High-performance computing and symbolic computation

Marc Moreno Maza

Ontario Research Center for Computer Algebra (ORCCA)
University of Western Ontario, Canada

September 9, 2016

Research themes and team members

- Symbolic computation: computing exact solutions of algebraic problems on computers with applications to sciences and engineering.
- High-performance computing: making best use of modern computer architectures, in particular hardware accelerators (multi-cores GPUUs)

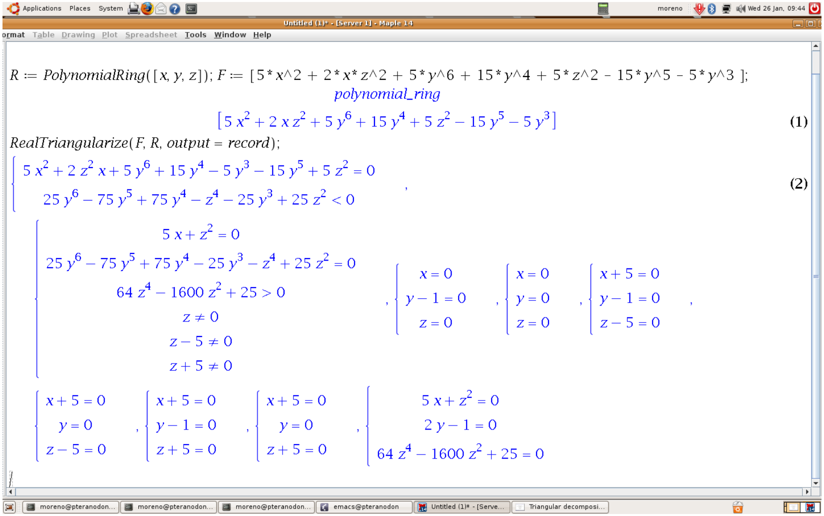
Current students

PhD:	Parisa Alvandi, Ning Xie, Mahsa Kazemi, Ruijuan Jing, Xiaohui Chen, Steven Thornton, Robert Moir, Egor Chesakov
MSc:	Masoud Ataei, Yiming Guan, Davood Mohajerani

Alumni

Moshin Ali (ANU , Australia) Jinlong Cai (Microsoft , USA), Changbo Chen (Chinese Acad. of Sc.), Svyatoslav Covanov (U. Lorraine , France) Akpodigha Filatei (Guaranty Turnkey Systems ltd , Nigeria) Oleg Golubitsky (Google Canada) Sardar A. Haque (GeoMechanica , Canada) Zunaid Haque (IBM Canada) François Lemaire (U. Lille 1 , France) Farnam Mansouri (Microsoft , Canada) Liyun Li (Banque de Montréal , Canada) Xin Li (U. Carlos III , Spain) Wei Pan (Intel Corp. , USA) Sushek Shekar (Ciena , Canada) Paul Vrbik (U. Newcastle , Australia) Yuzhen Xie (Critical Outcome Technologies , Canada) Li Zhang (IBM Canada) . . .

Solving polynomial systems symbolically


Applications Places System
Untitled (1)* - [Server 1] - Maple 14
ormat Table Drawing Plot Spreadsheet Tools Window Help
moreno
Wed 26 Jan, 09:44
moreno@pteranodon...
emacs@pteranodon
Untitled (1)* - [Serve...
Triangular decomposi...

```
R := PolynomialRing([x, y, z]); F := [5*x^2 + 2*x*z^2 + 5*y^6 + 15*y^4 + 5*z^2 - 15*y^5 - 5*y^3 ];
```

polynomial_ring

$$[5x^2 + 2xz^2 + 5y^6 + 15y^4 + 5z^2 - 15y^5 - 5y^3] \tag{1}$$

```
RealTriangularize(F, R, output = record);
```

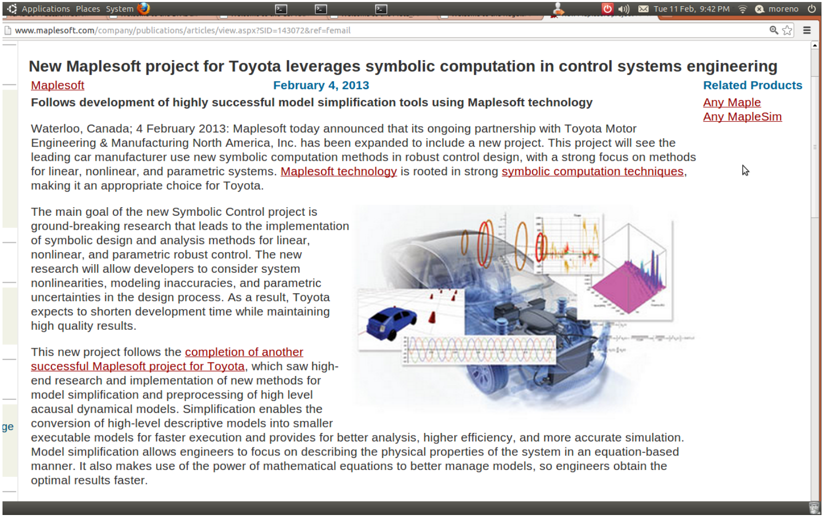
$$\left\{\begin{array}{c} 5x^2 + 2z^2x + 5y^6 + 15y^4 - 5y^3 - 15y^5 + 5z^2 = 0 \\ 25y^6 - 75y^5 + 75y^4 - z^4 - 25y^3 + 25z^2 < 0 \end{array}\right., \tag{2}$$

$$\left\{\begin{array}{c} 5x + z^2 = 0 \\ 25y^6 - 75y^5 + 75y^4 - 25y^3 - z^4 + 25z^2 = 0 \\ 64z^4 - 1600z^2 + 25 > 0 \\ z \neq 0 \\ z - 5 \neq 0 \\ z + 5 \neq 0 \end{array}\right., \left\{\begin{array}{c} x = 0 \\ y - 1 = 0 \\ z = 0 \end{array}\right., \left\{\begin{array}{c} x = 0 \\ y = 0 \\ z = 0 \end{array}\right., \left\{\begin{array}{c} x + 5 = 0 \\ y - 1 = 0 \\ z - 5 = 0 \end{array}\right.,$$

$$\left\{\begin{array}{c} x + 5 = 0 \\ y = 0 \\ z - 5 = 0 \end{array}\right., \left\{\begin{array}{c} x + 5 = 0 \\ y - 1 = 0 \\ z + 5 = 0 \end{array}\right., \left\{\begin{array}{c} x + 5 = 0 \\ y = 0 \\ z + 5 = 0 \end{array}\right., \left\{\begin{array}{c} 5x + z^2 = 0 \\ 2y - 1 = 0 \\ 64z^4 - 1600z^2 + 25 = 0 \end{array}\right.$$

Figure: The *RegularChains* solver designed in our UWO lab is at the heart of Maple, which has about 5,000,000 world-wide.

Application to mathematical sciences and engineering



Applications Places System
Tue 11 Feb, 9:42 PM
moreno
www.maplesoft.com/company/publications/articles/view.aspx?SID=143072&ref=femail

New Maplesoft project for Toyota leverages symbolic computation in control systems engineering

Maplesoft February 4, 2013

Follows development of highly successful model simplification tools using Maplesoft technology

Waterloo, Canada; 4 February 2013: Maplesoft today announced that its ongoing partnership with Toyota Motor Engineering & Manufacturing North America, Inc. has been expanded to include a new project. This project will see the leading car manufacturer use new symbolic computation methods in robust control design, with a strong focus on methods for linear, nonlinear, and parametric systems. Maplesoft technology is rooted in strong symbolic computation techniques, making it an appropriate choice for Toyota.

The main goal of the new Symbolic Control project is ground-breaking research that leads to the implementation of symbolic design and analysis methods for linear, nonlinear, and parametric robust control. The new research will allow developers to consider system nonlinearities, modeling inaccuracies, and parametric uncertainties in the design process. As a result, Toyota expects to shorten development time while maintaining high quality results.

This new project follows the completion of another successful Maplesoft project for Toyota, which saw high-end research and implementation of new methods for model simplification and preprocessing of high level acausal dynamical models. Simplification enables the conversion of high-level descriptive models into smaller executable models for faster execution and provides for better analysis, higher efficiency, and more accurate simulation. Model simplification allows engineers to focus on describing the physical properties of the system in an equation-based manner. It also makes use of the power of mathematical equations to better manage models, so engineers obtain the optimal results faster.

Related Products

Any Maple

Any MapleSim

Figure: Toyota engineers use our software to design control systems

High-performance computing: parallel program translation

```
int main(){
  int sum_a=0, sum_b=0;
  int a[ 5 ] = {0,1,2,3,4};
  int b[ 5 ] = {0,1,2,3,4};
  #pragma omp parallel
  {
    #pragma omp sections
    {
      #pragma omp section
      {
        for(int i=0; i<5; i++)
          sum_a += a[ i ];
      }
      #pragma omp section
      {
        for(int i=0; i<5; i++)
           sum_b += b[ i ];
      }  }  }
}
```

```
int main()
{
  int sum_a=0, sum_b=0;
  int a[ 5 ] = {0,1,2,3,4};
  int b[ 5 ] = {0,1,2,3,4};

  meta_fork shared(sum_a){
    for(int i=0; i<5; i++)
       sum_a += a[ i ];
      }

  meta_fork shared(sum_b){
    for(int i=0; i<5; i++)
       sum_b += b[ i ];
      }

  meta_join;
}
```

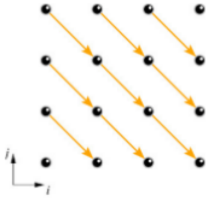
```
void fork_func0(int* sum_a,int* a)
{
        for(int i=0; i<5; i++)
          (*sum_a) += a[ i ];
}
void fork_func1(int* sum_b,int* b)
{
        for(int i=0; i<5; i++)
           (*sum_b) += b[ i ];
}
int main()
{
  int sum_a=0, sum_b=0;
  int a[ 5 ] = {0,1,2,3,4};
  int b[ 5 ] = {0,1,2,3,4};
  cilk_spawn fork_func0(&sum_a,a);
  cilk_spawn fork_func1(&sum_b,b);
  cilk_sync;
}
```

Our lab develops a compilation platform for translating parallel programs from one language to another; above we translate from `OpenMP` to `CilkPlus` through `MetaFork`. This project is supported by IBM Canada.

High-performance computing: automatic parallelization

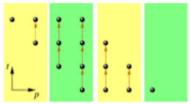
Serial dense univariate polynomial multiplication

```
for(i=0; i<=n; i++){
  for(j=0; j<=n; j++)
    c[i+j] += a[i] * b[j];
}
```



GPU-like multi-threaded dense univariate polynomial multiplication

```
meta_for (b=0; b<= 2 n / B; b++) {
    for (u=0; u<=min(B-1, 2*n - B * b); u++) {
       p = b * B + u;
       for (t=max(0,n-p); t<=min(n,2*n-p) ;t++)
           c[p] = c[p] + a[t+p-n] * b[n-t];
   }
}
```



p:	0	1	2	3	4	5	6
r:	0	0	1	1	0	0	1
o:	0	1	0	1	0	1	0
b:	0	0	1	1	2	2	3
s:	0	0	0	0	1	1	1

We use symbolic computation to automatically translate serial programs to GPU-like programs.

Research projects with publicly available software

Basic Polynomial Algebra Subprograms

www.bpaslib.org

MetaFork

www.metafork.org

CUDA Modular Polynomial $\in \mathbb{F}_p[X_1 .. X_s]$

www.cumodp.org

www.regularchains.org

Courses

Courses

CS 2101	Foundations of Programming for High Performance Computing
CS 3101	Theory and Practice of High-performance Computing
CS 3350	Computer Architecture
CS 9535/4402	Distributed and Parallel Systems

CS 9535/4402

- multi-core, GPGPU, hierarchical memory,
- fork-join concurrency, SIMD, message passing
- parallel algorithms: design and complexity analysis
- scheduling (work-stealing scheduler) and synchronization
- languages: `CilkPlus`, `CUDA`, `MPI`
- student evaluation: quizzes, assignments (mainly CS4402), research projects (mainly CS9535/)