

Two Functional MDD's for the Price of One - Part 1

Curtis d'Alves, Nhan Thai, Nassim Khoonkari,
Padma Pasupathi, Tanya Bouman, Christopher Anand

November 6, 2019

Outline

- 1 Model Driven Development
- 2 Optimization
- 3 Example: Parabola
- 4 References

Proto Model Driven Development

- Computer-aided software engineering (CASE) is the domain of software tools used to design and implement applications.
- ISDOS project started in 1968 at the University of Michigan
- Lots of tools.
 - ▶ DB-centric tools (e.g., Object Relation Mapping tools)
 - ▶ OO-oriented tools (e.g., Eclipse Modeling Framework)
- UML, standardized 1997
 - ▶ large-scale processes
 - ▶ documenting user interaction
- skeleton generation

Model Driven Development for Numerical Computation

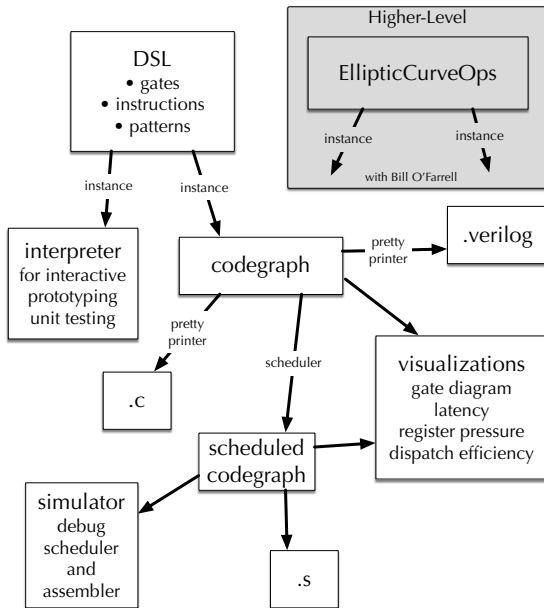
- linear algebra
 - ▶ VOP - vector-oriented programming
 - ▶ APL - Array Programming Language, 1966
 - ▶ Matlab - “mathematics for engineering”
 - ▶ Maple - symbolic computation, symbolic code generation
- optimization
 - ▶ AMPL, 1985
 - ▶ GAMS
- mathematical interface to some very powerful software
- great for design exploration

Model Driven Development

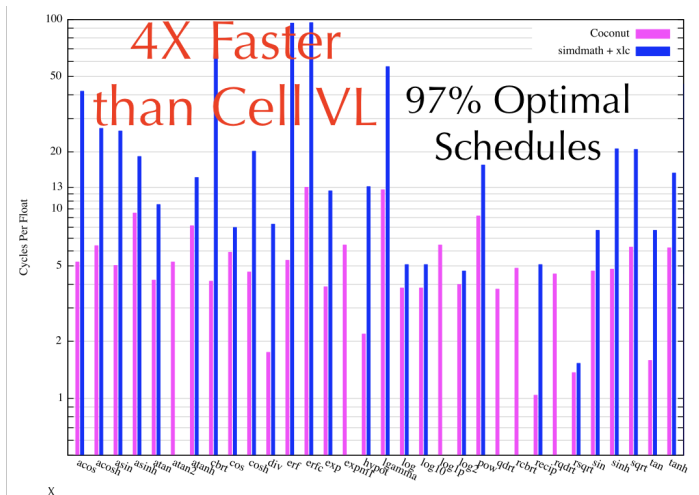
- some problems with CASE tools
 - ▶ generating skeletons, makes it hard to regenerate
 - ▶ suffers same problem as documentation
 - ▶ people who can update model are too busy to do so
 - ▶ easier to hack on generated code
- new vision (e.g. Selic, 2003)
 - ▶ primary focus and products are models rather than computer programs
 - ▶ technology-independent specifications
 - ▶ requires
 - ★ generating complete programs from models
 - ★ verifying models on a computer
 - ▶ looking at code should be as rare as looking at assembly

- COnstruction User Tool, 2004
 - ▶ nested Domain-Specific Languages
 - ▶ functional assembly language
 - ▶ higher-level patterns
 - ▶ principled graph transformations

MDD - Coconut



MDD - Coconut



MDD - Coconut

- C**O**de C**O**Nstruction User Tool

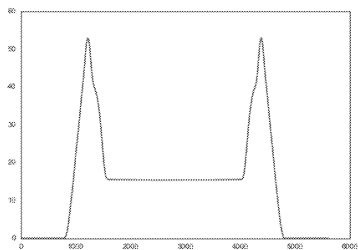
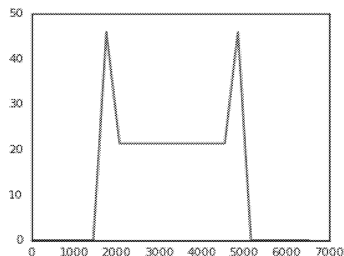
- ▶ used to generate MASS for all IBM platforms since Cell/B.E.
- ▶ but not used by users outside McMaster / OCA
- ▶ why?
 - ★ have to learn Haskell
 - ★ too many interfaces geared to abandoned research projects
- ▶ fix = fix up the code
 - ★ open source!
 - ★ potential embarrassment will motivate clean up
- ▶ open Coconut in pieces

today	HashedExpression	calculus used for continuous optimization
later	CodeGraph library	Intermediate Lanugage
later	DSLs + interpreters	front-ends
later	scheduler + simulator	back-end

Optimization

- maximize benefit
- minimize cost
- satisfy constraints
- make anything better!
- not just instruction scheduling

Optimization - MRI



- big and little magnets manipulate proton spins
- antenna picks up signal
- little magnets controlled by “gradient waveforms”
- common designs (left) are not realizable
- reformulate as an optimization problem (producing right)

Optimization - MRI

- key insight: important waveforms are periodic
- Fourier Transform is discrete
- optimize over finite set of coefficients

$$\min_{\{h_f: f \in F\}} \delta \quad (1)$$

$$\text{subject to } |\mu - g(t)| \leq \delta, \quad \forall t \in S \quad (2)$$

$$|g(t)| \leq g_{\text{peak}}, \quad \forall t \quad (3)$$

$$|\partial g / \partial t(t)| \leq g_{\text{slew}}, \quad \forall t \quad (4)$$

$$\int_{t \in S} g(t) = A \quad (5)$$

$$\text{where } g(t) = \sum_{f \in F} h_f \sin(2\pi t f) \quad (6)$$

Optimization - MRI

- developed with AMPL/neos
- needed parametrized family for production
- reimplement in Python
- want one tool for development and deployment



US 20190064301A1

(19) United States	
(12) Patent Application Publication	(10) Pub. No.: US 2019/0064301 A1
Curtis et al.	(43) Pub. Date: Feb. 28, 2019
<hr/>	
(54) METHOD AND SYSTEM OF FREQUENCY CONSTRAINED GRADIENT WAVEFORM PRODUCTION	Publication Classification
(71) Applicants: Andrew Thomas Curtis , Toronto (CA); Christopher Kumar Anand , Toronto (CA); Chad Tyler Harris , Toronto (CA); Jeff Alan Stainsby , Toronto (CA); Phil J. Beutry , Toronto (CA)	(51) Int. CL <i>G01R 33/565</i> (2006.01) <i>G01R 33/389</i> (2006.01)
(72) Inventors: Andrew Thomas Curtis , Toronto (CA); Christopher Kumar Anand , Toronto (CA); Chad Tyler Harris , Toronto (CA); Jeff Alan Stainsby , Toronto (CA); Phil J. Beutry , Toronto (CA)	(52) U.S. CL CPC <i>G01R 33/56509</i> (2013.01); <i>G01R 33/389</i> (2013.01)
(21) Appl. No.: 15/690,795	(57) ABSTRACT The present disclosure provides a method and system of magnetic resonance imaging using a constrained gradient waveform. The constrained gradient waveform is designed to have only predetermined frequencies, for example excluding one or more identified resonant frequencies associated with the MRI system. The application of such a constrained gradient waveform during imaging may aid in reducing noise, vibration and/or heating of the MRI system during imaging.
(22) Filed: Aug. 30, 2017	

Optimizing A Simple Parabola

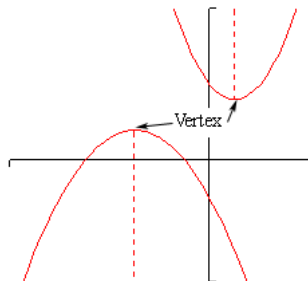


Figure: Example Parabola's: Solve for Vertex

$$\min_x \quad ax^2 + bx + c \quad (7)$$

$$\text{subject to } x \leq n \quad (8)$$

$$g(x) \leq m \quad (9)$$

AMPL - MDD Solution for Optimization

```
param a;  
param b;  
param c;  
param n;  
param m;  
  
var x >= n;  
  
minimize Obj:  
    a * x*x + b * x + c;  
  
subject to G:  
    x * x <= m;
```

```
data;  
  
param a := 1 ;  
param b := 0 ;  
param c := 5 ;  
param n := -10 ;  
param m := 0 ;
```

Figure: file: Quadratic.dat

Figure: file: Quadratic.mod

Categories of Optimization Problems / Solvers

- Many different categories of optimization
 - ▶ Linear / Non-Linear
 - ▶ Constrained / Un-Constrained
 - ▶ Integer / Mixed Integer
 - ▶ Semidefinite, stochastic, ...
- Many, many different solvers to choose from
- No one to rule them all! (Optimization is not Lord of the Rings)
- NEOS Server: fantastic resource for running AMPL models on a variety of different solvers
<https://neos-server.org/neos/solvers/index.html>

Optimization Without MDD

- Variety of solvers provide implemented in C / Java / Python
- however require you to supply auxiliary functions manually
- commonly required auxillary functions include:
 - ▶ objective function
 - ▶ gradient of the objective function
 - ▶ hessian of the objective function
 - ▶ constraint function(s)
 - ▶ jacobian of the constraint function(s)

Optimization Without MDD - C Example

```
1 void evaluate_partial_derivatives_and_objective() {
2     ptr[OBJ] = ptr[A]*ptr[X]*ptr[X] + ptr[B]*ptr[X] + ptr[C];
3     ptr[DERIVATIVE] = 2.0 *ptr[A]*ptr[X] + ptr[B];
4 }
5 void evaluate_objective() {
6     ptr[OBJ] = ptr[A]*ptr[X]*ptr[X] + ptr[B]*ptr[X] + ptr[C];
7 }
8 void evaluate_partial_derivatives() {
9     ptr[DERIVATIVE] = 2.0*ptr[A]*ptr[X] + ptr[B];
10 }
11 void evaluate_scalar_constraints() {
12     ptr[CONSTRAINT] = pow(ptr[X],2);
13 }
14 void evaluate_scalar_constraints_jacobian() {
15     ptr[CONSTRAINT_DERIVATIVE] = 2.0*ptr[X];
16 }
```

Symphony

```
1  variables :  
2      x = 0  
3  
4  constants :  
5      a = 1  
6      b = 0  
7      c = 5  
8      n = -10  
9      m = 0  
10  
11 constraints :  
12     x^2 <= m  
13  
14 minimize :  
15     a * x^2 + b * x + c  
16
```

Parts II and III

- more optimization: TODO
 - ▶ example: Integrating Blood Flow from PCA-MRI
 - ▶ example: MRI Image Reconstruction with Mask
 - ▶ example: Parallel MRI
 - ▶ example: Logistics
 - ▶ Hashed Expression (Symphony's Backend)
- PAL: Petri App Land
 - ▶ State Diagrams can model user interaction (grade 5+)
 - ▶ PALDraw: graphical modelling tool for State Diagrams
 - ▶ make a working app
 - ▶ Petri Nets can model multi-user interaction in web apps
 - ▶ PALDraw: graphical modelling tool for Petri App Land

References