MAPP: A Platform for Prototyping Algorithms and Models Quickly and Easily

Tianshi Wang, Karthik Aadithya, Bichen Wu and Jaijeet Roychowdhury

EECS Department
University of California at Berkeley
Motivation for MAPP

- **Berkeley Model and Algorithm Prototyping Platform**
- Developing **good** compact models: many pitfalls
  - examples: discontinuities/smoothness, well-posedness
    - problems usually discovered at deployment (ie, during simulation)
  - problems often hard to debug and resolve
    - compact model developer and simulator people blame each other
- Anyone working in simulation algorithms today needs
  - device models: BSIM, MOS1, MOS2, MOS3, ...
  - base algorithms: **robust nonlinear solution**, transient, HB/shooting, ...
  - parsing, equation formulation, output, ...
  - huge (waste of) effort of re-development of basic capabilities
- One goal of MAPP: to ease these problems
  - common, open-source simulation framework
  - in MATLAB
    - empowers non-programmers to debug models and algorithms
Why not use SPICE?

- SPICE: the original open-source simulator
  - de-facto standard
  - structure: all analyses in all models
  - prototyping models & algorithms: takes months to years
  - pain to write (even for those who can)
    - e.g., shooting method (S-SPICE)

- To be useful: modular, well-structured, flexible
  - separated models, algorithms, numerics, I/Os
  - simple, clean interfaces
  - short, easy to read, easy to modify
Excerpt from `dioaload.c` (SPICE3)

```c
#ifdef SENSDEBUG
    printf("vd = %7.2e \n",vd);
#endif /* SENSDEBUG */
goto next1;
}
if(ckt->CKTmode & MODEINITSMSIG) {
    vd= *(ckt->CKTstate0 + here->DIOvoltage);
} else if (ckt->CKTmode & MODEINITTRAN) {
    vd= *(ckt->CKTstate1 + here->DIOvoltage);
} else if ((ckt->CKTmode & MODEINITJCT) && (ckt->CKTmode & MODETRANOP) && (ckt->CKTmode & MODEUIC)) {
    vd=here->DIOinitCond;
} else if ( (ckt->CKTmode & MODEINITJCT) && here->DIOoff) {
    vd=0;
} else if ( ckt->CKTmode & MODEINITJCT) {
    vd=here->DIOtVcrit;
} else if ( ckt->CKTmode & MODEINITFIX && here->DIOoff) {
    vd=0;
} else {
    #ifndef PREDICTOR
    if (ckt->CKTmode & MODEINITPRED) {
```
Glimpse: Diode Model in MAPP

MOD.terminals
MOD.parms
MOD.explicit_outs
MOD.f: function handle
MOD.q: function handle

- executable (in Matlab)
- takes 10min to write
- works in all analyses
Glimpse: Shooting Method in MAPP

Shooting Algorithm in MAPP (pseudo-code)

```plaintext
shootObj = shoot(DAE);  // constructor
1: shootObj.DAE = DAE;
2: shootObj.tranObj = LMS(DAE); // transient simulation object
3: set up member functions: .solve, .g, and .J
4: return shootObj;

shootObj.solve (initguess, T):
1: x0 ← NR(@g, @J, initguess);
2: shootSols = tranObj.solve(x0, 0, T);
3: return shootSols;

shootObj.g (x0):
1: tranSols = tranObj.solve(x0, 0, T);
2: return gout = tranSols(:, n) - x0;

shootObj.J (x0):
1: tranSols = tranObj.solve(x0, 0, T);
2: Ci_pre = DAE.dq_dx(x0);
3: M = eye(n);
4: for i = 2:n do
5: x = tranSols(:, i); u = inputs(:, i);
6: Ci = DAE.dq_dx(x); Gi = DAE.df_dx(x, u);
7: M = (Ci + (tpts(i) - tpts(i-1)) * Gi) \ Ci_pre * M;
8: Ci_pre = Ci;
9: end for
10: return Jout = M - eye(n);
```

- object-oriented
- reuses LMS (transient) code
- 150 lines of code
- works with all devices, circuits, domains
- a pleasure to write (you too can do it)
Code Structuring of MAPP

Ckt structure descriptions (w behavioural extensions)

Device model equations

User commands

Equation Engine (eg, MNA, Tableau, Multi-physics, etc.)

Differential Algebraic Equation (DAE)

$$\frac{d}{dt} q(\bar{x}(t)) + f(\bar{x}(t), \bar{u}(t)) = 0$$

User commands

Analysis algorithms: DC, AC, transient, HB, shooting, envelope, multi-time, noise, macromodelling, etc.

Analysis outputs: system-level structures, macromodels

Analysis results: waveforms, files, tables

(arrows mean “needed for”)
MAPP: Compact Model Prototyping

(a) BSIM6.1.0
default: \( L=10\ \mu m, \ W=10\ \mu m \)

(b) PSP Level 103 v3.0
default: \( L=10\ \mu m, \ W=10\ \mu m \)

(c) MVS v1.0.1
default: \( L=80\ \text{nm}, \ W=1\ \mu m \)

(d) MOS11 v2
default: \( L=1\ \mu m, \ W=1\ \mu m \)
MAPP: Multi-Physics Support

Optical
Network Interface Layer
- electric fields, polarizations,
  modes, wavelengths,
  wave continuity, ...

Electrical
Network Interface Layer
- node voltages, branch currents,
  KCL, KVL

Mechanical
NIL

Spintronic
NIL

Biochemical
NIL

Thermal
NIL

ModSpec Core (Equations)
Optical System Modelling/Simulation Example

- $E_{in}$
- light source
- wave guide
- splitter
- wave guide
- joiner
- wave guide
- light sink

sweep $n$ (reflective index)

$|E_{out}|$ vs $n$ with $E_{in} = 1$, $l = 100\mu m$

$|E_{out}|$

$n$ of waveguide

- $\lambda = 1550\text{nm}$
- $\lambda = 1625\text{nm}$
LTI MOR Example in MAPP

Equation formulation for the RC line circuit

Extract C/G matrices, Arnoldi MOR

Equation formulation for the reduced-order system

AC simulation

AC analysis: RC line with 20 segments: line end voltages with and without MOR

n-stage

line end voltage
Phase-based Reduced-order Model in MAPP

Model osc. in MAPP

Simulate PSS: shooting or HB

Extract PPVs

Gen-Adler Analysis

\[
\frac{d}{dt} \Delta \phi(t) = f_0 - f_1 + f_0 \cdot g(\Delta \phi(t))
\]

DC locking range

TRAN phase dynamics

Observe Injection Locking properties

Standard TRAN simulation:

Phase-based TRAN:

\[\Delta \phi(t)\text{ captures phase response nicely}\]

Simulation Algorithms in MAPP: More Examples

MAPP: First Public Release

- **Open Source download:** [http://mapp.eecs.berkeley.edu](http://mapp.eecs.berkeley.edu)
  - mailing list (MAPP announcements/discussion)
  - bug reporting and tracking site
  - git repository access (you can contribute)

- **License**
  - primary: GPL-v3
  - alternative licensing available
    - eg, SRC contract terms apply for SRC company use
  - contributors can specify their own alternative licensing terms for their contributions
MAPP: Features

- **Works entirely in MATLAB**
  - C++ version to be released
- **Help system** (start with help MAPP)
  - quick start walk-through
- **Automatic differentiation** (vecvalder)
  - help MAPPautodiff
- **Executable device specification** (ModSpec)
  - examples, tutorial: part of help
- **DC, AC, transient** analyses
  - also noise, homotopy, HB, shooting, PPV, MOR, etc. (not released yet)
- **Automated testing system** exercising suite of tests
MAPP: Intended Uses

• Developing simulation-ready device models
  » including multi-physics devices, network connectivity

• Quickly prototyping new simulation algorithms
  » hours/days to implement a new analysis
    – assess strengths/limitations before investing resources to implement in “real simulators”

• Learning or teaching modelling/simulation
  » MATLAB → broadly accessible
  » help system, tutorials, supporting resources
Summary

Ckt structure descriptions (w behavioural extensions)  
Device model equations  
User commands

Differential Algebraic Equation (DAE)  
\[ \frac{d}{dt} \mathbf{q}(\mathbf{x}(t)) + \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) = \mathbf{0} \]

Analysis algorithms: DC, AC, transient, HB, shooting, envelope, multi-time, noise, macromodelling, etc.

Analysis outputs: system-level structures, macromodels

Analysis results: waveforms, files, tables

(arrows mean “needed for”)

AC analysis: RC line with 20 segments: line end voltages with and without MOR

Optical Network Interface Layer  
Electrical Network Interface Layer

ModSpec Core (Equations)  
Mechanical NIL  
Biological NIL  
Sensory NIL

Networks with polarizations, nodes, waveguides, neural connections

http://MAPP.eecs.berkeley.edu