 5118231195517016139y^{13} + 3055672666920259658y^4x^{18} + 726290256984149570y^4x^{19} +$
$7985057886730975320x^{18}y^2 + 772383770423397808x^{14} - 4732330723379600904y^{24}x - 5254095499466804878x^{13}y -$
$200178140028166078x^{12}y^2 + 72623980331316802570x^9y^5 + 21517502510234952969x^{11}y^3 + 63876364876007732018x^{10}y^4 -$
$31480092263497263557x^8y^6 + 16905735568649112042 0x^5y^9 - 32909838529888279258x^7y^7 + 94835204915018021250x^6y^8$
$+ 42673334079675881537x^4y^{10} + 14367035282824611135y^{13}x - 102055744547486257612y^{11}x^3 -$
$51811425205516815885y^{12}x^2 + 8211408502314345058y^{14} + 39938356037513043353y^{19}x^6 - 212573116312560950 4y^{25} +$
$92509715697773520636y^{17}x^3 + 54224029867338011 8x^{12}y^{13} - 10448486215131078933x^{11}y^{14} - 8107627329046061753x^{15}y -$
$229369028536070040y^{23}x - 12830510863721440653 4y^{16}x^4 - 7427454274612507298y^{15} - 84831360518117845512y^{13}x^9 -$
$40669208524637384 77y^{13}x^{10} + 41343322532925709888y^{14}x^8 + 46677388503183257872y^{14}x^9 + 32385704452078613813x^{12}y^3$
$- 109469892911427373951x^5y^{10} - 58919367523458301180x^{11}y^4 + 11725465437407464539x^{14}y -$
$26287692670256004624x^9y^6 - 97980080709564905027x^{10}y^5 + 41318528359200798310x^{13}y^2 - 12472980446386901283x^8y^7$
$+ 37693355041595440198x^7y^8 + 27895078726520439276x^6y^9 - 64399381462799056361x^4y^{11} - 31173381145835410654y^{14}x$

$+\ 3019682675179081627 3y^{12}x^3 - 13647395596308457405y^{13}x^2 - 44880847895694456x^{22} - 332273133100711382x^{11}y^7 - 5138494237208120504y^{23} + 235591858128148900yx^{21} - 3167023509928944484y^2x^{20} - 1355687917112321652x^{15} + 620465905331097024y^{24} - 50823643934613870990x^{14}y^2 + 22131141142022395516x^2y^{14} + 9463317257099149179 2x^{10}y^6 - 7541241703373416750 7x^7y^9 - 11466888479506849233x^9y^7 + 6621556814882111287 4x^{11}y^5 + 19012634722197254652 4x^6y^{10} - 17217344000858665860 2x^8y^8 - 4002003675982599268 0x^{12}y^4 - 9776051869992766882 2x^{13}y^3 + 20473404295114332831 7x^5y^{11} - 21988545483500038017x^4y^{12} - 7214614385757675524 6x^3y^{13} + 27393697522732666814y^{15}x + 4907141785115033728y^2x^{19} + 93331135252296000y^4x^{21} - 136683480143000124y^6x^{19} + 3628492893378775878x^{16} + 3726310835231960095y^{16} + 11670024760824834106x^9y^{16} - 6223350602669805971 3y^{12}x^{10} - 6263914183302700990 7y^{12}x^{11} + 5225003326873610274y^8x^{16} - 512176596335508 75y^9x^{15} - 13377939768843613199y^{10}x^{14} - 442190776944969311y^5x^{18} + 3677974601359595653y^5x^{17} - 29483367520134918900x^{17}y^2 - 6741164067893267403y^8x^{14} - 30455987425352360395y^8x^{15} + 19443903513046654615y^7x^{15} - 2022918359079632946 4y^7x^{16} - 2428051678212953357 1y^{21}x^4 - 891561028032000yx^{23} + 14692141728303056181y^{13}x^{11} - 11224033920000x^{24}y + 2464315602110872808x^{13}y^{12} - 88455403916909785x^{14}y^{11} - 2944196597266856088x^{15}y^{10} - 83429305547849632x^{16}y^9 + 1084158428391074118y^5x^{19} - 6388852716048078839 8y^{17}x^4 + 13690185184898078623y^{18}x + 38081607567953090539y^{18}x^2 + 5865142393129438886x^{18} - 4107273157351859220x^{19} - 2242497674412156908y^6x^{18} - 1854859146044578831 2y^{20}x^5 - 33442993914526372 95y^{17} + 31871324811837061 26x^{17}y^3 - 5461858988435563049x^{17} + 76240800944556599418y^{11}x^{11} + 43559224398397531 0y^{11}x^{12} - 33421036844964755808xy^{16} - 62290174589609635181x^2y^{15} + 57882028862075014470x^2y^{16} + 40179071279497330613x^3y^{14} + 74269777395389106 35x^3y^{15} + 65022985173619770 59y^7x^{17} - 14065850579336011765y^{11}x^{13} + 419675913396000x^{23}y^2 + 10384395846269813596 6x^4y^{13} - 14359962220937159971 1x^4y^{14} - 44569316000572067374x^5y^{12} - 12532158886097254 63x^5y^{13} - 64471236226295796949x^6y^{11} + 19431618611309429824 2x^6y^{12} + 25157820548650975258x^7y^{10} - 23079361042092473951x^7y^{11} + 5507552413046809485 0x^8y^9 - 23540514172325256521 2x^8y^{10} + 79452926642487096305x^9y^8 - 80580412373501049411x^9y^9 - 455978982360692864x^{10}y^7 + 32143989304219499614x^{10}y^8 - 72334308577067963732x^{11}y^6 - 11686546404156600x^{22}y^3 - 205252349642260020x^{20}y^5 + 466452617234223888y^2x^{21} + 21252607977876997021x^{12}y^5 - 12218287501985460085x^{12}y^6 + 97401012232175159053x^{13}y^4 - 87114498177968279162x^{13}y^5 - 56122503325526604591x^{14}y^4 + 62521797613080753050x^{14}y^3 + 16761192365449586961x^{15}y^2 + 19661231907296800815x^{15}y^3 + 23272484489277404364x^{16}y^2 - 7018300068987984y^7x^{18} - 1987651173236229534 4y^5x^{16} - 4976993381862209102 7y^6x^{15} - 23824690928002906556y^7x^{14} + 283218222166016739yx^{16} - 15321885022112417794yx^{18} + 10411946036688114069yx^{17} - 5368075032319200y^2x^{22} + 85506046337019060y^3x^{21} - 702393987520842948y^4x^{20} + 9032578351268259987x^{19}y - 4189342542675424446y^3x^{18} - 542918541322123312 4y^4x^{17} + 159432300000x^{25} + 4743712344718791025y^{19} + 16484622546256216198 0y^{15}x^7 - 79355919405245531997y^{15}x^8 + 14149981169938448899y^{16}x^6 - 24263305907870807485y^{16}x^7 - 14834845787661111433 4y^{17}x^5 + 96387524374039050890y^{17}x^6$

$- 1186868125481592290 7y^{18}x^4 - 12254952978035392697y^{18}x^3 + 12909194357899656724y^6x^{16} + 9197814026997244935y^6x^{17}$
$- 2321970060618088259 2y^{18}x^5 - 8677070585513992439y^{18}x^6 - 23077027646302747032y^{19}x + 7048339024713331418 4y^{19}x^3$
$+ 3898887366729305819 9y^{19}x^2 - 7993794787623144954 6y^{19}x^4 - 1826218071781778883 8y^{19}x^5 + 5682369227377878871y^{20}x^2$
$+ 15425193292561474014y^{20}x^3 + 14902069881621381546y^{20}x + 9770781809383632051y^{20}x^4 + 2856808513490258492 4y^{21}x^2$
$- 13558280184161240351y^{21}x + 10432059065841305410y^{21}x^3 - 5425397615604735028y^{22}x - 1457467095665985521y^{22}x^2$
$- 2516434234600838970y^3x^{20} + 7972069449489909639y^3x^{19} + 11049290118787600015x^{10}y^{15} - 3658981240722525824 8y^{17}x^8$
$+ 2886776315136492364y^{18}x^7 - 5595421274057482777 5y^{14}x^{10} + 15702267145390446 89y^{15}x^9 + 3456761436404532322 0y^{16}x^8$
$+ 21334831858226502403y^{17}x^7 - 8303245675919409487y^{20} - 632712405459673298y^{21} - 2196756197639150984y^{22} +$
$5521849193709307942 3y^{12}x^{12} + 15532839450000x^{24} - 9777475266015449706x^2y^{17} - 99400003680998123 47x^3y^{16} +$
$78789146291233514710x^4y^{15} + 58510872115537778328x^5y^{14} - 47219103079639638935x^6y^{13} + 84882506609144921523y^{10}x^{12}$
$+ 15661278806276527887y^{10}x^{13} - 42699936130043119649x^7y^{12} + 79945092467799215872x^8y^{11} -$
$13291047371191191964 0x^8y^{12} + 14868918173116276505 8x^9y^{10} - 13385324166268787099 1x^9y^{11} -$
$34547273940644392104x^{10}y^9 - 22235114149726462 75x^{10}y^{10} - 94020297899329776258x^{11}y^8 + 48582389730366548794x^{11}y^9$
$+ 34448881497277445964x^{12}y^7 + 54673126895969529274x^{12}y^8 + 62969405946116946132x^{13}y^6 +$
$64524672053935977 13x^{13}y^7 - 10690688412479877 85x^{14}y^6 + 47539884415426750626x^{14}y^5 + 12688070673015342786x^{15}y^5 +$
$41732532611290945 46x^{15}y^4 + 54284679558886731 64x^{16}y^4 - 29209618358703323001x^{16}y^3 - 2119162792630311582yx^{20} -$
$15308271359297385911y^9x^{13} + 31197948202993979 14y^9x^{14} + 44950809985166693144y^8x^{13} + 28718327141457454430y^9x^{12}$
$- 1278410685042393 - 12110793149961489981 3y^{10}x^{11} - 40205507963370868534y^{11}x^{10} + 11559178383195864405 1y^{12}x^9 +$
$7575164908459365698y^{23}x^2 + 2062971615934699539y^8x^{17}$

$R_{5,3} = - 86756181343880963x^3y + 8486492048952984x + 5541974659295388y + 33161579674001372y^{17}x$
$- 8014589270427027y^{13}x^8 + 114666389170512346y^{13}x^7 + 1056163081012416y^{18} + 761125129076036x^3 -$
$9085897930576919x^2y + 15472187334130122x^2 - 6943557630221943x^4 - 92468387651133049x^2y^2 + 18448160902796220y^2$
$+ 38867388082222617y^4 + 141938444882033 97xy + 6806754125056093xy^2 + 32895062771698343y^3 +$
$8217258162778692y^{14}x^7 + 173714540459526655y^{14}x^6 - 59335959566027578y^3x - 4793183669838007x^5 -$
$71673253007038335x^4y - 215342533721173084y^4x - 165394531365783294x^3y^2 - 220226681270606214x^2y^3 -$
$39592436463837330y^5 + 109085804250833473y^{15}x^6 + 120144238671266760x^4y^2 - 134988012020580174x^3y^3$
$- 162036856245533138y^6 - 1710023311683111x^6 - 377492963340560225y^5x - 261409850517648021x^2y^4 +$
$27824955872638904x^5y - 238236545608705024y^7 + 622620379884885639x^4y^3 - 347990514507210669xy^6 +$
$498675792595471522x^3y^4 + 366065028398724155x^5y^2 - 125087961232129518y^{15}x^5 + 595783955268257205x^5y^3 +$
$1245453918574180912x^2y^6 + 1377535973588380566x^4y^4 + 212360739048318256x^6y^2 + 117117660050267745y^5x^2$

$+ 70190513662332700y^7x + 1555301079432218642y^5x^3 + 744768895353320003x^6y + 35736451621278558x^7y + 1206103576299430x^7 + 1821235655714571x^8 - 224821350512313092y^8 + 19094063929012853x^8y - 37778480916136487x^7y^2 + 1237183127499638168x^4y^5 - 263255541981020559x^6y^3 + 173808139385678570x^5y^4 + 1975892079167666255x^3y^6 - 38479955020558l706x^7y^3 + 2437961697782021x^9 - 17572159393297631y^9 + 1886041694408699x^{10} - 358923999382824x^{11} - 309501064225714x^{12} + 233650378130292344y^{10} + 284170443999870211y^{11} + 155581886411741986y^{12} - 1074867513078739944x^6y^4 + 15481526673508650x^9y - 644539859606190946x^4y^6 - 1798236446828582908x^5y^5 - 53226567665250538x^8y^2 - 2388146250032083087x^3y^8 - 397912564700202822x^7y^4 - 3649007416141421094x^4y^7 - 1791160084550170692x^6y^5 - 56539870185027155x^8y^3 + 166160423786680088x^8y^4 - 3421208257543069381x^5y^7 + 248503372094252595x^7y^5 - 31060440720805183x^9y^3 - 3484551847994190539x^5y^6 - 4809043290578928066x^4y^8 - 753187538838934540x^6y^6 - 17135915570361009x^9y^2 - 40529011216337439x^{10}y^2 - 4369589443055795x^{10}y - 7748628631019839x^{11}y - 5068096126058489884x^3y^9 + 2319700194238100897y^7x^2 + 997891986786449037xy^8 + 673933146046915463x^3y^7 + 2145749656671841100x^2y^8 + 1645732492538786 77x^2y^9 - 2278735274978597437x^2y^{10} + 1561941448033398716y^9x + 1141625745270079290y^{10}x + 1469124023104571116y^{11}x + 99049307841x^{20} - 3101910493441546x^{10}y^3 + 138776942631563305 6x^7y^6 - 22110393587860113x^9y^4 - 2187549113574298x^{11}y^2 + 1424364329995406524x^6y^7 + 265839987515183635x^8y^5 + 58855960093944581y^{16}x^5 - 86479993777402x^{12}y - 672697736692157239x^5y^8 - 2608859395231037685x^4y^9 - 4465042965869428177y^{10}x^3 - 3347183930013301496y^{11}x^2 - 709448982558187852y^{12}x + 295994551538185x^{13} - 19707249602224670y^{13} + 53057269767478x^{18}y^2 - 39300385622398x^{14} + 2769664626407647x^{13}y + 35452210894778351x^{12}y^2 + 142508190656729 05x^9y^5 + 121882617700289392x^{11}y^3 + 123737432767808275x^{10}y^4 + 174814310943622684x^8y^6 + 2598333236051846993x^5y^9 + 1388312404893338772x^7y^7 + 3277455203458339655x^6y^8 + 1255323670179627253x^4y^{10} - 851623737322776424y^{13}x - 1565743982510738756y^{11}x^3 - 2416710864982287210y^{12}x^2 - 92605293503446212y^{14} - 99072120095132648y^{17}x^3 + 1587233905656140x^{15}y - 249973468530112884y^{16}x^4 - 72930287881101694y^{15} - 16273438376456870y^{13}x^9 + 490243002196500y^{13}x^{10} - 17465494425413860y^{14}x^8 - 323658909887250y^{14}x^9 + 116406583353270808x^{12}y^3 + 337068732780353000 8x^5y^{10} + 280598242250573023x^{11}y^4 + 2139789907825876x^{14}y + 1459278865622283 81x^9y^6 + 398780588397175262x^{10}y^5 + 25598653309150532x^{13}y^2 - 197877930434357802x^8y^7 + 651177205584197650x^7y^8 + 2511603847234483616x^6y^9 + 3445944858200696444x^4y^{11} - 545369768229609829y^{14}x + 1325837166046888426y^{12}x^3 - 813448902242872262y^{13}x^2 + 8888901442996487x^{11}y^7 + 51725335201145x^{15} + 14080143360423928x^{14}y^2 + 298713558342303841x^2y^{14} + 542499297583673613x^{10}y^6 - 696655762641087224x^7y^9 + 485609811615973156x^9y^7 + 334905195413860893x^{11}y^5 + 713133663635149657x^6y^{10} - 249975395945986976x^8y^8 + 146029635450411232x^{12}y^4 + 56453527595549149x^{13}y^3 + 1621046685891289817x^5y^{11} + 2739040978139968878x^4y^{12}$

$+ \ 2103950993712591669x^3y^{13} - 171355248250079638y^{15}\mathrm{x} + 119634535615913x^{16} - 30132657643717016y^{16} - 1007408651892006y^{12}x^{10} - 773555615920500y^{12}x^{11} + 561433122314148x^{17}y^2 + 1450349677418460y^8x^{14} - 97118955928750y^8x^{15} + 408585279456550y^7x^{15} - 40774028419305688y^{17}x^4 + 13482695626510138y^{18}\mathrm{x} - 2588340301528646y^{18}x^2 + 8691556972647x^{18} + 1588360648394x^{19} - 5046824088841892y^{17} + 355680809232903x^{17}y^3 + 15204928572236x^{17} - 3211050251534776y^{11}x^{11} + 945441459732000y^{11}x^{12} + 4355933463779646xy^{16} + 471222282915181255x^2y^{15} + 271343991757625748x^2y^{16} + 1420855826446866115x^3y^{14} + 431825800512364291x^3y^{15} + 915158229325515310x^4y^{13} - 335318823264808823x^4y^{14} - 315353713910995561x^5y^{12} - 1031788734366848832x^5y^{13} - 849333146231411755x^6y^{11} - 617817885503408785x^6y^{12} - 827713165934372181x^7y^{10} - 370264959065146121x^7y^{11} + 74916876680872278x^8y^9 + 229737309198380176x^8y^{10} + 541568858176106580x^9y^8 + 407029004047775068x^9y^9 + 454679177602968786x^{10}y^7 + 164462300963014041x^{10}y^8 + 197407579201422610x^{11}y^6 + 93695750327972541x^{12}y^5 + 16665201448505602x^{12}y^6 + 57894713755107862x^{13}y^4 + 45396730973187736x^{13}y^5 + 33631000236315459x^{14}y^4 + 28978116288452681x^{14}y^3 + 8211124056263385x^{15}y^2 + 14099663302691493x^{15}y^3 + 3014533976814917x^{16}y^2 + 577163354053236y^5x^{16} + 2370614045638873y^6x^{15} + 6607181837454910y^7x^{14} + 1061063798491938yx^{16} + 58451458165220yx^{18} + 318695005961704yx^{17} + 4219592853168x^{19}\mathrm{y} + 8542149513975y^3x^{18} + 72262664524962y^4x^{17} + 695880678554880y^{19} - 7646166555754636y^{15}x^7 - 4060676332734000y^{15}x^8 + 22165809832124574y^{16}x^6 - 3429872512290000y^{16}x^7 + 37872446132843714y^{17}x^5 + 867932069289250y^{17}x^6 + 11917335198710438y^{18}x^4 - 31714978288160786y^{18}x^3 - 21689656565150y^6x^{16} + 5694612977424000y^{18}x^5 + 1646426486654432y^{19}\mathrm{x} - 2328308716888014y^{19}x^3 - 7451343704605010y^{19}x^2 + 5111133458508000y^{19}x^4 - 506186945011440y^{20}x^2 + 1390853367461250y^{20}x^3 - 154369718735232y^{20}\mathrm{x} + 113234798989312y^{20} + 5970877619431290 6x^2y^{17} - 34421283114764051x^3y^{16} - 529158706290016699x^4y^{15} - 598722366132868085x^5y^{14} - 116230705747362073x^6y^{13} + 2429949109522480y^{10}x^{12} + 49666771775250y^{10}x^{13} + 57415380230793164x^7y^{12} + 275553937616918849x^8y^{11} + 54689191343869690x^8y^{12} + 88013458299969546x^9y^{10} - 34002400543337386x^9y^{11} - 58162036366163447x^{10}y^9 - 74700875054093968x^{10}y^{10} - 79939007161608505x^{11}y^8 - 64314395114033903x^{11}y^9 - 7110129071566728x^{12}y^7 - 5886702554361495x^{12}y^8 + 33080365899308171x^{13}y^6 + 20999011476516387x^{13}y^7 + 17587037796568531x^{14}y^6 + 28751451935053528x^{14}y^5 + 7634669471463023x^{15}y^5 + 12355150308937776x^{15}y^4 + 1746277813479810x^{16}y^4 + 3532503084724972x^{16}y^3 + 1510605994923531 + 2066430427258120y^9x^{13} + 273125275400250y^9x^{14} + 9202931941328310y^8x^{13} + 2717293330635731y^9x^{12} - 19457955770902723y^{10}x^{11} - 36099123772653944y^{11}x^{10} - 40335049078264757y^{12}x^9$

$R_{5,4} = 43267353965738294940x^3\mathrm{y} - 9526567654666088676\mathrm{x} - 60068896862916747224\mathrm{y} - 338346542043842111661 0y^{17}\mathrm{x} + 484377561479747123004 0y^{13}x^8 - 190997032262162439829 33y^{13}x^7 + 202443566157928956109 3y^{18} +$

$19432963410097413496 5x^3 - 18585989419282217863 6x^2y + 5739599171912066732 4x^2 + 11106460566970350674 4x^4 -$
$3226535322110281744 34x^2y^2 + 1334995096602935266 8y^2 - 2939312693579060474 00y^4 - 8688482056476276956 1xy$
$+ 1549489912998424073 02xy^2 - 1974712098897992699 64y^3 + 4732875502346242331 221y^{14}x^7 +$
$2042991901350280299 8028y^{14}x^6 + 4561414423969433673 12y^3x - 2995705802035814800 0y^{25} - 4114061889883226347 14x^5$
$+ 918280954543722215961x^4y - 651263868152218562934y^4x - 1049391926770454485443x^3y^2 +$
$1133989266219862064416x^2y^3 + 475495312424655051140y^5 - 5377616921984826204143y^{15}x^6 -$
$1733113263148303100186x^4y^2 + 2743429224197996975933x^3y^3 - 426039709605701771862y^6 - 436224866315725141932x^6$
$+ 143042730579638906877 1y^5x - 2011039267438964741021x^2y^4 + 1155071129852195055972x^5y$
$- 691983713415834828210y^7 - 2564085057099261146239x^4y^3 + 2821398868659225296350xy^6 +$
$2417755926630299945182x^3y^4 + 1423866171491741246922x^5y^2 - 9830473604965171815176y^{15}x^5 -$
$8510313896657262128585x^5y^3 + 8178852438365036473365x^2y^6 + 9375089839573465202905x^4y^4 +$
$3368335275437762699893x^6y^2 - 2800661728666500378942y^5x^2 - 4012003079509228532443y^7x -$
$9838032096193377788315y^5x^3 - 4975215711446137839 2x^6y - 1464403135547961597468x^7y - 534404223700579694 28x^7 +$
$554966822868402178464x^8 + 2076138121468474828266y^8 - 1671224953093103201374x^8y + 1528709308350541622925x^7y^2$
$- 1122523827356888997657 4x^4y^5 - 5268736673115588988629x^6y^3 + 8033423829277061092682x^5y^4 +$
$12610727663911758076274x^3y^6 - 7165577635000867910 23x^7y^3 + 4832898769933823666 37x^9 - 1143030988268695033374y^9$
$- 3038983622023471689 60x^{10} - 4088982366001518930 60x^{11} - 1281158178921493411 86x^{12} - 1047907393149461360386y^{10}$
$+ 331789642343143108889 1y^{11} - 2643573345315579148745y^{12} - 1781390710732762767560x^6y^4 -$
$642768194753710903332x^9y - 10641498045904078946092x^4y^6 + 8542223332490470665688x^5y^5 +$
$799181040579517001712x^8y^2 - 3025978851730119461266 2x^3y^8 - 8913699385973040481085x^7y^4 +$
$3488012827771369876827 5x^4y^7 + 2103358260971189408900 7x^6y^5 + 803269014550127596147 1x^8y^3$
$- 2160552186060014849600x^8y^4 + 2645068328794559501940 7x^5y^7 + 4938941440309834401192x^7y^5$
$+ 258842594520769821193x^9y^3 - 3456797040709231252669 0x^5y^6 - 3095747980599601386970 4x^4y^8$
$- 1476392231051028342717 5x^6y^6 - 161307208159044552940 3x^9y^2 + 409679748880773884864 0x^{10}y^2$
$+ 521189120592362481488x^{10}y - 156678556190890340481x^{11}y + 2926097793903505689410 5x^3y^9 -$
$9150686219873262318443y^7x^2 + 462868087084022679258 3xy^8 + 8743398358627006448526x^3y^7 -$
$8129956373780568840441x^2y^8 + 2058685636396478015378 7x^2y^9 - 2160131287753474138306 3x^2y^{10}$
$+ 352493410417873973048 4y^9x - 1113670529212924045955 4y^{10}x + 840469563734046311593 7y^{11}x$
$+ 158146601456715150825x^{20} - 6835926405085497596722x^{10}y^3 - 924078639463366559577x^7y^6 +$
$3832017005383690737867x^9y^4 + 5164508129562536939619x^{11}y^2 - 3731281777661952384524x^6y^7 -$

$2478901206433388851169x^8y^5 - 3567504521443314438129y^{16}x^5 - 1011019674518844928091x^{12}\mathrm{y} + 1159454238976258827853 7x^5y^8 - 6823296786819966205369x^4y^9 + 1887458740850539810280y^{10}x^3 - 2232662041253023309095y^{11}x^2 - 4341053669073204802035y^{12}\mathrm{x} + 6745115280454345328 40x^{13} + 3954594996581673531 40y^{13} - 1624871307368407537 44y^{24}\mathrm{x} + 1892720870421933823596x^{18}y^2 + 9333534704411472959 61x^{14} - 6513647768787765526 44x^{13}\mathrm{y} - 1178580107322475770161x^{12}y^2 + 1237238813694482541003 9x^9y^5 + 5125397228716782109080x^{11}y^3 - 1775884393415614034365 2x^{10}y^4 - 6802750075376492614444x^8y^6 - 4404204014153455614341 8x^5y^9 - 1141548270868341773741x^7y^7 + 2943870003133787483828 8x^6y^8 + 4153560519279741831219 0x^4y^{10} - 5073823417467049506131y^{13}\mathrm{x} - 3086497543644013996689 2y^{11}x^3 + 1596164312720699256415 1y^{12}x^2 + 504609256983584250989y^{14} + 668368443600394415730y^{17}x^3 + 203535890273477137 32x^{23} + 1293501340616534459 28x^{22} + 2114009896847366386416x^{15}\mathrm{y} - 4108656690964056788719y^{16}x^4 - 494508176333727387217y^{15} + 8643439378273456617341y^{13}x^9 - 886851530343348927665y^{13}x^{10} - 8650032208178867192547y^{14}x^8 - 1177938333081697066561y^{14}x^9 + 1338512125443605210363 4x^{12}y^3 + 1977581431761925434373 9x^5y^{10} - 3310154546496464407163 3x^{11}y^4 + 834889806925685295282x^{14}\mathrm{y} - 4407685344429598798962 3x^9y^6 + 4506768571284599789072 4x^{10}y^5 - 807157189910127632146 4x^{13}y^2 + 4334837168102406316350 6x^8y^7 - 3057104886304237071605 1x^7y^8 + 722958046151761794394 4x^6y^9 - 2645015163607891888354 2x^4y^{11} + 5632139286837828745116y^{14}\mathrm{x} + 2712743955218909410208 8y^{12}x^3 - 1345862286525519574464 6y^{13}x^2 - 384120382511437815595y^{20}x^5 - 3439373524584142107465 1x^{11}y^7 - 251356068793948958723y^{17}x^8 - 913228379177608696814y^{18}x^7 - 156744610898790218646y^{19}x^6 + 165677480881015410627y^7x^{18} + 398763400488043757526x^{15} + 972767284567158762x^{22}y^2 + 472965332714842780710x^{21}y^3 + 4193625377936714962169x^{16}y^7 - 5046505214605549994988x^{17}y^6 + 2293119275199561388788x^{18}y^5 + 422960339095041663147x^{19}y^4 - 131625698704503406411 5x^{20}y^3 - 1215328592615778462240 5x^{14}y^2 + 8853370204179443049165x^2y^{14} - 6235809664694776495391 1x^{10}y^6 + 5441497864594770265560 3x^7y^9 + 6647620530721663395213 7x^9y^7 + 5034410239776945047982 3x^{11}y^5 - 4793942425511826794892 5x^6y^{10} - 5898836299663444872586 3x^8y^8 - 3271536668790405282117 3x^{12}y^4 + 2044772065536258082404 6x^{13}y^3 + 2522425583607749173171 3x^5y^{11} - 1040831294353892600834 2x^4y^{12} - 962083314063980462280x^3y^{13} - 5395467681013965243965y^{15}\mathrm{x} + 2482742322863049656 79x^{22}y^3 - 495160566681977823888x^{16} + 1095244264836537520354y^{16} + 361396368614842166 20y^{12}x^{10} + 5053415079682074682701y^{12}x^{11} + 1242820467415552845138x^{12}y^{13} + 881868797421762303534x^{11}y^{14} - 687312643183033211873x^{10}y^{15} + 719133344638758968907x^{17}y^2 - 2204218162596518918112y^8x^{14} - 1303887772867324842085y^8x^{15} - 482094723946860699350 0y^7x^{15} - 208509584333902228472y^{23}x^2 + 1935123779833817082834x^{15}y^9$

$- 2974595192690899026515y^{17}x^{4} + 2424149909237402440008y^{18}\mathrm{x} - 7524681757270277977212y^{18}x^{2}$
$- 3251819069039852284 41x^{20}\mathrm{y} + 1853366229361880862480x^{19}y^{2} - 1752704326459743064 20y^{6}x^{19} -$
$1038510438384254099652x^{18} - 3406578467511247330 32x^{19} + 1457821101763894181 40x^{23}y^{2} - 2210924579323290454183y^{17}$
$- 3218404140448039090776x^{17}y^{3} - 1148944240075726627128x^{17} + 1043064229944770546138y^{11}x^{11} -$
$5388557737030986538698y^{11}x^{12} + 7085312754840271022216xy^{16} + 1043265805034237824212x^{2}y^{15} +$
$3039044777081930798105x^{2}y^{16} - 1276322957565311489 3046x^{3}y^{14} + 1604439635737903952 5388x^{3}y^{15}$
$- 191068429011600638072y^{23} + 246089172508736544567x^{21} + 3257940751924278720 4461x^{4}y^{13} -$
$1823306100109672156 4444x^{4}y^{14} - 6019551403141749692 4289x^{5}y^{12} + 3348685712536340157 2404x^{5}y^{13} +$
$6248692949784532380 7667x^{6}y^{11} - 3035866675146123934 8636x^{6}y^{12} - 5230204292834274907 9775x^{7}y^{10}$
$+ 606102630719905791 8869x^{7}y^{11} + 2247807363800548918 2075x^{8}y^{9} + 263361939292108422 0487x^{8}y^{10} -$
$1355505746191666477 5407x^{9}y^{8} - 1817522306166552194 1258x^{9}y^{9} + 726807504800176986 9817x^{10}y^{7} +$
$3265964881598600334 2606x^{10}y^{8} - 1060679954587137896 8890x^{11}y^{6} + 2417141257830347647 9372x^{12}y^{5}$
$+ 231954108061382674 60207x^{12}y^{6} - 1987832125671700115 6556x^{13}y^{4} - 783202848861500211 2281x^{13}y^{5}$
$+ 446961852852778711 5211x^{14}y^{4} + 1695349566334698573 7722x^{14}y^{3} - 912866081110618307 6607x^{15}y^{2}$
$+ 660649132774165074 5196x^{15}y^{3} - 352642060194946264 2438x^{16}y^{2} - 1805825599146114775 2784y^{5}x^{16}$
$+ 1013890795010211961 0943y^{6}x^{15} - 714992455764387703 1165y^{7}x^{14} + 137293080473687557 4063x^{20}y^{5}$
$+ 2518905596878580946 777yx^{16} + 1211762353339904169 951yx^{18} + 2248423920516096897 579yx^{17} +$
$349582341915118089738x^{19}\mathrm{y} - 675921690196774993 5894y^{3}x^{18} + 1024884003643563286 1320y^{4}x^{17} -$
$171478330915746863 0879y^{19} + 772323288074244667 8716y^{15}x^{7} + 567948876314578846 2164y^{15}x^{8} +$
$1483881053595633073 010y^{16}x^{6} - 364038076017388286 5597y^{16}x^{7} + 1226033989140566826 690y^{17}x^{5} +$
$992113944547987248591y^{17}x^{6} + 1623804081249750095 563y^{18}x^{4} - 5104019966150886642 138y^{18}x^{3} -$
$32041837581214208901 + 5183366064610965583 08y^{22} - 4460534482207071881 529x^{19}y^{3} + 4835581083461279317 296x^{18}y^{4}$
$- 836516474810349140 8159x^{17}y^{5} + 5197773089902837663 455y^{6}x^{16} - 1176624609535270872 250x^{8}y^{16}$
$+ 142011341818723235 2602y^{18}x^{5} - 165078502786358890 0024y^{19}\mathrm{x} + 750644994488864419 347y^{19}x^{3}$
$+ 506614741590518947 004y^{19}x^{2} + 1014666016786173698 442y^{19}x^{4} - 832176548500381455 381y^{20}x^{2}$
$+ 2025590779925088133 365y^{20}x^{3} - 787205171290885269 986y^{20}\mathrm{x} - 406227680295360401 334x^{9}y^{15} +$
$151375736492354273 9080y^{20} - 930371404607850917 502x^{13}y^{11} + 3270597264935377456 406x^{12}y^{12} +$
$1576648882402854685 148x^{11}y^{13} - 667434523219338678 738x^{10}y^{14} + 390229929759210078 1824x^{14}y^{10}$
$- 2802471396405653636 36y^{21}x^{4} - 985602144219039248 24y^{22}x^{3} + 2398260685158576856 955x^{2}y^{17} -$
$5182794778595081190 195x^{3}y^{16} + 709240612383411330 3881x^{4}y^{15} - 142101920616216076 780x^{5}y^{14} -$

$3852773328889978791235x^{6}y^{13} + 3313869126091902701569y^{10}x^{12} + 7837224157074575432395y^{10}x^{13} - 9605504939825643720092y^{21} + 1856375698010482345018x^{17}y^{8} + 2244717952606623205279 4x^{7}y^{12} + 7539969134673068541 87x^{9}y^{16} - 2311516858734115951 4000x^{8}y^{11} + 4473482499555230608264x^{8}y^{12} + 2143225443050314403995 9x^{9}y^{10} + 1153175672504751895657 5x^{9}y^{11} - 2886786061591144528935 3x^{10}y^{9} - 1814504528050292184205 3x^{10}y^{10} + 2400477648997996376357 7x^{11}y^{8} + 1315846274507934672306 5x^{11}y^{9} - 1792532068301610651009 2x^{12}y^{7} - 7976958059808987738481x^{12}y^{8} + 1853533054793132767925 9x^{13}y^{6} - 9532737434454981715907x^{13}y^{7} + 9899295245470731104332x^{14}y^{6} - 1224377274746102971225 6x^{14}y^{5} - 9848644403897375309690x^{15}y^{5} + 5867299990067753548368x^{15}y^{4} + 5515576544899916569738x^{16}y^{4} + 1696209208006810783932x^{16}y^{3} - 5452462835071711290 84x^{16}y^{8} + 5280075378193780783992x^{17}y^{7} + 9294126461006347413 44y^{21}\mathrm{x} - 4852699551952308553 59x^{21}\mathrm{y} + 1054661938326294616251x^{20}y^{2} - 2615459803152320466678x^{18}y^{6} - 3576933068398169927107y^{9}x^{13} + 4478741688458767316 17y^{9}x^{14} - 1192927674505432216284 0y^{8}x^{13} + 1729276542320364681034 2y^{9}x^{12} - 2550423125108085563918 1y^{10}x^{11} + 1342915116133968198059 4y^{11}x^{10} - 6564279566563660718584y^{12}x^{9} + 2023522207671932950010x^{19}y^{5} - 6637077910277878349 82x^{20}y^{4} - 7456991285814899177 67x^{21}y^{4} - 2847287274576150973 48y^{22}\mathrm{x} + 1563993761871230757522y^{21}x^{2} + 3351307258394646106 50x^{21}y^{2} - 2991383364478840734 42x^{22}\mathrm{y} - 1340367554903407879 68x^{23}\mathrm{y} - 7095699833844779398 8x^{24}\mathrm{y} - 5968295150140286566 05x^{13}y^{12} + 2491502383984318071554x^{14}y^{11} + 9142047098769218444 66x^{15}y^{10} - 2640898526506738869 89x^{16}y^{9} + 6276408534270088094 4y^{24} - 7933657315965244734 2x^{5}y^{19} - 2333533968032679270 38x^{4}y^{20} + 1082286282506658533 04x^{3}y^{21} + 2762136613800202803 20y^{23}\mathrm{x} - 7418402524636938687 54x^{7}y^{17} - 5308031986426845810 56x^{6}y^{18} + 2359696284720232284 68y^{22}x^{2}$

$R_{5,5} = - 392792270813388328 2x^{3}\mathrm{y} - 462696125077267130\mathrm{x} - 31695111533956471\mathrm{y} - 9063802100420642528y^{17}\mathrm{x} - 7933066594651978391y^{13}x^{8} - 2595588531354175154 0y^{13}x^{7} - 3684080826170368896y^{18} + 1188629740022703014x^{3} - 1374548807434647315x^{2}\mathrm{y} + 428651275126455250x^{2} + 234097402722937128x^{4} + 2577484576114553146x^{2}y^{2} + 2932910627289208996y^{2} + 469564813665288181y^{4} + 65598937420314306\mathrm{xy} + 5500368736020605660xy^{2} - 4194955946689359371y^{3} + 2708504428664432462y^{14}x^{7} - 3041035868789125323 7y^{14}x^{6} - 4275502992815816162y^{3}\mathrm{x} - 534391656780023161x^{5} - 1489663868061076448x^{4}\mathrm{y} - 4510254286412644051y^{4}\mathrm{x} - 1732984473517456992x^{3}y^{2} + 4611653437108334268x^{2}y^{3} - 6000001152385749441y^{5} + 1053456485198124960y^{15}x^{6} - 647295734814830821x^{4}y^{2} + 1359387081338817926 5x^{3}y^{3} + 1017780773401049825 0y^{6} + 508753508643373 66x^{6} - 1326266428793981707 2y^{5}\mathrm{x} - 969796766272993041 4x^{2}y^{4} + 3600434869218929421x^{5}\mathrm{y} + 9302021338303855367y^{7} + 246278459351759547x^{4}y^{3} + 1663478858014929951 3xy^{6} - 1298009352452701669 3x^{3}y^{4} - 5367008523798100967x^{5}y^{2} - 3930915856462166063 0y^{15}x^{5}$

$- 477636208935777103x^5y^3 + 3140791144324692962 4x^2y^6 + 13009828772909400393x^4y^4 - 7199599368328465193x^6y^2$
$- 18355038903048230809y^5x^2 + 15775421664507907250y^7x + 935105803546949 19y^5x^3 + 881973980165225309x^6y$
$- 2052306208657905176x^7y + 478060254582552546x^7 - 103507996946793258x^8 - 10275323896092512084y^8 +$
$2794510560787045246x^8y - 3001591817750443218x^7y^2 - 12312777588178715376x^4y^5 + 15237349429012885687x^6y^3$
$+ 14426536294715528698x^5y^4 - 7355984531563774630x^3y^6 + 14778321758190942869x^7y^3 - 977690598543933745x^9$
$- 1890353749832877128y^9 - 452229454937713327x^{10} + 466004921485321536x^{11} - 176838013794473585x^{12} -$
$21163797498558146368y^{10} + 28158412219790809712y^{11} - 11965665547735605024y^{12} - 19411580720482958550x^6y^4 +$
$5172008923350108889x^9y - 22559636671679180299x^4y^6 - 15826877265421118456x^5y^5 - 5806871845197528981x^8y^2 +$
$17785161657197334363x^3y^8 - 18906934278496533101x^7y^4 + 74217205459847127875x^4y^7 + 15534418119870818461x^6y^5 -$
$8296148648276149087x^8y^3 + 4367602491761575432 6x^8y^4 + 51910685493001450022x^5y^7 + 37770112201916264420x^7y^5 -$
$3107961429912849686x^9y^3 - 10758351559529087385x^5y^6 - 10905071220747047350 6x^4y^8 - 12754865609182031519x^6y^6$
$- 8279136652174874733x^9y^2 + 5735030386689756886x^{10}y^2 + 404013978584560610x^{10}y - 2270221936871651185x^{11}y -$
$11200825086614062100x^3y^9 - 18011241677573914148y^7x^2 + 12862090519264226415xy^8 + 17799357409472016793x^3y^7 +$
$8124001825706793149 2x^2y^8 - 11368923578943445864 4x^2y^9 + 35270849611877492592x^2y^{10} - 42142174805917930586y^9x$
$- 2773920995936930319y^{10}x - 6522551273657861116y^{11}x + 114756009218898141x^{20} - 9270251026646601387x^{10}y^3 -$
$67269832195829383826x^7y^6 + 19665928054456430353x^9y^4 + 6458821787617435282x^{11}y^2 - 57695482389939936 79x^6y^7 -$
$41322663124917202731x^8y^5 + 15697359225304448800y^{16}x^5 + 203719819456580180x^{12}y - 64187009708111079885x^5y^8 +$
$13385891262919610315 1x^4y^9 - 37557308516329819397y^{10}x^3 - 44655541098695941506y^{11}x^2 + 47254281990319127292y^{12}x$
$- 442021317635108531x^{13} + 19208654288647520032y^{13} - 21829251840000y^{24}x - 283491553952539791x^{18}y^2 +$
$177441262458004213x^{14} + 1830470996417276955x^{13}y + 308791147974944 07x^{12}y^2 - 13579369034295768339x^9y^5 -$
$9794985325046889526x^{11}y^3 - 70519420598843855 93x^{10}y^4 + 2868419324900517232x^8y^6 + 47225454628192269049x^5y^9 +$
$45654238846610270989x^7y^7 + 82047181721000871288x^6y^8 - 13857777845969235467 5x^4y^{10} - 23000448731773187836y^{13}x$
$- 9061578269464774984y^{11}x^3 + 82034604095611801362y^{12}x^2 - 21637545703468417184y^{14} - 8726389685073423568y^{17}x^3$
$+ 54871960708149871x^{23} - 44828389501794506x^{22} - 2564271739487966671x^{15}y - 9965522139246974160y^{16}x^4 +$
$12262128454308733824y^{15} - 1686291894650397275y^{13}x^9 - 679883534932037375y^{13}x^{10} - 5418657406795849 90y^{14}x^8 +$
$457909471663349456y^{14}x^9 - 3231387026053815491x^{12}y^3 - 34761138717014600222x^5y^{10} + 6103022852084411215x^{11}y^4 -$
$1421517895410865678x^{14}y + 7144842724982408940x^9y^6 + 4418271771251798141 8x^{10}y^5 + 2111571639738189796x^{13}y^2 +$
$4791172776747852469x^8y^7 + 35577458783758343276x^7y^8 - 13582993990123994515 9x^6y^9 + 74467358362782995825x^4y^{11}$
$+ 9558553078916865800y^{14}x + 49021825105656939144y^{12}x^3 - 17776352304395762026y^{13}x^2 + 724510147483216 14y^{20}x^5$
$- 28613435989248262710x^{11}y^7 - 3696857245802153 4y^{17}x^8 + 78163115676229012y^{18}x^7 + 158740202807197825y^{19}x^6$

$- 402414964352735735y^{7}x^{18} + 215692475549678634x^{15} - 297220446405490557x^{22}y^{2} + 408056940434266902x^{21}y^{3} -$
$2401407762655050104x^{16}y^{7} + 5471559705822746937x^{17}y^{6} - 2028583598800268184x^{18}y^{5} - 185456049007213204x^{19}y^{4} +$
$853481120281123015x^{20}y^{3} + 5686176009146814084x^{14}y^{2} - 13754473070876946208x^{2}y^{14} - 5588310303585319179x^{10}y^{6} -$
$8691494316353429358x^{7}y^{9} - 5124345222376508855x^{9}y^{7} + 600142643746974731x^{11}y^{5} + 7618774108357372673x^{6}y^{10} -$
$10706273107943235571x^{8}y^{8} + 2110476242824608437x^{12}y^{4} - 6865841510118961555x^{13}y^{3} + 1686224301509603728x^{5}y^{11} -$
$4159268953727510920x^{4}y^{12} - 2384403747231446603x^{3}y^{13} - 2478962236901031696y^{15}x + 1220547659806812117x^{22}y^{3}$
$+ 260295967229512504x^{16} - 10591034218736974784y^{16} - 271530232957802315 + 16424255948979201854y^{12}x^{10} +$
$8951644165204842162y^{12}x^{11} - 13299080533211527x^{12}y^{13} - 123593236342410887x^{11}y^{14} + 142566558767045691x^{10}y^{15} -$
$8686400931605339036x^{17}y^{2} - 4590499900184114911y^{8}x^{14} - 2236466063603474414y^{8}x^{15} - 4680788015809749042y^{7}x^{15}$
$+ 287634719820800y^{23}x^{2} + 3014495633117771704x^{15}y^{9} + 10207493331179105339y^{17}x^{4} + 5800948061440613504y^{18}x$
$- 9246381804336709744y^{18}x^{2} + 268238936114929244x^{20}y - 683433335811816448x^{19}y^{2} + 444172793232455132y^{6}x^{19}$
$- 191043796448172658x^{18} + 790931038121440622x^{19} - 47528865626964137x^{23}y^{2} + 7776331657977001280y^{17} +$
$2093127447452962919x^{17}y^{3} + 123154369909768153x^{17} - 7003225048335675748y^{11}x^{11} - 4580090415606191713y^{11}x^{12} +$
$19524728515137412528xy^{16} - 2471504294059943456x^{2}y^{15} - 4583188174348211180x^{2}y^{16} - 2093910258679296887x^{3}y^{14}$
$- 2868713905735765328 2x^{3}y^{15} - 712253476915200y^{23} - 143716584675129568x^{21} + 6857847245322976266 3x^{4}y^{13} -$
$4427554203604836424x^{4}y^{14} + 14175549147184374668x^{5}y^{12} - 3302369271016117112x^{5}y^{13} - 1114121556267240179x^{6}y^{11} -$
$3396125241945456381 7x^{6}y^{12} + 5057657614578787108 2x^{7}y^{10} - 2700811583959704200 6x^{7}y^{11} + 6270906402582176279 5x^{8}y^{9}$
$- 1171533062959419537 02x^{8}y^{10} + 6513694798335012163 6x^{9}y^{8} + 1204429468021402257x^{9}y^{9} + 8042463906017195154x^{10}y^{7}$
$+ 12515177127748296237x^{10}y^{8} + 868800680521369394x^{11}y^{6} - 5671030453544111643x^{12}y^{5} + 27641852933812187046x^{12}y^{6}$
$+ 380771977376084336x^{13}y^{4} + 2005724307919159055x^{13}y^{5} + 228547520352482831x^{14}y^{4} - 6516003251052108971x^{14}y^{3}$
$+ 2773388813174335410x^{15}y^{2} + 1428980436433251412x^{15}y^{3} - 624218816080335781x^{16}y^{2} - 860717140497857449y^{5}x^{16}$
$+ 7720352416482183062y^{6}x^{15} - 9058399177024485653y^{7}x^{14} - 103604260137627645x^{20}y^{5} - 275042602553427811yx^{16}$
$- 360992473412073793yx^{18} + 239401459776560609yx^{17} - 108409785441480599x^{19}y + 1440699147335453771y^{3}x^{18} -$
$464733794763754694y^{4}x^{17} + 2206676264225011968y^{19} - 902767186558186169y^{15}x^{7} + 3187686600337998792y^{15}x^{8} +$
$19248941908436604 49y^{16}x^{6} + 3174706086836363855y^{16}x^{7} - 1274021658159612444y^{17}x^{5} + 848943403490691099y^{17}x^{6}$
$- 3340009137963699201y^{18}x^{4} + 3820078185362775640y^{18}x^{3} - 9220962867966382x^{24} + 8086147984727040y^{22} +$
$1359771117303346260x^{19}y^{3} + 121042554345959497x^{18}y^{4} - 382567322217210600 2x^{17}y^{5} + 6147693277107898316y^{6}x^{16} -$
$7683552030167844 72x^{8}y^{16} + 655106184844762408y^{18}x^{5} - 2891550467358696192y^{19}x - 1871340948866434336y^{19}x^{3} +$
$3654643516241063536y^{19}x^{2} + 481920409931589642y^{19}x^{4} - 720724903884842208y^{20}x^{2} + 537276323260831680y^{20}x^{3} +$
$927719461804651456y^{20}x - 753428300013856584x^{9}y^{15} - 855096039445754368y^{20} - 1190727069493750145x^{13}y^{11} +$

$2082790238258049107x^{12}y^{12} - 1831953150444062584x^{11}y^{13} - 2032326710284731709x^{10}y^{14} - 3538497400705965351x^{14}y^{10}$
$- 3965930309915336y^{21}x^{4} - 2395178889987840y^{22}x^{3} + 1305876634620284564 8x^{2}y^{17} + 2370706220795923921 2x^{3}y^{16} +$
$8830119471853468513x^{4}y^{15} + 2651601857213217567 4x^{5}y^{14} + 5360505296153624528 6x^{6}y^{13} - 2865317614049242778y^{10}x^{12}$
$- 5098351781031300987y^{10}x^{13} + 57099565493762560y^{21} - 188395407154576524x^{17}y^{8} + 4232627509253470882 1x^{7}y^{12} +$
$423470558680727036x^{9}y^{16} + 9123977907864958064 8x^{8}y^{11} - 1791979715213590984 4x^{8}y^{12} - 4844151668360907318 0x^{9}y^{10} +$
$3281053382028298665 3x^{9}y^{11} + 1867930107277924196 3x^{10}y^{9} - 1270606716127897637 3x^{10}y^{10} + 1114801080485298272 0x^{11}y^{8}$
$+ 3312722645329069210 0x^{11}y^{9} - 4643351798194983771 2x^{12}y^{7} + 2513915933033774244 5x^{12}y^{8} + 7869701994366213599x^{13}y^{6}$
$- 8864937452016963130x^{13}y^{7} + 1407884033111756525 3x^{14}y^{6} - 3145255561978813061x^{14}y^{5} + 3108764848218210580x^{15}y^{5}$
$- 7668973383357228234x^{15}y^{4} - 3726429725787957501x^{16}y^{4} + 2634828737206720976x^{16}y^{3} - 1554082263981873925x^{16}y^{8}$
$- 1953190371493372172x^{17}y^{7} - 212378443115034752y^{21}x + 319481824802857664x^{21}y - 635450008081223105x^{20}y^{2} +$
$1773301740408950136x^{18}y^{6} + 1565952437198612060 6y^{9}x^{13} + 9132802190865525968y^{9}x^{14} - 7340399096419683346y^{8}x^{13} +$
$5797344307050385909y^{9}x^{12} - 2256416655602216641 0y^{10}x^{11} - 1799065722353554718 5y^{11}x^{10} - 4072516830429319288y^{12}x^{9}$
$- 876767582230944453x^{19}y^{5} - 11241925743117615x^{25} - 331575499058196390x^{20}y^{4} - 143173759819332224x^{21}y^{4}$
$+ 1448097349994240y^{22}x + 56781493549208512y^{21}x^{2} - 453095895620253502x^{21}y^{2} + 158034851530777483x^{22}y$
$+ 9845257992083267x^{23}y + 12690339134062616x^{24}y + 4056128509360272970x^{13}y^{12} + 5207827496858117 4x^{14}y^{11}$
$- 6602777477767665 69x^{15}y^{10} + 8259641933731425 34x^{16}y^{9} + 15717061324800y^{24} - 5070363198451135 38x^{5}y^{19} -$
$1181278953542747 48x^{4}y^{20} - 35840360271379488x^{3}y^{21} + 571718685852160y^{23}x - 1540796278729557620x^{7}y^{17} -$
$629435390735123897x^{6}y^{18} - 1388356274964736y^{22}x^{2}$

A.2.3 Symmetric polynomials

This section shows the polynomials defining the curves whose benchmarks are shown in Section A.1.5.

$S_{10,1} = - 4594124509610696212 8x^{4}y + 2814801634122374348 8x^{2}y^{5} + 3288248026424587633 0x^{6}y^{2} -$
$1471452422489893740 8x^{2}y^{6} - 5091199721251359330 0x^{8}y + 4386360029145644230 4x^{4}y^{5} + 7390938768914190848y^{2}$
$+ 9106056470547085088x^{2} - 1307061128501856870 4y^{3} + 1488131979405002496y^{5} + 1245937565730990433 6x^{6} -$
$1315854132455867904y^{6} - 8037250654592234174x^{8} + 5778727327663720448y^{4} + 1601179991242110519 8x^{10} +$
$2079300265720578283 2x^{6}y^{4} + 2813381248092530384x^{8}y^{3} - 1058168880606641956 2x^{8}y^{4} - 2183427756434112946 4x^{12}y +$
$2738319595558434698 4x^{4}y^{9} - 1249122483852657109 88x^{10}y^{4} - 1662800129594463712 0y^{11} + 2167542622846893353 4x^{12}$

$+ 1411502019635362848y^8 - 1181860144260748800y^9 + 14821994344172039680y^{12} + 2830547037471452425 8x^{14} +$
$27766327107374010752x^4y^2 + 34404766018638022728x^2y^4 + 98432987279208320x^6y - 7210940352791663205 6x^4y^3 +$
$33317862093710855408x^4y^4 + 45852820471280378624x^6y^3 - 69142057514668141924x^8y^2 + 51864968698451826656x^2y^2$
$- 35492769124505655680x^4y^6 - 16271044420712443 20y^{14} + 2058932515819943120 0x^{16} - 1515690512855304000y^{17} +$
$9504320929236807200y^{20} - 26516051788272358688x^2y^8 + 52656526925655862012x^{10}y^2 + 29541758060768059900x^8y^7 -$
$8342382943991484732x^{12}y^4 + 72327656673066344508x^{10}y - 79610768278774049 84x^4y^8 - 14558767557122736128x^2y^3 -$
$43835080154834036128x^8y^5 + 12752961155191878052x^6y^7 - 24820386056710861326x^6y^{12} - 29247209864908295200x^{10}y^9 -$
$8520523849344508100x^8y^6 + 39174098631060057060x^{10}y^6 - 69032527701679620 48x^4y^{14} + 39974226375994749248x^2y^9 -$
$46963602088579151104x^2y^{14} - 40800954227346382 80x^{10}y^7 + 13240864102635779024x^{14}y^6 + 17862388508457980316x^{12}y^3 -$
$8655558937110438128 8x^{14}y^3 - 20064590459988249168x^4y^7 - 77628385565736871040x^6y^5 + 60030105785089053100x^{10}y^3 -$
$23268510676397920448x^6y^9 + 27907892146139797410x^8y^8 + 11504387764653329344x^6y^{10} + 34028149798751393824x^2y^{11} -$
$14097433048744068142x^6y^{14} - 18133716155235357656x^{10}y^8 - 18643611668580573136x^{12}y^7 - 20115223489573218992y^{13}x^4$
$- 6325475294220090825 6x^2y^{10} + 39392487165769434948x^6y^8 - 11304569552797145856x^4y^{10} +$
$22519903458520722624x^4y^{12} - 13693492762162326744x^8y^9 - 19115022786641778860x^8y^{11} + 50011840055093540140x^6y^{13}$
$+ 4754868509184874 56x^4y^{15} - 30560953781797881088x^2y^{17} - 24528716279615356784x^4y^{11} + 49666738431695346976x^6y^{11}$
$+ 25656185030391542968x^8y^{10} + 14625974232434740545 2x^{12}y^5 - 11775222288600945066x^{12}y^6 -$
$36729956467827056176x^{12}y^8 + 13398113152111444658x^{10}y^{10} + 52613453559211308748x^8y^{12} - 12050796707786253392x^6y^6$
$- 11554388131742414696 8x^{10}y^5 + 38107190681198262260x^{12}y^2 - 6344217484606337692x^{14}y^2 +$
$46144263121196121280x^{14}y^4 + 54675966261767474400x^{14}y - 18953164091001117040x^{16}y^2 + 20824029815625346 88x^2y^{12}$
$+ 12748795646071081152x^2y^{15} + 76576243134587797 48x^{14}y^5 + 29074196926752330528x^2y^{16} + 43617557435747054 42x^{16}y^4$
$- 11285083125938338704x^4y^{16} + 19512528538463403552x^2y^{18}$

$S_{10,2} = 2399395100411790936x^2y^5 + 42281643635234735760x^6y^2 - 29979411107934646536x^2y^6 +$
$15484077898375189572x^8y + 25144705405129749260x^4y^5 + 53320850368828873380x^2y^7 + 18830677279583380 82y^2$
$+ 12601735355560050 00y^6 + 3080903839091815800y^4 + 20308684277027947808x^{10} - 12920454694755742384x^6y^4 +$
$43844347583935412032x^8y^3 + 94955538835544980308x^8y^4 + 42775651033757808788x^{12}y + 10655923120616841841 6x^4y^9 +$
$10051317095180493150 8x^{10}y^4 + 10898190929422611780y^{10} - 13372578656246853652y^{11} - 17451569775601956960x^{12}$
$+ 8915313500172891000y^{12} - 50338133368360144200x^{14} + 68133947699368533 64x^4y^2 + 22007055751503988050x^2y^4 +$
$38353647478453331096x^4y^4 + 13186149270105001232x^6y^3 + 28386826968093368410x^8y^2 - 51369006878951988 92x^4y^6 +$
$55069598626397474778x^{16} + 23741294227292038418y^{20} + 72310405473031440798x^2y^8 - 80176660245955552356x^{10}y^2 +$
$18757821425972899512x^8y^7 - 96486722017986945 16x^{12}y^4 + 13886934603936377 84x^4y^8 - 15856423889401145 12x^2y^3 -$

$7496850682649090080 0x^8y^5 + 2891442126364298136x^6y^7 - 7425329981267342053 6x^6y^{12} + 2539701332607826455 6x^{10}y^9 - 7055598036088913847 2x^8y^6 + 3958322816162387658 8x^{10}y^6 - 7432366735333304024 0x^4y^{14} - 2097347483317852708 8x^2y^9 + 3334702742665510350x^2y^{14} - 1188488284278285803 32x^{10}y^7 + 900968051858808921 8x^{14}y^6 - 7471034630882737763 2x^{12}y^3 + 7063319262245557866 4x^{14}y^3 + 4603638654502632633 6x^4y^7 + 5198906101042810282 8x^6y^5 + 6997920661694968064x^{10}y^3 - 5121492335718676048x^6y^9 + 7337615620459226034 6x^8y^8 - 1002049232563043393 36x^6y^{10} + 4435905928039917478 4x^2y^{11} + 1686729324897976976 4x^6y^{14} + 2571789851306331130 2x^{10}y^8 - 6732814884924552878 0x^{12}y^7 + 5128342871286480711 6y^{13}x^4 - 1444395135560230071 4x^2y^{10} - 318411659564985956 28x^6y^8 + 1095177987480256131 82x^4y^{10} - 5179940490512842848 0x^4y^{12} - 6963349122776690182 8x^8y^9 + 139591919210514616x^8y^{11} - 1374626293217364546 0x^6y^{13} + 6946790743512421482 8x^4y^{15} - 2438110551493553028 4x^4y^{11} - 9751121435836689036x^6y^{11} - 2600668118336233287 8x^8y^{10} - 7761389052811928800 4x^{12}y^5 + 3521284593042895027 6x^{12}y^6 + 3594879938559041687 4x^{12}y^8 + 2310501558573173529 6x^{10}y^{10} + 411970925708288391 2x^8y^{12} + 1152060452827911102 88x^6y^6 - 1335411505265677034 12x^{10}y^5 - 1105772125066274896x^{12}y^2 + 4605646794122492528 0x^{14}y^2 - 1523719235767829142x^{14}y^4 - 6343072861735686960x^{14}y - 7014903350931436950 0x^{16}y^2 + 6129480360388584316 6x^2y^{12} - 1553663540807885413 2x^2y^{15} + 3079390797188042755 6x^{14}y^5 - 1977950753717629280 8x^4y^{16} - 1093949737146674940 0y^{13} + 3601204481002528125 0x^{18} - 3869667385052340354 0y^{19} + 1576822805822853405 0y^{18} - 1965893481661695300 0x^{16}y + 3368060122728733328 4x^2y^{13} + 3602538097283114550 0x^{16}y^3$

$S_{10,3} = 2896074851898651038 0x^4y - 957137903678972652 8x^2y^5 + 2787964744292553450 4x^6y^2 + 3455603519103440539 8x^2y^6 + 2701328745663793907 2x^8y - 3899142784337713461 2x^4y^5 + 2624703306612803287 6x^2y^7 - 2900377227143066980 0y^2 - 1393824915313730918 0x^2 + 1922967714363324104 2x^4 - 922966671248290731 2x^6 - 2849960220818413142 0y^6 - 283252797332392151 6x^8 + 3575958912241874420 0y^4 + 764179383464015587 6x^{10} + 8389208370504934638x^6y^4 + 2143167431976915184 4x^8y^3 - 4148140114765894952x^8y^4 - 20499268071474828x^{12}y + 5399636253966053375 6x^4y^9 - 2011601524805839198 4x^{10}y^4 + 2591461927192715820 0y^{10} + 2781613715860538500 0y^{11} - 2641590049098851680x^{12} + 7027596655907487836 0y^8 - 1128051142949628250 0y^9 - 2937458345136122207 8y^{12} - 1115864769186013104x^{14} - 2256427948436192324x^4y^2 - 5289268713749062543 2x^2y^4 - 3431875692180629016 4x^6y - 4248292235621765476 8x^4y^3 - 5798884369459082946x^4y^4 + 6774239701144585232 0x^6y^3 + 4368690360499039240 6x^8y^2 + 6729015917810786162 0x^2y^2 - 6870642591195704580 8x^4y^6 + 514867895774838162x^{16} + 2854783727098864020 0y^{20} + 313834777259609451 26x^2y^8 + 730305341503027832x^{10}y^2 + 4321445177871759236 8x^8y^7 + 32451975391035588x^{12}y^4 + 1683354096753756366 4x^{10}y - 2200014636568522335 2x^4y^8 - 1537865720269350940 0x^2y^3 + 9060785624882340152x^8y^5 + 2960216168304776653 2x^6y^7 + 3672608228488971926 2x^6y^{12} - 3790221210913679406 4x^{10}y^9 + 4112548266085521283 6x^8y^6 + 2936956309562477963 4x^{10}y^6 + 2628454257046548684 0x^4y^{14} + 2109552591570429415 6x^2y^9 + 2625534677080879380 0x^2y^{14} - 2137990357090777951 6x^{10}y^7 - 4456214266639318824x^{14}y^6 - 1379714002079461111 2x^{12}y^3 + 8041833516522076144x^{14}y^3$

$- 474041489210057230000x^4y^7 + 1292669391607235658 8x^6y^5 + 1929640820822783644x^{10}y^3 + 1261938261188286451 2x^6y^9 + 17557460979004236638x^8y^8 - 3797705049022611216x^6y^{10} - 5728999529297298330 0x^2y^{11} + 22320021119604395760x^6y^{14} + 30180428403444275588x^{10}y^8 - 13142988172951693368x^{12}y^7 + 9262515783917518980y^{13}x^4 - 50760953236349137752x^2y^{10} + 92933062472773241948x^6y^8 - 75938788203244384804x^4y^{10} - 5229586719631639556 8x^4y^{12} + 14994789157092942104x^8y^9 + 11351260310641485460x^8y^{11} - 26496971702928693840x^6y^{13} - 25606563830934638000x^4y^{15} + 16343976832897478400x^2y^{17} + 54036429636262938528x^4y^{11} - 653156082926208240x^6y^{11} + 492508933793790544x^8y^{10} + 14357793939682137540x^{12}y^5 + 2305653699123871666x^{12}y^6 - 7381909175112933800x^{12}y^8 + 22996340995364948832x^{10}y^{10} + 18904899105382634888x^8y^{12} - 48966930758992122174x^6y^6 + 27467315990162659300x^{10}y^5 + 2620901623953590340x^{12}y^2 - 1658905814851491344x^{14}y^2 + 372409965852552050x^{14}y^4 + 5317861958390995704x^{16}y^2 + 30559656612957098320x^2y^{12} + 6712343722951382828x^{14}y^5 - 7114455957554640000x^2y^{16} + 1137931552063672802x^{16}y^4 + 5882573291897309328 0x^4y^{16} + 770102131775157122x^{18} - 2485347758681071500 0y^{19} + 5409300731739031250y^{18} - 39631228789395762000x^2y^{13} - 62790967336997271240y^{16} + 27332611653740541500y^{15} + 5881071529465824050$

$S_{12,1} = - 55886728734578830388x^2y + 44528210337591820280x^4y + 99270636997546831000x^2y^5 - 88518836532733645556x^6y^2 - 40520186116030145864x^2y^6 + 70598197685237724716x^8y - 50071396151996961472x^4y^5 + 57557594005782688784x^2y^7 + 2316208128080052322x^2 + 57705212917149009768x^4 - 60058050481799265844x^6 - 67192828751545365000y^6 - 12893458849832196834x^8 + 75221880575157388970x^{10} - 21987354741071154262x^6y^4 + 12892727182131128144x^8y^3 + 18910042570064008257 6x^8y^4 - 11790938449332220652 0x^{12}y - 18779062368107572804x^4y^9 + 16250535779145280094 2x^{10}y^4 - 45215244445017209200y^{11} + 9233361538321085930 4x^{12} + 3628182185099956125 0y^{12} + 58656197644452137498x^{14} + 40016010692866973 56x^4y^2 + 20837532308156153984x^2y^4 + 61268466957713279640x^6y + 60232738062325045748x^4y^3 + 12769233032457910402x^4y^4 + 92243233696003437724x^6y^3 - 19749067942569877 80x^8y^2 + 10378714369449592044 2x^2y^2 - 47024189880246646076x^4y^6 + 194412440608014680x^{16} + 48829360941759199800y^{17} - 55484360117127377196x^2y^8 + 13674140506977026279 8x^{10}y^2 - 47904462871296881812x^8y^7 + 21149453830958873529 2x^{12}y^4 - 10914202666400022499 6x^{10}y + 21015611087145467676x^4y^8 - 32283835342306505816x^2y^3 - 18641727373695197344 8x^8y^5 + 37030855021060136196x^6y^7 + 86942345519756930584x^6y^{12} + 33740204223192655648x^{10}y^9 + 85138983048336230854x^8y^6 - 55824643403232130 88x^{10}y^6 + 28017193462874380568x^4y^{14} - 15408303135884068424x^2y^9 + 56653359190195765 60x^2y^{14} - 22237131504335640000x^{10}y^7 - 67206097264981481934x^{14}y^6 - 96682329701960569540x^{12}y^3 + 94495097355621859 20x^{14}y^3 - 84029119739943723148x^4y^7 + 10521315426838568188 4x^6y^5 - 19424444526047220195 6x^{10}y^3 + 1372237426228766460x^6y^9 - 15907540042152894709 2x^8y^8 + 94041164644707105596x^6y^{10} - 34275563340551508968x^2y^{11} + 69490954750947783378x^6y^{14} - 25275578466509934 56x^{10}y^8 + 80553742055298220464x^{12}y^7 - 12918351360152422888y^{13}x^4 + 19946548295642819608x^2y^{10} +$

$20338449040782634188x^6y^8 - 25514077458290398768x^4y^{10} + 69488733201165167848x^4y^{12} + 83910642009399562364x^8y^9 + 63991060706380786964x^8y^{11} + 62368476048070716 92x^6y^{13} + 40386489663799588568x^4y^{15} + 17099641583974723208x^4y^{11} - 23542969749393836056x^6y^{11} - 11939750319509422 7264x^8y^{10} - 1306362521601907220x^{12}y^5 - 85882194793356691488x^{12}y^6 - 90816636275915688188x^{12}y^8 + 2624437612030348468x^{10}y^{10} + 38469195828462668106x^8y^{12} + 12886326224957049 6368x^6y^6 - 15856028103291185 6152x^{10}y^5 + 16371374703287539 3926x^{12}y^2 + 17247728850194660 5604x^{14}y^2 + 4329700232039994 4132x^{14}y^4 - 2597456162647502 6092x^{14}y + 71434056838316088316x^{16}y^2 + 18506838908238366848x^2y^{12} - 35915030118693326808x^2y^{15} + 46911652071300026760x^{14}y^5 - 94527146785196103100x^2y^{16} - 10838045295274701052x^{16}y^4 - 2740394486143087200x^4y^{16} - 13248523884169844256x^2y^{18} - 3086333882987717 0724x^{18} - 44108727832505645336x^{16}y - 46289781446862628680x^2y^{13} - 11189319581609189052x^{16}y^3 - 8675757814046144948x^{18}y - 50935786707575960176x^{10}y^{11} - 56125313302067371516x^{14}y^7 - 14041970673901112556x^{18}y^3 + 2379187155123723 9896x^{20} + 50518217994443704450x^{22} + 38335235602229766992x^{18}y^4 - 24541521100425418562x^{16}y^6 + 59336074910919561 04x^{16}y^7 - 36856868595261674240x^{14}y^8 + 61913564606437315 28x^{16}y^8 + 16429070870340758792y^{22} + 1133582802061112 6258x^{24} + 55052926775176790170x^{18}y^2 - 29039123224526988976x^{16}y^5 - 44293599195510855040x^8y^{13} + 42643139754067606796x^{12}y^9 + 30970639713413147800x^4y^{17} + 53119165619707051640x^{10}y^{12} + 94968869124974558 76x^{20}y + 45156857746259224184x^{12}y^{10} - 9665679966668608168x^8y^{14} + 80288350239176891 24x^{20}y^3 + 84007173559656716 04x^{10}y^{13} - 27549306094085155264x^{18}y^5 - 17800819189475029192x^{14}y^9 - 20171091599163193552x^8y^{15} + 16755196453895363336x^{20}y^4 + 24542576503029649544x^{10}y^{14} + 24303736025050786080x^{22}y - 27293744479247606312y^{11}x^{12} + 6796551165874570800y^{20}x^2 - 41664361567166393512y^{21}x^2 + 56583930031783768484x^{14}y^{10} - 19307927444705025476x^6y^{16} + 2931290256913565400x^6y^{15} + 9506297764211829280x^{20}y^2 + 9549418066935555200x^{22}y^2 - 28511998280517624320x^{18}y^6 + 12571701077245472x^6y^{18} + 70291722695260500 0x^4y^{18} - 8618076074404049400x^4y^{19} + 2641535602499301 5858x^4y^{20} + 31109767958841605000$

$S_{12,2} = - 3306291159080417320x^4y + 8583492427940479536x^2y^5 + 4588450492068274184x^6y^2 - 18018780817192383676x^2y^6 - 8929435105344182580 0x^8y - 46813014452996402912x^4y^5 - 48732288232094893816x^2y^7 + 707448271220511756 8x^4 + 2810914016738680474 4x^6 + 22561340778315394952y^6 - 31157273368680370184y^7 + 28771284986543776424x^8 - 14120227434608085176x^{10} - 50585898154257127776x^6y^4 - 49715709835691427880x^8y^3 + 13475937724805617 7780x^8y^4 - 51050432040942998688x^{12}y + 50147016886313695 08x^4y^9 + 38382629434639570164x^{10}y^4 + 28677342761880010432y^{11} + 4490861418011617634x^{12} + 10757069951087861282y^8 - 71933184689109565504y^{12} + 142298143703526096x^{14} - 5400130983774862632x^4y^2 + 42837754588958464184x^2y^4 + 27736944009052951 84x^6y + 29774460105859993416x^4y^3 + 25223826529303650566x^4y^4 - 10256286498663647 4728x^6y^3 - 18122049720630961 80x^8y^2 + 811491588843572450x^2y^2 - 30207077541897671764x^4y^6 - 26562021821024387032x^{16} - 33131700722857840256y^{17} -$

$34325412496784925472y^{20} + 24004929060470392380x^2y^8 + 23060403760779364176x^{10}y^2 - 15198843039795087160x^8y^7 + 21207321731182309120x^{12}y^4 + 404521390411455272x^{10}y + 13765920938261975676 6x^4y^8 - 24956694444201212952x^2y^3 + 5265485812595499696 8x^8y^5 + 10185737265087399436 4x^6y^7 - 5916387291344906941 6x^6y^{12} - 34328130694272421636x^{10}y^9 + 24901541934232995096x^8y^6 + 63906181291416333802x^{10}y^6 + 47166788617672224130x^4y^{14} + 52624422817651011892x^2y^9 - 18562010458257074964x^2y^{14} - 13521214656347366922 4x^{10}y^7 + 5648121484828992269 6x^{14}y^6 - 65989480819013952772x^{12}y^3 - 49573765289598745540x^{14}y^3 + 62813103424362356 80x^4y^7 - 26276630618232697524x^6y^5 + 10483897840319625712x^{10}y^3 + 67570800718395417012x^6y^9 - 44403502509156088472x^8y^8 - 28286753925892778732x^6y^{10} - 64249264902685105020x^2y^{11} + 91397692950483708022x^6y^{14} - 12940114490987331707 8x^{10}y^8 + 47920567588929674776x^{12}y^7 + 10747901374800845880y^{13}x^4 + 33601918748395553472x^2y^{10} - 19696169607207919304x^6y^8 - 10994161007903993616x^4y^{10} - 99777863300706101000x^4y^{12} + 65980325550178459416x^8y^9 + 19959061480059368952x^8y^{11} - 20080280423654212696x^6y^{13} + 62289339701359844804x^4y^{15} + 3377047932282052896x^2y^{17} - 98243968076711271748x^4y^{11} - 99855936827483350160x^6y^{11} + 69346516292795056 52x^8y^{10} + 31938867204405996400x^{12}y^5 + 12509631623301204324 2x^{12}y^6 - 12330507285371129678x^{12}y^8 + 39549857428231694764x^{10}y^{10} - 18318353735010676132x^8y^{12} + 77273149532116951402x^6y^6 - 38733200914348256748x^{10}y^5 + 13312746829141837252x^{12}y^2 + 87683378959496415168x^{14}y^2 - 71922446023494731988x^{14}y^4 + 76406753285223390288x^{14}y - 60983179703630326608x^{16}y^2 + 12907745232748643 76x^2y^{12} + 52749930774434692256x^2y^{15} - 59555230726620311356x^{14}y^5 + 15720779364210252022x^2y^{16} - 27559574439479061146x^{16}y^4 + 29713410769341527 28x^4y^{16} + 35996830848762793072y^{13} - 2820240556031399967 6x^{18} + 30114423282694173728y^{18} + 56874656250822253 68x^{16}y + 25945952470967625208x^2y^{13} + 54048716518383235536x^{16}y^3 + 46406593874511299692y^{16} - 54009699233754333784y^{15} + 13636400052740961976 0x^{10}y^{11} + 11139699148010588378 0x^{14}y^7 + 47150165268421318572x^{18}y^3 + 11068751946519469872x^{20} + 18806294334546938304x^{22} + 84463345994032267772x^{18}y^4 + 19597713814571483078x^{16}y^6 + 14569794004558658288x^{16}y^7 + 10504197341875593927 4x^{14}y^8 + 10696476982178676638 2x^{16}y^8 + 7000219304844568002x^{24} + 35770942911043009784x^{18}y^2 - 42665823675655921360x^{16}y^5 - 61987925505601737268x^8y^{13} - 89368168759668336 60x^{12}y^9 - 81648611401367715000x^4y^{17} - 92253304060830374044x^{10}y^{12} - 21200964997881328332x^{12}y^{10} + 55563147050969958594x^8y^{14} + 13526696220433221996x^{20}y^3 + 37634587899224795020x^{18}y^5 - 30458339749199995380x^{14}y^9 + 19079719422358105356x^8y^{15} - 94108675479045073 16x^{20}y^4 + 54650900409014928 98x^{10}y^{14} + 91408316808340214008y^{11}x^{12} + 34987205838492581 52y^{20}x^2 - 54299488402929634 30x^{14}y^{10} - 82125322895816661 32x^6y^{16} + 58601332735507436 00x^6y^{15} - 56717323450025946244x^{20}y^2 + 20410491525668910245 0x^{22}y^2 - 16871264607910114900x^{18}y^6 - 47394184848861765508x^4y^{19} - 65673059491729077520x^4y^{20} + 62398849363510641872y^{21} + 32323517914816196882y^{24} - 54045181872680047060x^2y^{19} - 24485532204065376236x^8y^{16} + 11581383124607700184x^{12}y^{12}$

$S_{12,3} = - 2885197415442351116 8x^2y - 2713244387910272950 4x^4y - 4510690442449887512 8x^2y^5 +$
$7185842187016282674 0x^6y^2 - 5709958229168451195 0x^2y^6 - 5510552745458273594 0x^8y + 4114489140270515955 2x^4y^5$
$+ 4181371139641058144x^2y^7 + 23366669694566923648y^2 + 7971803600784261842x^2 + 8810589157985382760x^4 +$
$23174115485018451080x^6 + 26407591194748927460y^6 - 11181321868027823616x^8 + 35860901011910921458y^4 +$
$9663832841671202592x^{10} + 40406861733205616482x^6y^4 - 21030974648208667400x^8y^3 + 24259943978963080468x^8y^4 +$
$39141468876487124084x^{12}y + 9150181368063592040x^4y^9 + 24417952743483486472x^{10}y^4 + 16326143497966815744y^{10}$
$- 26915656039686403356y^{11} + 17169690963126381474x^{12} + 8096735434741608050y^8 - 21185128637486273152y^9 +$
$30088769128393926768y^{12} - 22158044322059125236x^{14} + 77875248006143185006x^4y^2 - 20683144448643555166x^2y^4 +$
$4442794572142904864x^6y - 32862667884617720232x^4y^3 + 25536979498421007308x^4y^4 - 34906780562777978692x^6y^3 -$
$94380437984008528336x^8y^2 + 17900226518749660604x^2y^2 - 90300790383252473808x^4y^6 + 18450816009153440880y^{14}$
$+ 6783894951761338450x^{16} - 9753692166330883612y^{20} - 27273344340027423272x^2y^8 + 73123646742555253 88x^{10}y^2 -$
$38412451143916376840x^8y^7 - 17192990248103298584x^{12}y^4 + 17333639670307659604x^{10}y + 112234538166392657820x^4y^8 -$
$64307435893315586624x^2y^3 - 2118378408086816236x^8y^5 + 31489807700848476248x^6y^7 + 16811379375813403704x^6y^{12} +$
$28099636835683074432x^{10}y^9 - 82671029791756612788x^8y^6 + 26495364053362657084x^{10}y^6 + 81317146504614279272x^4y^{14}$
$- 58875702261758028464x^2y^9 + 29747633972068893540x^2y^{14} + 52369024139837555468x^{10}y^7 - 22176567400597020958x^{14}y^6$
$- 37086416257620336868x^{12}y^3 + 2400287476351904152x^{14}y^3 + 102060417990545697004x^4y^7 + 51043420224282998496x^6y^5$
$- 22433079075147595 64x^{10}y^3 - 37627947283827748344x^6y^9 - 29950730791173990 16x^8y^8 + 47643428362760317926x^6y^{10}$
$- 11958920808009248403 2x^2y^{11} + 18794868176609231780x^6y^{14} + 24552366100252640262x^{10}y^8 +$
$55941164030543062764x^{12}y^7 + 56847109776472461924y^{13}x^4 - 19220486433489718716x^2y^{10} + 63643116764507075112x^6y^8$
$+ 64093976830601438628x^4y^{10} - 66930192959590020212x^4y^{12} - 2584541328108908864x^8y^9 - 43016990039367096396x^8y^{11}$
$- 12189842986953189204x^6y^{13} + 68352798237310829892x^4y^{15} - 45073447556290336308x^2y^{17} -$
$43651259588026221844x^4y^{11} + 10780263409076050348x^6y^{11} - 34345411023614255880x^8y^{10} + 56059932655717224572x^{12}y^5$
$+ 52474483611220325022x^{12}y^6 + 19274426424075089198x^{12}y^8 - 23375034780058576944x^{10}y^{10} -$
$80891079950170781900x^8y^{12} - 89248199441815697776x^6y^6 - 119561471371834877120x^{10}y^5 + 60884306514188111630x^{12}y^2$
$- 42450133694013189160x^{14}y^2 + 61965361679796040260x^{14}y^4 - 40528227999347260992x^{14}y -$
$15701356223209467758x^{16}y^2 - 63596606507315805900x^2y^{12} + 42323697446548489328x^2y^{15} + 65461876099539831432x^{14}y^5$
$+ 72753416776759890 96x^2y^{16} + 24710671656060289106x^{16}y^4 + 70922598501481263490x^4y^{16} - 3894198289892309836x^2y^{18}$
$- 15902361626609181 04y^{13} + 3388200454202560656x^{18} - 27279644590382662032y^{19} + 17699304796142920178y^{18} -$
$33289905510815572016x^{16}y - 12017086125053672508x^2y^{13} - 35594693017313326 92x^{16}y^3 + 13706475442582248260y^{15} +$
$23018137052630503404x^{18}y - 81084523000554592424x^{10}y^{11} - 83276385783224370532x^{14}y^7 + 29177843716011974740x^{18}y^3$

$- 1830508841810758736x^{20} - 13002179457816997720x^{18}y^4 - 75393643320679507380x^{16}y^6 + 17985236293498200376x^{16}y^7$
$+ 45307363279098607960x^{14}y^8 + 28047267927917181682x^{16}y^8 + 5800716553365611858y^{22} + 139952127523063328x^{24} +$
$5131023981508107172x^{18}y^2 - 35896539936568349320x^{16}y^5 - 63846548491920162404x^8y^{13} + 46515909608411062844x^{12}y^9 +$
$6252608086186179204x^4y^{17} - 77933148736218303076x^{10}y^{12} - 477446881676193968x^{20}y + 64333239918010783726x^{12}y^{10} +$
$11559644508156670298x^8y^{14} + 1335986691045269472x^{20}y^3 - 62041765498197635288x^{10}y^{13} - 40748111612807327508x^{18}y^5$
$+ 49390038366279305604x^{14}y^9 + 31792066535649982548x^8y^{15} + 4046845934458101808x^{20}y^4 + 396993236598715596x^{10}y^{14}$
$- 3008073778020546224x^{22}y - 17152489098496057132y^{11}x^{12} - 13581313128386884610y^{20}x^2 - 1679168262561771324y^{21}x^2$
$+ 62793783491268546576x^6y^{16} + 80801271862367669844x^6y^{15} + 13128213294303918818x^{20}y^2 + 405305090835416738x^{18}y^6$
$- 23970171859048284018x^4y^{18} - 4813104358761157528x^4y^{19} + 15617137213100487408y^{21} - 38483852029251610604x^2y^{19}$
$+ 2603745478346208800x^8y^{16} - 17455254401398279120x^{12}y^{12} + 97212959736251058x^2y^{22} + 13957195275810242320x^6y^{17}$
$+ 6339368700580923392$

$S_{14,1} = 20344516069268379348x^2y + 23163722318216555296x^2y^5 - 3297084204990269590x^6y^2 -$
$78961623858736529540x^2y^6 + 8809105578171586208x^8y + 7984948208101804620x^4y^5 - 23786492036175638952x^2y^7$
$+ 47858994265760244132y^2 + 5073457227707723640x^4 + 24049194849380208860y^5 + 4046886773685209840y^7 +$
$347392448636128200x^8 + 30912882480940155858y^4 - 45084867652740671152x^{10} - 6230292883126584900x^6y^4 -$
$52451117031681045056x^8y^3 - 49263151978451466076x^8y^4 - 19065847857409700540x^{12}y + 48721274712522226260x^4y^9$
$+ 18991491299263265752x^{10}y^4 + 1936861038633325766y^{10} + 37071217742709367424y^{11} - 4543059046832726668y^8 -$
$34906099398586749420y^9 - 17540351583798814100y^{12} + 8366260034868039552x^{14} + 76035774308525121966x^4y^2 -$
$23581869741879069782x^2y^4 + 2786080944970707480x^6y - 27639577791017603300x^4y^3 + 52468000953818096472x^4y^4 +$
$10495592323979643168x^6y^3 - 106740061156130040988x^8y^2 + 22135233567181934738x^2y^2 - 21017206889665493034x^4y^6 +$
$4381841560820819330y^{14} + 36819171259493410704x^{16} - 16525630000286832552x^2y^8 - 151201158660684635076x^{10}y^2 +$
$47271092631581810928x^8y^7 + 87530650701970373492x^{12}y^4 + 99663365448508983136x^{10}y + 13242773059510478064x^4y^8 -$
$2512701315116772484x^2y^3 + 72622792314815067624x^8y^5 - 44190372900719340784x^6y^7 - 79792450736402871206x^6y^{12} -$
$97997827018574332044x^{10}y^9 + 79304246585382483532x^8y^6 + 113857624959502154296x^{10}y^6 + 27405395127529293942x^4y^{14}$
$- 46307851608185592828x^2y^9 + 390142544436124024x^2y^{14} + 128932112720337731164x^{10}y^7 + 4034900952405717372x^{14}y^6$
$+ 22673596205660007092x^{12}y^3 - 46543839069869354636x^{14}y^3 - 8516937591826545292x^4y^7 + 26112893792748017348x^6y^5$
$- 20276548949996548440x^{10}y^3 + 27701381401623354340x^6y^9 - 575935073868382788x^8y^8 - 89236535586364364996x^6y^{10} -$
$30103808771143105508x^2y^{11} + 2082266353183423218x^6y^{14} - 36759281549755216656x^{10}y^8 + 31006019308688616240x^{12}y^7$
$+ 35337082712622819036y^{13}x^4 - 29379288221686901712x^2y^{10} - 48165214450781155228x^6y^8 -$
$92850199039998998090x^4y^{10} + 7969759414134904728x^4y^{12} + 69719659845272952560x^8y^9 - 7624001662163779416x^8y^{11} -$

$93865600977375160496x^6y^{13} + 15204022962532907920x^4y^{15} + 3446091792713929308x^2y^{17} + 18817875615277645384x^4y^{11} + 14385626199081938616x^6y^{11} - 17625639989116359392x^8y^{10} + 51299097945450923980x^{12}y^5 + 10881294855059075108x^{12}y^6 - 41113107266741918752x^{12}y^8 - 61178730258097216072x^{10}y^{10} - 45397250235536689356x^8y^{12} + 28088375970482375962x^6y^6 - 110169709590322790860x^{10}y^5 + 35054981549432103680x^{12}y^2 - 14614706348527180664x^{14}y^2 - 1889259853113951892x^{14}y^4 - 117158216660729930632x^{14}\mathrm{y} + 23389281410894254928x^{16}y^2 + 52383861613140825036x^2y^{12} - 53145430135397035716x^2y^{15} + 71530742587176862620x^{14}y^5 - 5813158320141158372x^2y^{16} + 15566816310651150682x^{16}y^4 + 27423204346088091896x^4y^{16} + 15060078382885899832x^2y^{18} + 44940661364488055692y^{13} - 10484873051789107822x^{18} - 8098076198186269088y^{19} - 27038002599219693440y^{18} - 83573203509837972504x^{16}\mathrm{y} - 8794788601323097228x^2y^{13} - 10273173650231531376x^{16}y^3 + 17274383458556066658y^{16} + 3313493588724348580y^{15} + 23041290385253563296x^{18}\mathrm{y} + 16013956449441162324x^{10}y^{11} - 84633200250584387248x^{14}y^7 + 76108442902753195340x^{18}y^3 - 60029058664614445640x^{20} - 36930312771655847472x^{22} + 28758659497087061312x^{18}y^4 - 148031580922290564576x^{16}y^6 - 122558028855324359188x^{16}y^7 + 33285592899218533080x^{14}y^8 + 12391240327318549816x^{16}y^8 + 19218544911754243232y^{22} + 51916180430494734704x^{24} + 85810376822878974672x^{18}y^2 - 91946371226242963624x^{16}y^5 - 27566075039808986368x^8y^{13} - 117481778634298212848x^{12}y^9 - 1394221491615333324x^4y^{17} + 108559876318365199236x^{10}y^{12} - 28320638098324803232x^{20}\mathrm{y} + 37834394249111476314x^{12}y^{10} - 31676156449481016724x^8y^{14} - 50040788928603506132x^{20}y^3 + 27222741576107976680x^{10}y^{13} - 94018890782796624388x^{18}y^5 + 57427485138556991936x^{14}y^9 - 35200144834503132064x^8y^{15} - 120008169977569838540x^{20}y^4 - 67829014629326829684x^{10}y^{14} + 34518564719289223136x^{22}\mathrm{y} + 49791195987019596972y^{11}x^{12} - 31285807061947334126y^{20}x^2 - 152128981239987360y^{21}x^2 + 136020683893321091284x^{14}y^{10} + 31456575644916590348x^6y^{16} + 13235494307882279116x^6y^{15} + 29480271894946777986x^{20}y^2 - 8325314764051084464x^{22}y^2 + 536469762762797416x^{18}y^6 + 33820447224779227426x^6y^{18} + 13264453199633908346x^4y^{18} + 6888946069851414804x^4y^{19} + 9838048186050439368x^4y^{20} + 3990979262862967200y^{21} + 4505939775901390488x^2y^{19} + 9881979060419506788x^8y^{16} - 34492331179325975100x^{12}y^{12} - 9000788023932059904x^2y^{22} - 5778425793666935296x^6y^{17} - 12118826962554301024x^{10}y^{15} - 70843293475804484828x^{12}y^{13} + 16575652526301050568x^{18}y^8 + 50245278218573514216x^{16}y^{10} + 96792653296743997260x^{18}y^9 - 15336580176265744586x^{14}y^{12} - 1209836897993570444x^{16}y^{11} + 60206576737844520536x^8y^{18} - 15635561496199574996x^{14}y^{13} + 35386787173809373948x^{20}y^8 + 30472741304665722894x^{18}y^{10} + 24176271209637910856x^{20}y^5 - 5475464065265617920y^{25} + 31762658852444494848x^{22}y^4 - 39533023953017442952x^{20}y^6 + 43547854254127290108x^{18}y^7 + 431565246326645760x^4y^{23} + 25208483679831436488x^{10}y^{18} + 32707055393628089432x^{22}y^5 + 20159161074358602524x^8y^{17} + 20765511359559563528x^6y^{19} + 134578517057540285360x^{16}y^9 + 81229030489306908240x^{20}y^7 + 21725305288693150924x^{12}y^{14} + 31029171178001513444x^{10}y^{16} -$

$$10269765432217054072x^{12}y^{15} + 10155549212440532000x^{6}y^{20} + 28590629343307268336x^{10}y^{17} + 49604290976905320980x^{16}y^{12} + 40449902089075539938x^{22}y^{6} - 988220462751854092x^{14}y^{11} + 44702673610918105658x^{14}y^{14} - 1556040502172601330 8x^{22}y^{3} + 34318089034806924872x^{26} + 404106469382044800y^{28} + 45635528945595966496x^{24}y + 48583268890469470136x^{24}y^{3} + 8553800015091160032x^{24}y^{2} + 32112229118255083056y^{19}x^{8} + 4673888562647166240y^{23}x^{2} + 1328572930678499520y^{25}x^{2} + 25429844135805551360y^{22}x^{6} + 49198615435488800288y^{16}x^{12} + 1091980805721099912x^{4}y^{22} + 8183585653990155362x^{24}y^{4} + 18523695853519216082$$

$$S_{14,2} = - 41246170357736977220x^{2}y + 14336855389823519392x^{4}y - 49643649885990819988x^{2}y^{5} - 5748784963083378816x^{6}y^{2} + 55200695437884098106x^{2}y^{6} - 17704399696426839788x^{8}y - 13796558800180632480x^{4}y^{5} - 53276857072682145752x^{2}y^{7} + 27716431180138958978y^{2} - 191655527572208448y^{3} + 15345108449591264450x^{4} - 62347349262729871396y^{5} - 10667686181521147040x^{6} + 215561917730587968y^{6} + 18540033268785470 72x^{8} + 331318280212992y^{4} + 13173389481751942060x^{10} - 7439362239981260328 4x^{6}y^{4} + 24202786397329687828x^{8}y^{3} + 12184876730290786084x^{8}y^{4} + 21353001336913322660x^{12}y + 116433490297492368x^{4}y^{9} + 10233621631018355234x^{10}y^{4} + 51448455377122412728y^{10} + 49945871910025776664y^{11} - 4578970080267643616x^{12} + 3506216163640060521 8y^{8} + 59922582344645542920y^{12} - 46842398467536258020x^{14} + 16489299608453167824x^{4}y^{2} + 46467912557196807524x^{2}y^{4} + 25269759028768066900x^{4}y^{4} + 22220323641867521044x^{6}y^{3} - 79042603733583938604x^{8}y^{2} + 142605959780093760x^{2}y^{2} + 31338008811292766624x^{4}y^{6} + 5740715503149602316y^{14} + 19109317603944062034x^{16} - 69864921114521662084y^{17} + 78437867975995460024y^{20} + 94466217598865362876x^{2}y^{8} + 65468533085501247534x^{10}y^{2} - 13765641554768252950 0x^{8}y^{7} - 10288199119452108785 0x^{12}y^{4} + 46556912091480398360x^{10}y + 93220725605490388342x^{4}y^{8} - 22227440881047975888x^{2}y^{3} + 49326829664216823576x^{8}y^{5} + 18553283538558297120x^{6}y^{7} + 53621031670859281084x^{6}y^{12} + 46803704428671888512x^{10}y^{9} + 67029974612321020908x^{8}y^{6} + 73574248985496622212x^{10}y^{6} + 21166426868592620866x^{4}y^{14} - 14311984350734562551 2x^{2}y^{9} - 14097240867202594508x^{2}y^{14} + 20803187390700996420x^{10}y^{7} + 63359854121908244560x^{14}y^{6} + 662255667635326872x^{12}y^{3} - 107901004156822516x^{14}y^{3} - 29980400160024353412x^{4}y^{7} + 40855868560685456956x^{6}y^{5} + 24759698712748447156x^{10}y^{3} + 48953621983255782568x^{6}y^{9} - 10122324359582291596 8x^{8}y^{8} + 4432961490204486812x^{6}y^{10} - 79488354822519554424x^{2}y^{11} + 5708025582001889129 2x^{6}y^{14} - 19074838069250942088x^{10}y^{8} + 28684634533132939964x^{12}y^{7} + 34675397798147550860y^{13}x^{4} + 16823683627434881030x^{2}y^{10} + 70999378760629408516x^{6}y^{8} + 961370719219935924x^{4}y^{10} + 17738259902121826528x^{4}y^{12} - 67588169307449418676x^{8}y^{9} - 83031491299137202084x^{8}y^{11} - 53544084531299763396x^{6}y^{13} + 29782674128741336668x^{4}y^{15} - 10862207366892708814 4x^{2}y^{17} + 58595330790694843744x^{4}y^{11} - 10492531923783104952 4x^{6}y^{11} - 77070896198539970624x^{8}y^{10} + 29243367616707203 80x^{12}y^{5} + 40283200154426750388x^{12}y^{6} + 12422494002231976820x^{12}y^{8} - 333088599431909914 8x^{10}y^{10} - 11079524264264160030 0x^{8}y^{12} - 23319869046567070624x^{6}y^{6} - 10716482401549990972x^{10}y^{5} - 25799977167833293736x^{12}y^{2} - 7336850555734780480x^{14}y^{2} +$$

$62253515267834316504x^{14}y^4 + 28653611689484471744x^{14}y + 62339127316839070940x^{16}y^2 + 78380561760649212540x^2y^{12} - 34990404064228524640x^2y^{15} + 51131837766922883612x^{14}y^5 - 101291870295383948756x^2y^{16} - 27228036230954812486x^{16}y^4 - 79579887587399396872x^4y^{16} - 33955725299367512912x^2y^{18} - 58073719943701602040y^{13} + 46520953242827972944y^{19} + 39493740642119059520y^{18} + 13095217169964927752x^{16}y - 135171524253764243396x^2y^{13} + 6085321535117586200x^{16}y^3 + 48011276531804544y^{16} - 81693166135451007744y^{15} - 10910396482255039936x^{18}y + 31678047370043258656x^{10}y^{11} - 69361907234242001180x^{14}y^7 + 14133124185807330052x^{18}y^3 - 20106510204843548108x^{20} + 9920449031138927828x^{18}y^4 + 911666322881435984x^{16}y^6 + 145846076014339034876x^{16}y^7 - 140753226944438979830x^{14}y^8 - 39835582305418682564x^{16}y^8 + 90227463763136200960y^{22} + 35747715654787187618x^{24} - 48485088315776903582x^{18}y^2 - 24315303624408463104x^{16}y^5 + 19728445408867606072x^8y^{13} + 66581845231342552404x^{12}y^9 - 68326689790199120564x^4y^{17} + 32852410718093747224x^{10}y^{12} - 62552862785486753560x^{20}y + 42049298899161378448x^{12}y^{10} - 119586690592338912874x^8y^{14} - 5928312484835159112x^{20}y^3 - 78249120996942007856x^{10}y^{13} - 9048187013306374608x^{18}y^5 + 5841803523984732220x^{14}y^9 - 8655646259694801712x^8y^{15} + 10988858238830493744x^{20}y^4 + 131126427995509728272x^{10}y^{14} + 38795568913265659712x^{22}y + 31019576314815507336y^{11}x^{12} - 23847400260401496252y^{20}x^2 - 37982330861448118992y^{21}x^2 + 9252165858763848900x^{14}y^{10} + 118387231785940250508x^6y^{16} + 53715242355198541648x^6y^{15} + 53213459722104314044x^{20}y^2 + 298126300800855740x^{22}y^2 + 94245927965761888534x^{18}y^6 + 107233065086509077176x^6y^{18} + 48164881349774297630x^4y^{18} - 101868343601908702624x^4y^{19} - 34931125934063997086x^4y^{20} + 54340168126226786352y^{21} + 54785572132817939832y^{24} - 47956143238246861952x^2y^{19} + 31097409472578604310x^8y^{16} - 117511269509634997880x^{12}y^{12} + 101631955297184186428x^2y^{22} + 83317061561603573484x^6y^{17} - 91314412003628323064x^{10}y^{15} + 4025225540941033984x^{12}y^{13} + 5791458122600383612x^{18}y^8 - 3792797352758903546x^{16}y^{10} - 11150781868921615568x^{18}y^9 + 38662511294653761796x^{14}y^{12} - 11783281762250702584x^{16}y^{11} + 85019310635776898928x^8y^{18} - 37455025991875225932x^{14}y^{13} + 35180526755925570320x^{20}y^8 + 6305913311765124978x^{18}y^{10} + 84349472592295824928x^{20}y^5 - 15054509744864965872y^{25} - 11303846993098080644x^{22}y^4 - 91392951321296294928x^{20}y^6 + 11655211799885855416x^{18}y^7 - 49733445175183538296x^4y^{23} - 46844848345349824280x^{10}y^{18} - 28141858394685289936x^{22}y^5 - 113443539164914272388x^8y^{17} + 41917767995278092596x^6y^{19} + 70126723400680071276x^{16}y^9 + 6274684350989743432x^{20}y^7 - 1505148770411616856x^{12}y^{14} - 38572723985545238332x^{10}y^{16} - 58506956713365869388x^{12}y^{15} + 34544431703611840256x^6y^{20} - 28713608589913106096x^{10}y^{17} + 56011131059035243072x^{16}y^{12} - 19936939700238811668x^{22}y^6 - 50055778884172019220x^{14}y^{11} - 29215840929261861400x^{14}y^{14} - 21132312555067517440x^{22}y^3 + 1739326511603551752y^{28} - 30248888267626370496x^{24}y + 16757133067838655360x^{24}y^3 - 65837938694749973976x^{24}y^2 - 32308588627136565440y^{19}x^8 + 22588799674106708876y^{23}x^2 + 52098766604980179236y^{25}x^2 - 37026318145076677460y^{22}x^6 +$

$5332203571262846672 8y^{16}x^{12} - 6999636641798239682x^4y^{22} + 1097056250381470580 0x^{24}y^4 + 1909227962957314808 6x^2y^{24} + 432791966016275561 68y^{23} - 4550450451628725959 6y^{21}x^4 - 1556074506487858207 2y^{27} + 348032392651377408 98y^{26} - 1133237505660439275 6y^{21}x^6 + 1575828690307895520 2x^{26}y^2 + 3162836948346682123 4y^{20}x^8 + 3052329406967161356 2y^{24}x^4 - 1163307429038107332 0y^{26}x^2$

$S_{14,3} = - 3091291077704190549 6x^4y + 2293401568972585852 8x^2y^5 + 1067621417456256270 60x^6y^2 + 2074274951835735399 2x^2y^6 + 216969390228726015 2x^8y - 5448392552537899209 6x^4y^5 + 6058409096168231579 2x^6 - 4047211387996157494 4x^8 + 5257384769892594737 0x^{10} + 8230352307823285635 2x^6y^4 - 1879509000576704033 6x^8y^3 - 1067200920835054443 04x^8y^4 - 9051121277989104107 2x^{12}y + 1138311239095824093 64x^4y^9 + 4409119192514314971 2x^{10}y^4 + 2050597285433390407 2y^{10} + 424547670145789653 6x^{12} - 1867858083571005384 8y^8 + 1099134953459933340 0y^9 + 1224758717705220220 8y^{12} + 6522576352445558886 0x^{14} - 6093478870334133323 2x^4y^2 + 2560932820826408817 8x^2y^4 + 1507565850164142355 2x^6y - 3633398567538112940 8x^4y^3 - 7169382744870546902 4x^4y^4 + 4621034026879688751 2x^6y^3 - 4731510902013698518 2x^8y^2 - 2123913380122293970 4x^4y^6 + 4510309582765355367 6x^{16} - 7277197821719381700y^{17} - 656466062836360302y^{20} - 2437557586098954234 4x^2y^8 + 3814323738502582270 8x^{10}y^2 + 5488080779571064500 0x^8y^7 + 5594132811376850793 6x^{12}y^4 - 1637534515834919816 8x^{10}y + 1008484745674260240 0x^4y^8 + 8562945838906418414 0x^8y^5 - 5413056678484522706 8x^6y^7 + 1268918545473192887 00x^6y^{12} + 2664235493055467832 8x^{10}y^9 + 3455714172496558028 8x^8y^6 - 9316413749143018461 2x^{10}y^6 - 9759928257073888917 2x^4y^{14} - 1941698946246313004 0x^2y^9 + 6019926627824663784 8x^2y^{14} - 1272017275361796229 64x^{10}y^7 - 8158528527001031795 8x^{14}y^6 + 4557373239447384750 8x^{12}y^3 - 4170806422761103181 2x^{14}y^3 + 2953906152103818048 0x^4y^7 + 4185548983876068544 0x^6y^5 + 3599664665208763070 0x^{10}y^3 - 6851569073573065792 8x^6y^9 - 6414923082156894563 6x^8y^8 - 3003871704987589585 2x^6y^{10} - 3976610253793811898 0x^2y^{11} + 7159604870124807188 4x^6y^{14} + 1295445646703472646 8x^{10}y^8 - 2516726768741498430 8x^{12}y^7 + 8021588341825204072y^{13}x^4 - 3558258292220188374 4x^2y^{10} - 3121996586358706327 2x^6y^8 - 2724390384816818370x^4y^{10} - 9470086926519753668x^4y^{12} + 9795413291173417967 2x^8y^9 + 1582194437789875300 8x^8y^{11} + 6271083183501028596x^6y^{13} - 3106288601371272478 4x^4y^{15} - 3674384917356649164 4x^2y^{17} - 914855398061178804 8x^4y^{11} - 4288208238060985940 0x^6y^{11} + 7250769551333338840x^8y^{10} - 4111232478770634562 8x^{12}y^5 - 1135810744911060624 60x^{12}y^6 - 930842169643448692 12x^{12}y^8 + 1181206135867511789 58x^{10}y^{10} - 3178746470339863054 8x^8y^{12} - 9144839136127640356x^6y^6 - 3584678889634463381 2x^{10}y^5 - 1484369070713814265 6x^{12}y^2 + 9370156230403119514 4x^{14}y^2 - 1452599184346157825 36x^{14}y^4 - 2348749574709753409 6x^{14}y + 3096676750798599654 2x^{16}y^2 - 4380209309004611838 4x^2y^{12} + 7727739224019798702 4x^2y^{15} - 5647808370292324938 0x^{14}y^5 + 6228919468382130770 4x^2y^{16} + 5184158509417102442 0x^{16}y^4 - 5477179526350205216 8x^4y^{16} + 6006945318888922758x^2y^{18} + 1748343373889477814 6x^{18} + 7989151975748736300y^{19} - 1143555711233693298 6y^{18} + 8572582884527943013 6x^{16}y + 2982097803836905048 4x^2y^{13} + 8994725493767197270 8x^{16}y^3 +$

$6183395739647613362y^{16} - 10025834561447070846 4x^{18}y + 2818362023880756910 8x^{10}y^{11} + 7714514123753615285 2x^{14}y^{7}$
$- 3474010587090867242 0x^{18}y^{3} + 5690505428712018711 8x^{20} + 901823023551761994x^{22} - 3259146983340446439 2x^{18}y^{4} -$
$8529352685840229662 0x^{16}y^{6} - 6022006742633671507 6x^{16}y^{7} - 8492107916543958608 0x^{14}y^{8} - 9611658214294068623 2x^{16}y^{8}$
$+ 890225854301960625 6y^{22} + 2529254928454230522 8x^{24} - 1892116248434717179 0x^{18}y^{2} + 1543513512263566739 12x^{16}y^{5} -$
$7416264497329523986 4x^{8}y^{13} + 1936850500270712723 2x^{12}y^{9} - 1603164761926393442 4x^{4}y^{17} + 6482904618694714734 8x^{10}y^{12}$
$+ 6428952775949414190 0x^{20}y - 1037324692412787468 40x^{12}y^{10} - 1041381783511972117 84x^{8}y^{14} +$
$6971154117918154571 2x^{20}y^{3} + 7515028497926289798 8x^{10}y^{13} - 3545747453377555292 0x^{18}y^{5} - 5967267994919758553 2x^{14}y^{9}$
$+ 4809595870171388376 0x^{8}y^{15} - 2537097208784615754 8x^{20}y^{4} + 1610508613394964407 82x^{10}y^{14} -$
$7241216393961602218 8x^{22}y + 1989339023601505646 4y^{11}x^{12} - 1876558391108456748 8y^{20}x^{2} - 3029175521419368154 8y^{21}x^{2}$
$+ 4642335424487534829 2x^{14}y^{10} + 7637508464579120226 0x^{6}y^{16} + 800086647807268933 2x^{6}y^{15} +$
$1363199284206612958 60x^{20}y^{2} - 2361607560330949706 8x^{22}y^{2} - 1210064453742513982 8x^{18}y^{6} + 817354354429929049 4x^{6}y^{18}$
$+ 2620690093268231845 0x^{4}y^{18} + 3846098895121045269 6x^{4}y^{19} - 3018583075691702766 8x^{4}y^{20} + 4771674867063013200y^{21}$
$+ 2658522674169218592y^{24} - 2610497765867001920 4x^{2}y^{19} + 2445881245420467538 8x^{8}y^{16} + 6308126389038407659 6x^{12}y^{12}$
$- 3237373831605414740 4x^{2}y^{22} - 2505959238841062024 4x^{6}y^{17} + 14105897338864074248 - 5750981739054221766 4x^{10}y^{15}$
$+ 3835968893651371941 6x^{12}y^{13} + 1067230387906922012 84x^{18}y^{8} + 1785105784878581692x^{16}y^{10}$
$+ 2566773978190397905 6x^{18}y^{9} + 5016418968594709824 2x^{14}y^{12} - 1161588316325280558 00x^{16}y^{11}$
$- 2659380090297152934 0x^{8}y^{18} - 7524150969709941623 6x^{14}y^{13} - 3381152032208350224 8x^{20}y^{8} +$
$1647011441394549354 8x^{18}y^{10} - 4182073529212314408 0x^{20}y^{5} - 1457727455798722039 2x^{22}y^{4} - 1420126211419203010 4x^{20}y^{6}$
$+ 4337430822382308828 8x^{18}y^{7} + 726789551159140830 0x^{4}y^{23} + 2547899769648280300 0x^{10}y^{18} +$
$6171855413174043129 6x^{22}y^{5} + 1085575241682699503 6x^{8}y^{17} - 3327273581271551746 8x^{6}y^{19} + 9131834594131233673 2x^{16}y^{9}$
$+ 7861625214343928911 2x^{20}y^{7} + 3607949609049591387 2x^{12}y^{14} - 6717015513995882654 8x^{10}y^{16}$
$- 1147811170458893170 4x^{12}y^{15} - 3406713351480549109 0x^{6}y^{20} + 3254102509772238952x^{10}y^{17} +$
$1315548336391864264 2x^{16}y^{12} - 5112877909097869596 4x^{22}y^{6} + 3970614361106734809 6x^{14}y^{11} -$
$1584468094729786711 6x^{14}y^{14} - 6671710848695396972 0x^{22}y^{3} + 827608743656469320x^{26} - 3954983042176066788x^{24}y +$
$4230606737032989984x^{24}y^{3} - 4663263551237098616 6x^{24}y^{2} + 1864301445854784683 6y^{19}x^{8} + 3263688136286307618 0y^{25}x^{2}$
$- 4983351546613311152y^{22}x^{6} - 1094083821308895875 6y^{16}x^{12} + 2252779721396212288 8x^{4}y^{22} +$
$1250903798549480043 2x^{24}y^{4} + 2345632027025893216 2x^{2}y^{24} - 1317554007798261066 0y^{21}x^{4} - 2418312490066929687 2y^{21}x^{6}$
$+ 2854210809069900789 0x^{26}y^{2} + 2351899103955357893 8y^{20}x^{8} + 2323984640128581152y^{24}x^{4} + 9367363259735390450y^{26}x^{2}$
$- 3189477857292959526 0x^{26}y + 1251531003507477245 0x^{28}$

Bibliography

[1] John Abbott. Quadratic interval refinement for real roots. In *International Symposium on Symbolic and Algebraic Computation (ISSAC 2006), poster presentation*, 2006.

[2] Niels Henrik Abel. Mémoire sur les équations algébriques où on démontre l'impossibilité de la résolution de l'équation générale du cinquième dégré. Christiania, 1824.

[3] Lionel Alberti, Bernard Mourrain, and Julien Wintz. Topology and arrangement computation of semi-algebraic planar curves. *Computer Aided Geometric Design*, 25(8):631–651, 2008.

[4] Götz Alefeld and Jürgen Herzberger. *Introduction to Interval Computations*. Academic Press, New York, 1983.

[5] Andrei Alexandrescu. *Modern C++ design: generic programming and design patterns applied*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.

[6] Arthur Appel. Some techniques for shading machine renderings of solids. In *AFIPS Conference Proceedings: 1968 Spring Joint Computer Conference, Atlantic City, NJ, USA, 30 April - 2 May 1968*, volume 32, pages 37–45. Thomson Book Company, Washington DC, 1968.

[7] Dennis S. Arnon, George Edwin Collins, and Scott McCallum. Cylindrical algebraic decomposition II: An adjacency algorithm for the plane. *SIAM Journal on Computing*, 13(4):878–889, 1984.

[8] Dennis S. Arnon and Scott McCallum. A polynomial time algorithm for the topological type of a real algebraic curve. *Journal of Symbolic Computation*, 5:213–236, 1988.

[9] Philippe Aubry and Marc Moreno Maza. Triangular sets for solving polynomial systems: a comparative implementation of four methods. *Journal of Symbolic Computation*, 28(1-2):125–154, 1999.

[10] Matthew H. Austern. *Generic programming and the STL: using and extending the C++ Standard Template Library*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1998.

[11] Axel algebraic geometric modeler. `http://axel.inria.fr`.

[12] Magali Bardet, Jean-Charles Faugére, and Bruno Salvy. On the complexity of Gröbner basis computation of semi-regular overdetermined algebraic equations. In *International Conference on Polynomial System Solving*, pages 71–74, November 2004. Proceedings of a conference held in Paris, France in honor of Daniel Lazard.

[13] Saugata Basu, Richard Pollack, and Marie-Françoise Roy. *Algorithms in Real Algebraic Geometry*, volume 10 of *Algorithms and Computation in Mathematics*. Springer-Verlag, 2nd edition, 2006.

[14] Riccardo Benedetti and Jean-Jacques Risler. *Real Algebraic and Semi-algebraic Sets, Actualités Mathématiques*. Hermann, Paris, 1990.

[15] Jon Louis Bentley and Jerome H. Friedman. Data structures for range searching. *ACM Computing Surveys*, 11(4):397–409, 1979.

[16] Jon Louis Bentley and Thomas A. Ottmann. Algorithms for reporting and counting geometric intersections. *Computers, IEEE Transactions on*, C-28(9):643–647, September 1979.

[17] Eric Berberich, Arno Eigenwillig, Michael Hemmer, Susan Hert, Lutz Kettner, Kurt Mehlhorn, Joachim Reichel, Susanne Schmitt, Elmar Schömer, and Nicola Wolpert. EXACUS: Efficient and exact algorithms for curves and surfaces. In *ESA '05: Proceedings of the 13th Annual European Symposium on Algorithms*, volume 1669 of *LNCS*, pages 155–166. Springer, 2005.

[18] Eric Berberich, Michael Hemmer, Menelaos Karavelas, Sylvain Pion, Monique Teillaud, and Elias Tsigaridas. Prototype implementation of the algebraic kernel. Technical Report ACS-TR-121202-01, ACS European Project, 2006.

[19] Eric Berberich, Michael Hemmer, Menelaos I. Karavelas, and Monique Teillaud. Revision of the interface specification of algebaic kernel. Technical Report ACS-TR-243301-01, ACS European Project, 2007.

[20] Eric Berberich, Michael Hemmer, Sylvain Lazard, Luis Peñaranda, and Monique Teillaud. Algebraic kernel. In *CGAL User and Reference Manual*. CGAL Editorial Board, 3.6 edition, 2010. `http://www.cgal.org/Manual/3.6/doc_html/cgal_manual/Algebraic_kernel_d/Chapter_main.html`.

[21] Joan S. Birman. *Braids, Links, and Mapping Class Groups*, volume 82 of *Annals of Mathematical Studies*. Princeton University Press, Princeton, NJ, 1975.

[22] Jean-Daniel Boissonnat and Monique Teillaud, editors. *Effective Computational Geometry for Curves and Surfaces*. Springer-Verlag, Mathematics and Visualization, 2006.

[23] Boost: free peer-reviewed portable C++ source libraries. `http://www.boost.org`.

[24] Jorge Luis Borges. *El libro de arena*. Autores argentinos. Ultramar, Madrid, 1975.

[25] Yacine Bouzidi. Un algorithme pour la résolution de systèmes bivariés à coefficients rationnels - Application au calcul de topologie de courbes planes. Master's thesis, Université Pierre et Marie Curie (Paris 6), 2010.

[26] Frederick P. Brooks Jr. No silver bullet - essence and accidents of software engineering (invited paper). In *Proceedings of the IFIP Tenth World Computing Conference*, pages 1069–1076, 1986.

[27] Christopher W. Brown. Cad2d: minor modification of QEPCAD for quantifier-free formulas in 2 variables. `http://www.usna.edu/Users/cs/qepcad/B/user/cad2d.html`.

[28] Christopher W. Brown. Improved projection for cylindrical algebraic decomposition. *Journal of Symbolic Computation*, 32(5):447–465, 2001.

[29] Christopher W. Brown. Constructing cylindrical algebraic decomposition of the plane quickly, 2002. Manuscript.

[30] Bruno Buchberger. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal.* PhD thesis, University of Innsbruck, 1965.

[31] Bruno Buchberger. *Gröbner bases: an algorithmic method in polynomial ideal theory.* Recent trends in multidimensional systems theory. Reider ed. Bose, 1985.

[32] Bruno Buchberger. An algorithmic criterion for the solvability of algebraic systems of equations. *Proceedings of the International Conference "33 Years of Gröbner Bases"*, 251:535–545, 1998.

[33] Ferdinand François Désiré Budan de Boislaurent. *Nouvelle méthode pour la résolution des équations numériques d'un degré quelconque.* Bachelier. Dondey-Dupré, Paris, 2nd edition, 1822.

[34] Christoph Burnikel, Rudolf Fleischer, Kurt Mehlhorn, and Stefan Schirra. A strong and easily computable separation bound for arithmetic expressions involving radicals. *Algorithmica*, 27:87–99, 2000.

[35] Christoph Burnikel, Stefan Funke, Kurt Mehlhorn, Stefan Schirra, and Susanne Schmitt. A separation bound for real algebraic expressions. In *Proceedings of the 9th Annual European Symposium on Algorithms*, volume 2161 of *LNCS*, pages 254–265, London, UK, 2001. Springer Verlag.

[36] Michael Burr, Sungwoo Choi, Ben Galehouse, and Chee Keng Yap. Complete subdivision algorithms, II: Isotopic meshing of singular algebraic curves. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation (ISSAC 2008)*, 2008.

[37] John F. Canny. *The Complexity of Robot Motion Planning*. MIT, 1988.

[38] CGAL: Computational geometry algorithms library. `http://www.cgal.org`.

[39] Gregory J. Chaitin. A theory of program size formally identical to information theory. *Journal of the ACM*, 22(3):329–340, 1975.

[40] Jinsan Cheng, Sylvain Lazard, Luis Peñaranda, Marc Pouget, Fabrice Rouillier, and Elias Tsigaridas. On the topology of real algebraic plane curves. *Mathematics in Computer Science (special issue on Computational Geometry and Computer Aided Geometric Design)*, 4(1):113–137, 2010.

[41] Jinsan Cheng, Sylvain Lazard, Luis Peñaranda, Marc Pouget, Fabrice Rouillier, and Elias P. Tsigaridas. On the topology of planar algebraic curves. In *Proceedings of the 24th European Workshop on Computational Geometry*, pages 213–216, March 2008.

[42] Jinsan Cheng, Sylvain Lazard, Luis Peñaranda, Marc Pouget, Fabrice Rouillier, and Elias P. Tsigaridas. On the topology of planar algebraic curves. In *Proceedings*

of the 25th Symposium on Computational Geometry, Aarhus, Denmark, June 8-10, 2009, pages 361–370. ACM, 2009.

[43] George Edwin Collins and Alkiviadis G. Akritas. Polynomial real root isolation using Descarte's rule of signs. In *SYMSAC '76: Proceedings of the third symposium on Symbolic and algebraic computation*, pages 272–275, New York, NY, USA, 1976. ACM.

[44] George Edwin Collins, Jeremy R. Johnson, and Werner Krandick. Interval arithmetic in cylindrical algebraic decomposition. *Journal of Symbolic Computation*, 34(2):145–157, 2002.

[45] The Core library. `http://cs.nyu.edu/exact`.

[46] Julio Cortázar. Instrucciones para subir una escalera. In *Historias de Cronopios y de Famas*, Buenos Aires, 1962. Minotauro.

[47] Michel Coste and Marie-Françoise Roy. Thom's lemma, the coding of real algebraic numbers and the computation of the topology of semi-algebraic sets. *Journal of Symbolic Computation*, 5(1/2):121–129, 1988.

[48] David A. Cox, John B. Little, and Don O'Shea. *Ideals, Varieties, and Algorithms: an Introduction to Computational Algebraic Geometry and Commutative Algebra.* Undergraduate Texts in Mathematics. Springer-Verlag, New York, 3rd edition, July 2005.

[49] David A. Cox, John B. Little, and Don O'Shea. *Using Algebraic Geometry.* Number 185 in Graduate Texts in Mathematics. Springer, New York, 2nd edition, 2005.

[50] James Harold Davenport. Cylindrical algebraic decomposition. Technical Report 88–10, School of Mathematical Sciences, University of Bath, England, 1988.

[51] Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars. *Computational Geometry: Algorithms and Applications*. Springer, 3rd edition, April 2008. http://www.cs.uu.nl/geobook.

[52] Pedro M.M. de Castro, Frédéric Cazals, Sébastien Loriot, and Monique Teillaud. Design of the cgal 3D spherical kernel and application to arrangements of circles on a sphere. *Computational Geometry: Theory and Applications*, 42(6-7):536–550, 2009.

[53] Wolfram Decker, Gert-Martin Greuel, Gerhard Pfister, and Hans Schönemann. Singular: a computer algebra system for polynomial computations. http://www.singular.uni-kl.de.

[54] Boris Nikolaevich Delaunay. Sur la sphère vide. *Bulletin of Academy of Sciences of the USSR*, 6:793–800, 1934.

[55] René Descartes. *La géométrie*. Jean Maire, Leyde, 1637.

[56] Dimitrios I. Diochnos, Ioannis Z. Emiris, and Elias P. Tsigaridas. On the complexity of real solving bivariate systems. In C. W. Brown, editor, *Proceedings of the International Symposium on Symbolic and Algebraic Computation*, pages 127–134, Waterloo, Canada, 2007.

[57] Zilin Du, Vikram Sharma, and Chee Keng Yap. Amortized bound for root isolation via Sturm sequences. In D. Wang and L. Zhi, editors, *International Workshop on Symbolic Numeric Computing*, pages 113–129, Basel, 2007. Birkhäuser.

[58] Alan Edelman and Eric Kostlan. How may zeros of a random polynomial are real? *Bulletin of the American Mathematical Society*, 32(1):1–37, January 1995.

[59] Herbert Edelsbrunner. *Algorithms in combinatorial geometry*. Springer-Verlag, New York, NY, USA, 1987.

[60] Arno Eigenwillig and Michael Kerber. Exact and efficient 2D-arrangements of arbitrary algebraic curves. In *Proceedings of the 19th Annual Symposium on Discrete Algorithms (SODA08)*, pages 122–131, San Francisco, USA, January 2008. ACM-SIAM, ACM/SIAM.

[61] Arno Eigenwillig, Michael Kerber, and Nicola Wolpert. Fast and exact geometric analysis of real algebraic plane curves. In C. W. Brown, editor, *Proceedings of the International Symposium on Symbolic and Algebraic Computation*, pages 151–158, Waterloo, Canada, 2007. ACM.

[62] Arno Eigenwillig, Lutz Kettner, Werner Krandick, Kurt Mehlhorn, Susanne Schmitt, and Nicola Wolpert. A Descartes algorithm for polynomials with bit-stream coefficients. In V. Ganzha, E. Mayr, and E. Vorozhtsov, editors, *CASC*, volume 3718 of *LNCS*, pages 138–149. Springer, 2005.

[63] Arno Eigenwillig, Vikram Sharma, and Chee Keng Yap. Almost tight recursion tree bounds for the Descartes method. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation*, pages 71–78, New York, NY, USA, 2006. ACM.

[64] Pavel Emeliyanenko. Visualization of points and segments of real algebraic plane curves. Master's thesis, Universität des Saarlandes, February 2007.

[65] Pavel Emeliyanenko, Eric Berberich, and Michael Sagraloff. Visualizing arcs of implicit algebraic curves, exactly and fast. In G. Bebis, R. Boyle, B. Parvin, D. Koracin, Y. Kuno, J. Wang, R. Pajarola, P. Lindstrom, A. Hinkenjann, M. Encarnação, C. Silva, and D. Coming, editors, *Advances in Visual Computing : 5th International Symposium, ISVC 2009*, volume 5875 of *Lecture Notes in Computer Science*, pages 608–619, Las Vegas, USA, 2009. Springer.

[66] Pavel Emeliyanenko and Michael Kerber. An implementation of the 2D algebraic kernel. Technical Report ACS-TR-363602-01, ACS European Project, Groningen, The Netherlands, May 2008.

[67] Ioannis Z. Emiris, André Galligo, and Elias P. Tsigaridas. Random polynomials and expected complexity of bisection methods for real solving. In S. Watt, editor, *Proceedings of the International Symposium on Symbolic and Algebraic Computation (ISSAC 2010)*, pages 235–242, Munich, Germany, July 2010. ACM.

[68] Ioannis Z. Emiris, Michael Hemmer, Menelaos Karavelas, Sebastian Limbach, Bernard Mourrain, Elias P. Tsigaridas, and Zafeirakis Zafeirakopoulos. Cross-benchmarks for univariate algebraic kernels. Technical Report ACS-TR-363602-02, ACS European Project, 2008.

[69] Ioannis Z. Emiris, Athanasios Kakargias, Sylvain Pion, Monique Teillaud, and Elias P. Tsigaridas. Towards and open curved kernel. In J. Snoeyink and J-D. Boissonnat, editors, *Proceedings of the 20th Symposium on Computational Geometry, New York, USA, June 8-11, 2004*, pages 438–446, New York, USA, 2004. ACM.

[70] Ioannis Z. Emiris, Bernard Mourrain, and Elias P. Tsigaridas. Real algebraic numbers: Complexity analysis and experimentation. In P. Hertling, C. Hoffmann, W. Luther, and N. Revol, editors, *Reliable Implementations of Real Number Algorithms: Theory and Practice*, volume 5045 of *LNCS*, pages 57–82. Springer Verlag, 2008.

[71] Pál Erdős and Pál Turán. On the distribution of roots of polynomials. *The Annals of Mathematics, Second Series*, 51(1):105–119, January 1950.

[72] Hazel Everett, Daniel Lazard, Sylvain Lazard, and Mohab Safey El Din. The Voronoi diagram of three lines. *Journal of Discrete and Computational Geometry*, 42:94–130, 2009.

[73] Jean-Charles Faugère. FGb: a software for computing Gröbner bases. `http://www-salsa.lip6.fr/SALSAWEB/Software.html`.

[74] Jean-Charles Faugère. A new efficient algorithm for computing Gröbner bases (F_4). *Journal of Pure and Applied Algebra*, 139(1-3):61–88, June 1999.

[75] Huichun Feng. *Decomposition and Computation of the Topology of Plane Real Algebraic Curves*. PhD thesis, The Royal Institute of Technology, Stockholm, 1992.

[76] Efi Fogel, Dan Halperin, Lutz Kettner, Monique Teillaud, Ron Wein, and Nicola Wolpert. Arrangements. In Jean-Daniel Boissonnat and Monique Teillaud, editors, *Effective Computational Geometry for Curves and Surfaces*, pages 1–66. Springer-Verlag, Mathematics and Visualization, 2006.

[77] Jean Baptiste Joseph Fourier. Sur l'usage du théorème de Descartes dans la recherche des limites des racines. *Bulletin des sciences par la Société philomathique de Paris*, pages 156–165, October 1820.

[78] Jean Baptiste Joseph Fourier. Sur l'usage du théorème de Descartes dans la recherche des limites des racines. *Bulletin des sciences par la Société philomathique de Paris*, pages 181–187, December 1820.

[79] Erich Gamma, Richard Helm, Ralph E. Johnson, and John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, Reading, MA, 1995.

[80] Joachim Von Zur Gathen and Jürgen Gerhard. Fast algorithms for taylor shifts and certain difference equations. In *International Symposium on Symbolic and Algebraic Computation (ISSAC 1997)*, pages 40–47. ACM, 1997.

[81] Joachim Von Zur Gathen and Jürgen Gerhard. *Modern Computer Algebra*. Cambridge University Press, New York, NY, USA, 2nd edition, 2003.

[82] Marc Giusti, Grégoire Lecerf, and Bruno Salvy. A Gröbner free alternative for solving polynomial systems. *Journal of Complexity*, 17(1):154–211, 2001.

[83] GMP: the GNU multiple precision arithmetic library. `http://gmplib.org`.

[84] Laureano González Vega and M'hammed El Kahoui. An improved upper complexity bound for the topology computation of a real algebraic plane curve. *Journal of Complexity*, 12(4):527–544, 1996.

[85] Laureano González Vega, Henri Lombardi, Tomás Recio, and Marie-Françoise Roy. Sturm-Habicht sequence. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation*, pages 136–146, 1989.

[86] Laureano González Vega and Ioana Necula. Efficient topology determination of implicitly defined algebraic plane curves. *Computer Aided Geometric Design*, 19(9), 2002.

[87] Ronald Lewis Graham. An efficient algorith for determining the convex hull of a finite planar set. *Information Processing Letters*, 1(4):132–133, 1972.

[88] Gert-Martin Greuel, Gerhard Pfister, and Hans Schönemann. SINGULAR 3.0 — a computer algebra system for polynomial computations. In M. Kerber and M. Kohlhase, editors, *Symbolic computation and automated reasoning, The Calculemus-2000 Symposium*, pages 227–233. A. K. Peters, Ltd., Natick, MA, USA, 2001.

[89] Michael Hemmer. Algebraic foundations. In *CGAL User and Reference Manual*. CGAL Editorial Board, 3.6 edition, 2010. http://www.cgal.org/Manual/3.6/doc_html/cgal_manual/Algebraic_foundations/Chapter_main.html.

[90] Michael Hemmer. Polynomial. In *CGAL User and Reference Manual*. CGAL Editorial Board, 3.6 edition, 2010. http://www.cgal.org/Manual/3.6/doc_html/cgal_manual/Polynomial/Chapter_main.html.

[91] Michael Hemmer, Susan Hert, Sylvain Pion, and Stefan Schirra. Number types. In *CGAL User and Reference Manual*. CGAL Editorial Board, 3.6 edition, 2010. http://www.cgal.org/Manual/3.6/doc_html/cgal_manual/NumberTypeSupport/Chapter_main.html.

[92] Michael Hemmer and Dominik Hülse. Generic implementation of a modular gcd over algebraic extension fields. In *Proceedings of the 25th European Workshop on Computational Geometry*, pages 321–324, Brussels, Belgium, 2009. Université Libre de Bruxelles.

[93] Michael Hemmer and Sebastian Limbach. Benchmarks on a generic univariate algebraic kernel. Technical Report ACS-TR-243306-03, ACS European Project, 2006.

[94] Michael Hemmer, Elias P. Tsigaridas, Zafeirakis Zafeirakopoulos, Ioannis Z. Emiris, Menelaos I. Karavelas, and Bernard Mourrain. Experimental evaluation and cross-benchmarking of univariate real solvers. In *SNC '09: Proceedings of the 2009 conference on Symbolic numeric computation*, pages 45–54, New York, NY, USA, 2009. ACM.

[95] Charles Antony Richard Hoare. The 1980 Turing award lecture: The emperor's old clothes. *Communications of the ACM*, 24(2):75–83, 1981.

[96] Hoon Hong. QEPCAD: Quantifier elimination by partial cylindrical algebraic decomposition. `http://www.usna.edu/Users/cs/qepcad`. With contributions of Christopher W. Brown, George E. Collins, Mark J. Encarnación, Jeremy R. Johnson, Werner Krandick, Richard Liska, Scott McCallum, Nicolas Robidoux and Stanly Steinberg.

[97] Hoon Hong. An efficient method for analyzing the topology of plane real algebraic curves. *Mathematics and Computers in Simulation*, 42(4–6):571–582, 1996.

[98] Gerhard Jaeschke. On strong pseudoprimes to several bases. *Mathematics of Computation*, 61(204):915–926, October 1993.

[99] R. A. Jarvis. On the identification of the convex hull of a finite set of points in the plane. *Information Processing Letters*, 2(1):18–21, 1973.

[100] Mark Kac. On the average number of real roots of a random algebraic equation. *Bulletin of American Mathematical Society*, 49:314–320 and 938, 1943.

[101] L. V. Kantorovich. On Newton's method for functional equations. *Dokl. Akad. Nauk. SSSR*, 59:1237–1240, 1948.

[102] Vijay Karamcheti, Chen Li, Igor Pechtchanski, and Chee Keng Yap. A core library for robust numeric and geometric computation. In *SCG '99: Proceedings of the fifteenth annual symposium on Computational geometry*, pages 351–359, New York, NY, USA, 1999. ACM.

[103] Michael Kerber. Analysis of real algebraic plane curves. Master's thesis, MPII, 2006.

[104] Michael Kerber. *Geometric Algorithms for Algebraic Curves and Surfaces.* PhD thesis, Saarland University, 2009.

[105] Lutz Kettner, Kurt Mehlhorn, Sylvain Pion, Stefan Schirra, and Chee Keng Yap. Classroom examples of robustness problems in geometric computations. *Computational Geometry*, 40:61–78, 2008.

[106] John Keyser, Tim Culver, Dinesh Manocha, and Shankar Krishnan. Efficient and exact manipulation of algebraic points and curves. *Computer-Aided Design*, 32(11):649–662, 2000.

[107] John Keyser, Koji Ouchi, and J. Maurice Rojas. The exact rational univariate representation for detecting degeneracies. In *DIMACS Series in Discrete Mathematics and Theoretical Computer Science.* AMS Press, 2005.

[108] Donald Ervin Knuth. *The Art of Computer Programming, Volume II: Seminumerical Algorithms.* Addison-Wesley, Reading, Massachusetts, 1969.

[109] Oliver Labs. A list of challenges for real algebraic plane curve visualization software. In I. Emiris, F. Sottile, and T. Theobald, editors, *Nonlinear Computational*

Geometry, volume 151 of *The IMA Volumes in Mathematics and its Applications*, pages 137–164. Springer, New York, 2009.

[110] Steven M. LaValle. *Planning Algorithms.* Cambridge University Press, Cambridge, U.K., 2006. Available at `http://planning.cs.uiuc.edu`.

[111] Daniel Lazard. Gröbner bases, Gaussian elimination, and resolution of systems of algebraic equations. In *EUROCAL '83 European Computer Algebra Conference*, volume 162 of *LNCS*, pages 146–156. Springer, 1983.

[112] Sylvain Lazard, Luis Peñaranda, and Elias P. Tsigaridas. Univariate algebraic kernel and application to arrangements. In *Experimental Algorithms, Proceedings of the 8th International Symposium, SEA 2009, Dortmund, Germany, June 4-6, 2009*, volume 5526 of *Lecture Notes in Computer Science*, pages 209–220. Springer, 2009.

[113] Der-Tsai Lee and Franco P. Preparata. Computational geometry - a survey. *IEEE Transactions on Computers*, 33(12):1072–1101, 1984.

[114] Chen Li, Sylvain Pion, and Chee Keng Yap. Recent progress in exact geometric computation. *Journal of Logic and Algebraic Programming*, 64(1):85–111, 2004. Special issue on "Practical Development of Exact Real Number Computation".

[115] Thomas Lickteig and Marie-Françoise Roy. Sylvester-Habicht sequences and fast Cauchy index computation. *Journal of Symbolic Computation*, 31(3):315–341, 2001.

[116] Rüdiger G. K. Loos. Computing in algebraic extensions. *Computing Supplementum*, 4:173–187, 1982.

[117] Dinesh Manocha. Solving polynomial systems for curve, surface and solid modeling. In *SMA '93: Proceedings on the second symposium on Solid modeling and applications*, pages 169–178, New York, NY, USA, 1993. ACM.

[118] Maple computer algebra system. `http://www.maplesoft.com`.

[119] Maplesoft. *Maple 14 User Manual*, 2010. `http://www.maplesoft.com/documentation_center`.

[120] Scott McCallum and George Edwin Collins. Local box adjacency algorithms for cylindrical algebraic decompositions. *Journal of Symbolic Computation*, 33(3):321–342, 2002.

[121] Kurt Mehlhorn. *Data Structures and Algorithms 1: Sorting and Searching*, volume 1 of *Monographs in Theoretical Computer Science. An EATCS Series.* Springer, 1984.

[122] Kurt Mehlhorn and Chee Keng Yap. Robust geometric computation. Book in preparation, `http://cs.nyu.edu/yap/book/egc`.

[123] Maurice Mignotte. *Mathématiques pour le calcul formel.* Presses universitaires de France, Paris, April 1989.

[124] Victor J. Milenkovic and Lee R. Nackman. Finding compact coordinate representations for polygons and polyhedra. *IBM Journal of Research and Development*, 34(35):753–769, September 1990.

[125] Bernard Mourrain, Jean-Pascal Pavone, Philippe Trébuchet, Elias P. Tsigaridas, and Julien Wintz. Synaps, a library for dedicated applications in symbolic numeric computations. In M. Stillman, N. Takayama, and J. Verschelde, editors, *IMA Volumes in Mathematics and its Applications*, pages 81–110. Springer, New York, 2007.

[126] Bernard Mourrain, Sylvain Pion, Susanne Schmitt, Jean-Pierre Técourt, Elias Tsigaridas, and Nicola Wolpert. Algebraic issues in computational geometry. In Jean-Daniel Boissonnat and Monique Teillaud, editors, *Effective Computational Geometry for Curves and Surfaces*, pages 117–155. Springer-Verlag, Mathematics and Visualization, 2006.

[127] Bernard Mourrain and Philippe Trébuchet. Generalized normal forms and polynomial system solving. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation*, pages 253–260, 2005.

[128] MPFI multiple precision interval arithmetic library. https://gforge.inria.fr/projects/mpfi.

[129] MPFR: a library for multiple-precision floating-point computations. http://mpfr.org.

[130] IEEE Task P754. *ANSI/IEEE 754-1985, Standard for Binary Floating-Point Arithmetic.* IEEE, New York, August 12 1985.

[131] Victor Y. Pan. Solving a polynomial equation: Some history and recent progress. *SIAM Review*, 39(2):187–220, 1997.

[132] Christos Harilaos Papadimitriou. *Computational complexity.* Addison-Wesley, Reading, Massachusetts, 1994.

[133] Sylvain Pion. *De la géométrie algorithmique au calcul géométrique.* PhD thesis, Université de Nice Sophia-Antipolis, November 1999.

[134] Sylvain Pion and Monique Teillaud. Towards a CGAL-like kernel for curves. Technical Report ECG-TR-30220601, MPI, INRIA, 2003.

[135] Simon Plantinga and Gert Vegter. Isotopic approximation of implicit curves and surfaces. In *SGP '04: Proceedings of the 2004 Eurographics/SIGGRAPH symposium on Geometry processing*, pages 245–254, New York, NY, USA, 2004. ACM.

[136] Franco P. Preparata and Michael Ian Shamos. *Computational Geometry - An Introduction.* Springer, 1985.

[137] Michael O. Rabin. Probabilistic algorithm for testing primality. *Journal of Number Theory*, 12(1):128–138, 1980.

[138] Richard E. Zippel. *Effective Polynomial Computation.* Kluwer Academic, Norwell, Massachusetts, 1993.

[139] Fabrice Rouillier. RS: a software for real solving of algebraic systems. `http://www-salsa.lip6.fr/SALSAWEB/Software.html`.

[140] Fabrice Rouillier. Solving zero-dimensional systems through the rational univariate representation. *Applicable Algebra in Engineering, Communication and Computing*, 9(5):433–461, 1999.

[141] Fabrice Rouillier and Paul Zimmermann. Efficient isolation of a polynomial real roots. *Journal of Computational and Applied Mathematics*, 162(1):33–50, 2004.

[142] Ernesto Sabato. *Uno y el universo.* Sudamericana, 1945.

[143] Takis Sakkalis. The topological configuration of a real algebraic curve. *Bulletin of the Australian Mathematical Society*, 43:37–50, 1991.

[144] Takis Sakkalis and Rida T. Farouki. Singular points of algebraic curves. *Journal of Symbolic Computation*, 9(4):405–421, 1990.

[145] Stefan Schirra. Robustness and precision issues in geometric computation. In J-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, chapter 14, pages 597–632. Elsevier, Amsterdam, The Netherlands, January 2000.

[146] Jacob Theodore Schwartz and Micha Sharir. On the piano movers' problem: V. The case of a rod moving in three-dimensional space amidst polyhedral obstacles. *Communications on Pure and Applied Mathematics*, 37(6):815–848, 1984.

[147] Raimund Seidel and Nicola Wolpert. On the exact computation of the topology of real algebraic curves. In *Proceedings of the 21st Symposium on Computational Geometry, Pisa, Italy, June 6-8, 2005*, pages 107–115, 2005.

[148] Vikram Sharma. Complexity of real root isolation using continued fractions. *Theoretical Computer Science*, 409(2):292–310, 2008.

[149] John M. Snyder. Interval analysis for computer graphics. *SIGGRAPH Computer Graphics*, 26(2):121–130, 1992.

[150] Alexander Stepanov and Meng Lee. The Standard Template Library. Technical Report 95-11(R.1), HP Laboratories, November 14 1995.

[151] Adam W. Strzebonski. Cylindrical algebraic decomposition using validated numerics. *Journal of Symbolic Computation*, 41(9):1021–1038, 2006.

[152] Bernd Sturmfels. *Solving Systems of Polynomial Equations*. Number 97 in CBMS Regional Conferences Series. American Mathematical Society, Providence, Rhode Island, 2002.

[153] Bernard Teissier. Cycles évanescents, sections planes et conditions de Whitney. In *Singularités à Cargèse (Rencontre Singularités Géom. Anal., Inst. Études Sci., Cargèse, 1972)*, number 7–8 in Asterisque, pages 285–362. Soc. Math. France, Paris, 1973.

[154] Sir William Thomson. *Popular lectures and addresses*. Macmillan & co., 1889.

[155] Elias P. Tsigaridas and Ioannis Z. Emiris. On the complexity of real root isolation using continued fractions. *Theoretical Computer Science*, 392:158–173, 2008.

[156] Elias P. Tsigaridas and Ioannis Z. Emiris. Real algebraic numbers and polynomial systems of small degree. *Theoretical Computer Science*, 409(2):186–199, 2008.

[157] Alan Mathison Turing. On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 2(42):230–265, 1937.

[158] James Victor Uspensky. *Theory of equations*. McGraw-Hill, New York, 1948.

[159] Joris van der Hoeven, Grégoire Lecerf, Bernard Mourrain, and Olivier Ruatta. Mathemagix: a free computer algebra system. `http://www-sop.inria.fr/galaad/mathemagix`.

[160] Alexandre Joseph Hidulphe Vincent. Sur la résolution des équations numériques. *Journal de mathématiques pures et appliquées*, 1:341–372, 1836.

[161] Georgi Fedoseevich Voronoi. Nouvelles applications des paramètres continus à la théorie de formes quadratiques. *Journal für die reine und angewandte Mathematik*, 134:198–287, 1908.

[162] Dongming Wang. *Elimination Methods (Texts and Monographs in Symbolic Computation).* Springer, January 2001.

[163] Joe Warren and Henrik Weimer. *Subdivision Methods for Geometric Design.* Morgan Kaufmann Publishers, New York, 2002.

[164] Ron Wein and Efi Fogel. The new design of CGAL's arrangement package. Technical report, Tel-Aviv University, 2005.

[165] Ron Wein, Efi Fogel, Baruch Zukerman, and Dan Halperin. 2D arrangements. In *CGAL User and Reference Manual*. CGAL Editorial Board, 3.6 edition, 2010. `http://www.cgal.org/Manual/3.6/doc_html/cgal_manual/Arrangement_on_surface_2/Chapter_main.html`.

[166] Chee Keng Yap. *Fundamental Problems of Algorithmic Algebra.* Oxford University Press, Oxford-New York, 2000.

[167] Chee Keng Yap. Robust geometric computation. In J. E. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, chapter 41, pages 927–952. Chapmen & Hall/CRC, Boca Raton, FL, 2nd edition, 2004. Revised and expanded from 1997 version.

[168] Chee Keng Yap. Complete subdivision algorithms, I: intersection of bézier curves. In *Proceedings of the 22nd Symposium on Computational Geometry, Sedona, Arizona, USA, June 5-7, 2006*, pages 217–226, 2006.

[169] Chee Keng Yap and Thomas Dubé. The exact computation paradigm. In D-Z. Du and F.K. Hwang, editors, *Computing in Euclidean Geometry*, pages 452–486. World Scientific Press, 2nd edition, 1995.

Résumé

Nous abordons dans cette thèse le problème du calcul de la topologie de courbes algébriques planes. Nous présentons un algorithme qui, grâce à l'application d'outils algébriques comme les bases de Gröbner et les représentations rationnelles univariées, ne nécessite pas de traitement particulier de cas dégénérés. Nous avons implanté cet algorithme et démontré son efficacité par un ensemble de comparaisons avec les logiciels similaires. Nous présentons également une analyse de complexité sensible a la sortie de cet algorithme. Nous discutons ensuite des outils nécessaires pour l'implantation d'algorithmes de géométrie non-linéaire dans CGAL, la bibliothèque de référence de la communauté de géométrie algorithmique. Nous présentons un noyau univarié pour CGAL, un ensemble de fonctions nécessaires pour le traitement d'objets courbes définis par des polynômes univariés. Nous avons validé notre approche en la comparant avec les implantations similaires.

Mot clés: géométrie algorithmique, géométrie non-linéaire, courbes algébriques, topologie, isotopie, bit-complexité, CGAL, noyau algébrique.

Abstract

We tackle in this thesis the problem of computing the topology of plane algebraic curves. We present an algorithm that avoids special treatment of degenerate cases, based on algebraic tools such as Gröbner bases and rational univariate representations. We implemented this algorithm and showed its performance by comparing to similar existing programs. We also present an output-sensitive complexity analysis of this algorithm. We then discuss the tools that are necessary for the implementation of non-linear geometric algorithms in CGAL, the reference library in the computational geometry community. We present an univariate algebraic kernel for CGAL, a set of functions aimed to handle curved objects defined by univariate polynomials. We validated our approach by comparing it to other similar implementations.

Keywords: computational geometry, non-linear geometry, algebraic curves, topology, isotopy, bit-complexity, CGAL, algebraic kernel.