Background and expected outcomes

- Outline the goals of the partnership and explain the potential outcomes and impacts
- Describe the importance of the topic to Canada and how the expected outcomes will benefit Canada
- Explain the new concepts or directions needed to to address the topic and how this research will fill knowledge gaps related to developing new and innovative policies, standards, products, services, processes or technologies in Canada. Position the proposed project relative to other efforts by the researchers and partner organizations and to any related research.
- Outline efforts the partner organizations will invest following the project's completion to advance the results in Canada.

Maplesoft is a Canadian software company based in Waterloo, Ontario. Maplesoft's flagship product, Maple, is a "computer algebra system". Maple has a large software library of tools for performing a wide range of mathematical computations. It is used by scientists and engineers in academia and industry all over the world, for example at Boeing, GM, and Toyota. In Canada, most universities and many colleges use Maple for teaching mathematics and for research in the mathematical, physical and engineering sciences. Maple is a Canadian success story.

The main facility Maple provides users with is software for computing with polynomials in one or more variables with various kinds of coefficients – the integers and algebraic numbers being the most important in practice. Four core operations with polynomials are (1) polynomial multiplication and division, (2) polynomial GCD computation, (3) polynomial factorization and (4) Gröbner basis computation. Many computations in Maple do these operations and consequently, the overall speed and capability of Maple depends on the algorithms used for these operations.

Previous joint work by Monagan and Pearce, in [1, 2, 3, 4], included new serial algorithms and parallel algorithms for (1) and (4) and a new polynomial data structure [5, 6]. This work was integrated into Maple under an NSERC CRD grant with Maplesoft. It dramatically improved Maple's performance on (1) and (4).

This R&D proposal aims to improve Maple's performance on (2) and (3) by developing new algorithms which use new sparse polynomial interpolation tools and parallelizing the main steps in those algorithms for multi-core computers. The kind of speedups we aim for are factors of 100 or more for serial computations multiplied by the parallel speedup. This will enable much larger computations to be undertaken.

Maplesoft's main competitors in the international market are Wolfram's *Mathematica* and the open source system *SageMath*. If successful this project will give Maple a competitive advantage on (2) and (3) thus improving Canada's ability to compete in the international market. Another benefit to Canada is the HQP trained in computer algebra, parallel programming, and Maple.

Maplesoft will help integrate the software developed under this R&D proposal into Maple where it will be announced to and deployed to Canadian universities, colleges and companies that use Maple through license agreements.

Partnership

- List all partner organizations expected to play a key role in the activities or to make cash and/or in-kind contributions.
- Describe the core activity of the partner organizations and their experience related to the research project, such as any efforts to date that the partner organizations have invested

toward addressing this problem, the need for this research project and how the topic is relevant and aligned with the partner organizations' activities.

- Explain how each partner organization will be actively involved (through cash and/or in-kind contributions) in co-designing and implementing the research program. Describe the value added through in-kind contributions and how these are important to realizing the project's intended outcomes.
- Outline each partner organization's strategy and capacity to translate the research results into practical application to achieve the desired outcomes and impacts, including any planned knowledge translation activities and integration of the research results into its operations.

The research partner on this research proposal is Maplesoft. The main R&D group at Maplesoft has as its primary mission the development and application of Maple. Maplesoft personnel have previously worked with the applicant and other researchers on R&D projects in computer algebra.

Maplesoft will make a cash contribution of $45,000 (includes university overhead) plus the inkind contributions described in the proposal section which include help with algorithm design and implementation, parallel implementations, student training, software integration into Maple, and Maple kernel support for critical subalgorithms.

The software developed by this research effort will be integrated into the Maple library so that it is automatically used by Maple users. During the software development process, when subalgorithms critical to the efficiency are identified, Maplesoft personnel will assist by coding those subalgorithms in the Maple kernel in the C++ programming language. Another area where Maplesoft personnel will play a key role is the design and integration of parallel algorithms into Maple.

The software developed will be announced to the computer algebra research community at the ISSAC and ACA conferences and to the Maple user community at the annual Maple conference.

Proposal

- Indicate approximate timelines for the activities to lead to milestones and deliverables using a Gantt chart, table or diagram.
- Explain how sex, gender and diversity have been considered in the research design, if applicable.
- Identify the indicators and methods for monitoring progress during the project and for assessing the outcomes. You may include a chart or table.
- Outline the research objectives. Detail the resources and activities needed to achieve the anticipated results.

Our proposal consists of three software projects P1, P2 and P3, described below. Approximate timelines are listed for each project. Consideration of EDI in the research design is not applicable.

Project P1: The polynomial GCD problem.

Team members: Monagan, Wittkopf (Maplesoft) and one PhD student.

Let A and B be two polynomials in $\mathbb{Z}[x_0, x_1, \ldots, x_n]$ and let $G = \gcd(A, B)$. To compute the GCD G, most computer algebra systems, including Maple, Magma and Mathematica, use Zippel's GCD algorithm from [7]. Zippel's algorithm picks a main variable, say x_0, computes G modulo a sequence of primes $p_1, p_2, \ldots$ then recovers the integer coefficients of G using Chinese remaindering. The most expensive and difficult step is computing G modulo p_1 the first prime.

Let $d_i = \deg(G, x_i)$ and $G = \sum_{i=0}^{d_0} a_i(x_1, \ldots, x_n) x_0^i$. Let $D = \sum_{i=1}^{n} d_i$ and t be the number of terms of the largest coefficient a_i of G. Zippel's algorithm uses $O(Dt)$ points to interpolate the coefficients $a_i(x_1, \ldots, x_n)$ modulo p_1. In [9] Monagan and Hu showed how to combine a Kronecker substitution with Ben-Or/Tiwari sparse interpolation [8, 10, 11] to reduce the number of points to $O(t)$ for a speedup of a factor of $O(D)$. Their algorithm is practical and they obtained speedups of 1000 and more for large inputs A and B.

To build a GCD code that works well in practice for all input types we propose to re-design the GCD algorithm so that it interpolates the smaller of G, $\bar{A}$ and $\bar{B}$ where $\bar{A} = A/G$ and $\bar{B} = B/G$. Second, we must handle inputs where Kronecker substitutions require $p > 2^{63}$. Ideas for compressing a Kronecker substitution by van der Hoeven and Lecerf in [12] will be explored. Monagan and Wittkopf will do this in the first year of the project.

For multivariate polynomials with coefficients in an algebraic number field $\mathbb{Q}(\alpha_1, \ldots, \alpha_k)$ Maple currently uses Zippel's sparse interpolation [13]. Monagan, Wittkopf, and a PhD student will apply the new sparse interpolation technology to the algebraic number case in years 2 and 3. We will also try using a primitive element $\gamma = \sum_{i=1}^{k} c_i \alpha_i$ to map $\mathbb{Q}(\alpha_1, \ldots, \alpha_k)$ into $\mathbb{Q}(\gamma)$, after reduction modulo p, to speed up coefficient arithmetic.

Project P2: The polynomial factorization problem.

Team members: Monagan, Chen (PhD student), Gerhard (Maplesoft) and one MSc student.

The problem is to factor a polynomial in n variables with integer coefficients. In previous work, Monagan and Tuncer in [14, 15] used sparse interpolation to solve the multivariate polynomial diophantine (MPD) equations that arise in multivariate Hensel lifting [16, 17, 18]. This work was integrated into Maple by Tuncer in 2018 under a MITACS internship with Maplesoft [19].

Subsequent work by Monagan and Tuncer in [20] and Monagan and Chen in [21] experimented with a new approach where we interpolate the factors directly from bivariate images obtained with Hensel lifting thereby eliminating the expensive MPD equations from the factorization algorithm. To speed up bivariate Hensel lifting, in [22], Monagan developed a new cubic cost algorithm. These experiments were for monic polynomials with two factors only.

To build a complete new polynomial factorizer Chen, Monagan and Gerhard will work on the non-monic case and multi-factor case of the new approach in the first year. In the second year they will work on factoring an input polynomial given by a black box. Our interest in black box factorization is motivated by the observation that the factorization algorithms in [20, 21] are often dominated by the cost of evaluating the input polynomial. Unlike previous work on factoring polynomials given by black boxes by Kaltofen, Trager and Diaz [23, 24] and Rubinfeld and Zippel [25], our work aims to interpolate the factors in their sparse representation directly.

To further speed up the algorithms in P1 and P2 we must either reduce the number of evaluation points needed, or speed up evaluations, or parallelize the evaluations. In the second and third years the team will work on parallelizing the evaluations for multi-core computers.

Project P3: A 64 bit integer library for $\mathbb{Z}_p[x]$ and $\mathbb{Z}_p[x, y]$.

Team members: Monagan, Paluck (PhD student), DeMarco (Maplesoft).

Projects P1 and P2 require various computations in $\mathbb{Z}_p[x]$ and $\mathbb{Z}_p[x, y]$ where p is a machine prime. Because of the Kronecker substitution in P1, and the failure probabilities of the algorithms in P1 and P2 (they are probabilistic), it is necessary to use 63 bit primes rather than 31 bit primes. Currently Maple only has C code for 31 bit primes. Project P3 is to design and implement a new C library for $\mathbb{Z}_p[x]$ and $\mathbb{Z}_p[x,y]$ which supports 63 bit primes and fast (FFT based) arithmetic. The focus will be support subroutines for P1 and P2.

In the first year we will build a library that includes polynomial arithmetic, bivariate Hensel lifting and sparse interpolation tools (root finding, solving Vandermonde systems, etc.) In the second year, to support P1 we will add an implementation of the Tangent-Graeffe root finding algorithm from [26] which we showed in [27] is significantly faster than Cantor-Zassenhaus [28], and for P2 we will add a bivariate GCD that uses our new Hensel lifting [22]. In the third year we will experiment with using the new sparse interpolation tools to interpolate the determinants of matrices of multivariate polynomials, for example, Dixon resultant matrices (see Lewis [29]). Dixon resultants are an alternative to Gröbner bases for solving systems of polynomial equations.

Milestones and Progress Monitoring: We propose to meet once a year in person, in Waterloo, and at least twice a year on-line for progress updates and to set new milestones. Progress indicators include demonstrating working software, benchmarking the software against other computer algebra systems, and integration of the software into Maple.

Resources needed: For computing resources the academic team will use the CECM Research Centre's computing facilities (see www.cecm.sfu.ca) at Simon Fraser which includes a new multi-core server with two 24 core Intel Gold 6342 CPUs which Maplesoft helped pay for.

Team

- List the applicant, any co-applicants and key staff of the partner organizations.
- Explain how the knowledge, experience and achievements of these individuals provide the expertise needed to accomplish the project objectives. Discuss the role of each individual and how their contributions, including those of staff from the partner organizations, will be integrated into the project.
- Explain how equity, diversity and inclusion have been considered in the academic team composition.

The applicant is Michael Monagan and the Maplesoft personnel contributing to this project include Dr. Paulina Chin, Dr. Jürgen Gerhard, Mr. Paul DeMarco, and Dr. Allan Wittkopf.

Michael Monagan, has 38 years experience in Computer Algebra which includes parallel algorithm design and implementation [1, 2, 4, 9, 20, 21, 27]. He is a leading researcher in polynomial GCD computation and polynomial factorization. Monagan has worked on three joint R&D projects with Maplesoft since 2004; two NSERC CRD projects from 2004–2007 and 2013–2017 as PI and a MITACS NCE project from 2005–2012 as co-PI.

Paulina Chin is a senior software architect at Maplesoft. She will provide student training in Maple programming, testing, documentation and do code reviews.

Paul DeMarco is the director of Maple development at Maplesoft. He leads the Maple kernel development. Paul will oversee the development and integration of C code into the Maple kernel for project P3. Paul will also develop Maple kernel support software needed to support parallel algorithms in all three projects.

Jürgen Gerhard is the director of research at Maplesoft. He is co-author of the main textbook in computer algebra [30] which includes chapters on asymptotically fast polynomial arithmetic and polynomial factorization. Jürgen will help with the design and implementation of the algorithms for the polynomial factorization project P2 and project management and reporting.

Allan Wittkopf is a senior software architect at Maplesoft. He is the primary author of Maple's current polynomial GCD algorithm [31] and modular resultant algorithm – both for $\mathbb{Z}[x_1, \ldots, x_n]$. Allan will help with the design and implementation of the GCD algorithms for project P1.

We have considered EDI by asking Dr. Paulina Chin to provide student training. To recruit graduate students, when I will advertise the project on my personal web page, I will encourage students from racial, religious and gender minorities to apply and I will make a conscious effort to select from those who apply in an unbiased manner.

Training Plan

- Indicate how the knowledge and experience gained by research trainees and the partners' staff members are relevant to the advancement of the field, to applying knowledge or to strengthening the partners' sectors.
- Describe how the project and the partnership offer opportunities for enriched training experiences that will allow research trainees (undergraduates, graduates and postdoctoral fellows) to develop relevant technical skills as well as professional skills, such as leadership, communication, collaboration and entrepreneurship. Include the nature of the planned interactions with the partners and other relevant activities.
- Explain how equity, diversity and inclusion are considered in the training plan.

General skills that students will acquire include (1) programming and software engineering skills, (2) parallel algorithm development, (3) presentation skills (oral and written), and (4) how to use Maple to perform a variety of mathematical computations.

Monagan will inform and train the research partner and students on sparse interpolation tools and where they can be applied. Monagan and DeMarco will train students in the Cilk C and/or Open C parallel programming environments. The experience gained from the development of parallel algorithms will enable the research partner to develop future applications for Maple which exploit multi-core computers.

The research trainees will accompany Monagan on field trips to the Maplesoft company site in Waterloo to work with the research partner on the project and receive Maple training from company personnel, including Maple programming, software testing and software engineering. The budget allows for longer term visits (3 to 4 weeks) for two trainees, a strength of the proposal. We will attend the annual Maple conference where trainees can (i) present their work to the wider Maple user community, (ii) see and learn how to use Maple to do mathematics and teach mathematics, and (iii) connect with Maplesoft company personnel.

The department of mathematics at Simon Fraser University wants EDI training to be given often and from a variety of faculty and staff so that a culture of respect for others is promoted. I am attending the formal EDI training events for faculty every semester and EDI training will be given during the Fall semester to all graduate students.

Also, our department has started a monthly EDI discussion group which I am attending. Because my graduate students come from different races, cultures and religions, I will require them to attend those meetings which discuss racism, sexual harassment in the workplace, and the LGBTQ community in particular.

To ensure equity when it comes to conference travel, I have constructed the travel budget so that each graduate student can go to at least one international conference and two Maplesoft site visits. In addition, our department's travel budget will pay for each PhD student to present their work at one national or international conference.

References

[1] Michael Monagan and Roman Pearce. Parallel Sparse Polynomial Multiplication using Heaps. *Proceedings of ISSAC '09*, pp. 263–269, ACM, 2009.

[2] Roman Pearce and Michael Monagan. Parallel Sparse Polynomial Division using Heaps. *Proceedings of PASCO 2010*, pp. 105–111, ACM, 2010.

[3] Michael Monagan and Roman Pearce. Sparse Polynomial Division Using a Heap. *J. Symb. Cmpt.* **46**(7):807–822, Elsevier, 2011.

[4] Michael Monagan and Roman Pearce. A Compact Parallel Implementation of F4. *Proceedings of PASCO 2015*, pp. 95–100, ACM, 2015.

[5] Michael Monagan and Roman Pearce. POLY: A new polynomial data structure for Maple 17. In *Computer Mathematics*, pp. 325–348, Ruyong Feng, Wen-shin Lee, Yosuke Sato editors, Springer, 2014. `http://link.springer.com/chapter/10.1007%2F978-3-662-43799-5_24`

[6] Michael Monagan and Roman Pearce. The design of Maple's sum-of-products and POLY data structures for representing mathematical objects. *Communications of Computer Algebra*, **48**(4):166–186, December 2014.

[7] Richard Zippel, Probabilistic algorithms for sparse polynomials, *Proceedings of EUROSAM '79*, **2**:216–226, Springer, 1979.

[8] Michael Ben-Or and Prasoon Tiwari. A deterministic algorithm for sparse multivariate polynomial interpolation. *Proceedings of STOC '88*, pages 301–309. ACM, 1988.

[9] Jiaxiong Hu and Michael Monagan. A Fast Parallel Sparse Polynomial GCD Algorithm. *J. Symb. Cmpt.* **105**:(1) 28–63, Elsevier, 2021. https://doi.org/10.1016/j.jsc.2020.06.001

[10] Hirokazu Murao and Tetsuro Fujise. Modular Algorithm for Sparse Multivariate Polynomial Interpolation and its Parallel Implementation. *J. Symb. Cmpt.* **21**:377–396, Elsevier, 1996.

[11] Erich Kaltofen: Fifteen years after DSC and WLSS2 what parallel computations I do today. Invited talk. *Proceedings of PASCO 2010*, 10–17, ACM, 2010.

[12] Joris van der Hoeven and Grégoire Lecerf. Sparse Polynomial Interpolation in Practice. *Communications in Computer Algebra* **48**:187–191, ACM, 2015.

[13] Mahdi Javadi and Michael Monagan. A Sparse Modular GCD Algorithm for Polynomial GCD Computation over Algebraic Function Fields. *Proceedings of ISSAC '07*, pp. 187–194. ACM, 2007.

[14] Michael Monagan and Baris Tuncer. Using Sparse Interpolation in Hensel Lifting. *Proceedings of CASC 2016*, LNCS **9890**:381–400, Springer, 2016.

[15] Michael Monagan and Baris Tuncer. Factoring multivariate polynomials with many factors and huge coefficients. *Proceedings of CASC 2018*, LNCS **11077**:319–334, Springer, 2018.

[16] K.O. Geddes, S.R. Czapor, G. Labahn. Chapter 6. Newton's Iteration and the Hensel Construction. *Algorithms for Computer Algebra*, Kluwer Acad. Publ. (1992).

[17] Wang, P.S., Rothschild, L.P. Factoring multivariate polynomials over the integers. *Mathematics of Computation*, **29**(131):935–950, 1975.

[18] Erich Kaltofen, Sparse Hensel lifting. *Proceedings of EUROCAL '85*, LNCS **204**:4–17, Springer, 1985.

[19] Michael Monagan and Baris Tuncer. Polynomial Factorization in Maple 2019. *Proceedings of the 2019 Maple conference*, *Communications in Computer and Information Science*, **1125**:341–345, Springer, 2020.

[20] Michael Monagan and Baris Tuncer. Sparse multivariate polynomial factorization: a high-performance design and implementation. *Proceedings of ICMS 2018*, LNCS **10931**:359–368, Springer, July 2018.

[21] Tian Chen and Michael Monagan. The Complexity and Parallel Implementation of two Sparse Multivariate Hensel Lifting Algorithms for Polynomial Factorization. *Proceedings of CASC 2020*, LNCS **12291**:150–169, Springer, 2020.

[22] Michael Monagan. Linear Hensel Lifting for $\mathbb{F}_p[x, y]$ and $\mathbb{Z}[x]$ with Cubic Cost. *Proceedings of ISSAC 2019*, pp. 299–306, ACM Digital Library, July 2019.

[23] E. Kaltofen, B. Trager, Computing with polynomials given by black boxes for their evaluations: Greatest common divisors, factorization, separation of numerators and denominators. *J. Symb. Comp.*, **9**:301–320, Elsevier, 1990.

[24] Diaz A., Kaltofen E.: FOXBOX: A system for manipulating symbolic objects in black box representation. *Proceedings of ISSAC '98*, pp. 30–37, ACM, 1998.

[25] Ronitt Rubinfeld and Richard Zippel, A New Modular Interpolation Algorithm for Factoring Multivariate Polynomials. Technical Report 93–1326, Dept. Comp. Sci., Cornell Univ., 1993.

[26] Bruno Grenet and Joris van der Hoeven. Randomized Root Finding over Finite FFT-fields using Tangent Graeffe Transforms. *Proceedings of ISSAC 2015*, pp. 197–204, ACM, 2015.

[27] Joris van der Hoven and Michael Monagan. Computing one billion roots using the tangent Graeffe method. *Communications in Computer Algebra* **54**(3): 65–85, September 2020. DOI https://doi.org/10.1145/3457341.3457342

[28] G. Cantor and H. Zassenhaus. A new algorithm for factoring polynomials over finite fields. *Math. Comp.* **36**(154):587–592, 1981.

[29] Robert Lewis. Dixon-EDF: The Premier Method for Solution of Parametric Polynomial Systems. In *Proceedings in Mathematics and Statistics*, **198**, pp. 238–256, Springer, 2017.

[30] Joachim von zur Gathen and Jürgen Gerhard. *Modern Computer Algebra.* 3rd edition, University of Cambridge Press, 2013.

[31] J. de Kleine, M. B. Monagan, A. Wittkopf. Algorithms for the Non-monic case of the Sparse Modular GCD Algorithm. *Proceedings ISSAC '05*, pp. 124–131, ACM, 2005.