

High-performance computing and symbolic computation

Marc Moreno Maza

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

September 19, 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

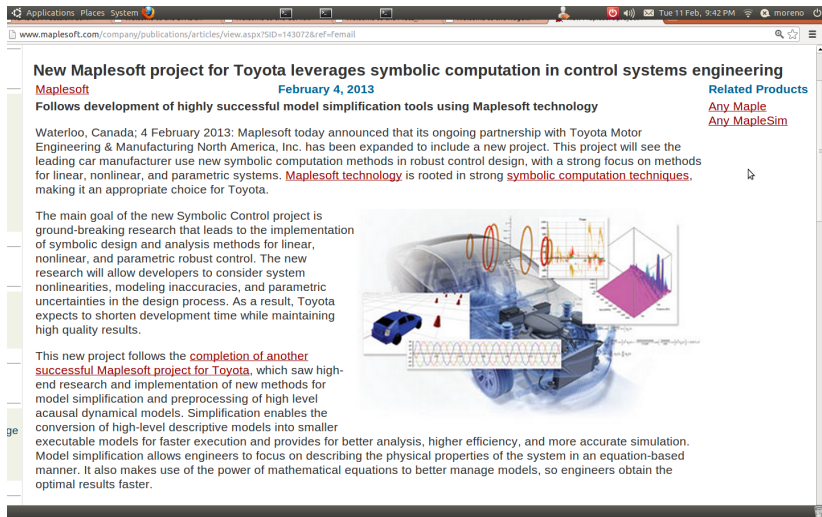
PhD: Parisa Alvandi, Ning Xie, Mahsa Kazemi, Ruijuan Jing, Xiaohui Chen, Steven Thornton, Robert Moir, Egor Chesakov

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**) ...

Application to mathematical sciences and engineering



New Maplesoft project for Toyota leverages symbolic computation in control systems engineering
Maplesoft February 4, 2013 **Related Products**
Any Maple
Any MapleSim

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.

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 ];
}

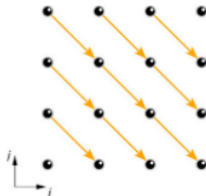
int main()
{
  int sum_a=0, sum_b=0;
  int a[ 5 ] = {0,1,2,3,4};
  int b[ 5 ] = {0,1,2,3,4};
  cilk_spawn fork_func0(&sum_a,a);
  cilk_spawn fork_func1(&sum_b,b);
  cilk_sync;
}
```

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

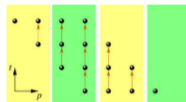
Serial dense univariate polynomial multiplication

```
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

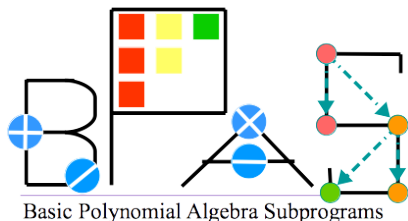
```
meta_for (b=0; b<= 2 n / B; b++) {  
  for (u=0; u<=min(B-1, 2*n - B * b); u++) {  
    p = b * B + u;  
    for (t=max(0,n-p); t<=min(n,2*n-p) ;t++)  
      c[p] = c[p] + a[t+p-n] * b[n-t];  
  }  
}
```



<i>p:</i>	0	1	2	3	4	5	6
<i>r:</i>	0	0	1	1	0	0	1
<i>a:</i>	0	1	0	1	0	1	0
<i>b:</i>	0	0	1	1	2	2	3
<i>s:</i>	0	0	0	0	1	1	1

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

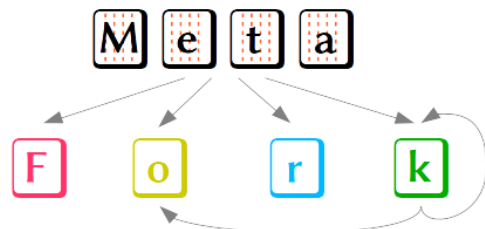
Research projects with publicly available software



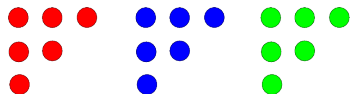
www.bpaslib.org

CUMODP $\in \mathbb{F}_p[X_1 \dots X_s]$
DA ular polynomial

www.cumodp.org



www.metafork.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/)