MAPP: The Berkeley Model and Algorithm Prototyping Platform

Tianshi Wang, A. Gokcen Mahmutoglu, Archit Gupta, Aadithya V. Karthik*, Jaijeet Roychowdhury

EECS Department
University of California, Berkeley

*Sandia National Laboratories
Albuquerque, NM
Compact Modelling

EDA: Electronic Design Automation

Analysis/Modelling for Individual Devices ("device simulation")

- Provides detailed information about device operation & characteristics
- **Computationally intensive**
  - EM simulation, drift-diffusion eqns., numerical solution of PDEs, etc.

**Compact Model** of Device

- **Simple** enough to be incorporated in circuit simulators
- **Accurate** enough to have predictive value for circuits
- **Terminal behaviour** important
  - internal details less important
- **Purpose**: use in circuit design
  - via **circuit simulation**

T. Wang, UC Berkeley
Compact Modelling

"device-level" simulation of a capacitor (eg, finite element electrostatic)

\[
q(v) = Cv \\
i(t) = \frac{d}{dt} q(v(t))
\]

geometrical parameters

electrical parameters

Compact Model

- Schichman-Hodges (core)

\[
I_{ds} = f(V_{gs}, V_{ds}) = \begin{cases} 
\beta [ (V_{gs} - V_T) - \frac{V_{ds}}{2} ] V_{ds}, & \text{if } V_{ds} < V_{gs} - V_T \\
\frac{\beta}{2} (V_{gs} - V_T)^2, & \text{if } V_{ds} \geq V_{gs} - V_T \\
0, & \text{if } V_{gs} < V_T
\end{cases}
\]

- BSIM/EKV/PSP/MVS...

T. Wang, UC Berkeley
Simulation Algorithms

- DC operating point and DC sweep
- small-signal AC
- transient: FE, BE, TRAP, LMS, GEAR, …
- PSS (periodic steady-state): HB, shooting
- noise analyses
- sensitivity analyses
- distortion analyses
- stochastic and statistical methods
- macro-modelling, MOR, “analog verification”, …
Modelling and Simulation Today

• motivation for MAPP

<table>
<thead>
<tr>
<th>develop good compact models</th>
<th>prototype simulation algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td>many pitfalls:</td>
<td>you will need:</td>
</tr>
<tr>
<td></td>
<td>• parsing, equation formulation, output, …</td>
</tr>
<tr>
<td></td>
<td>• huge (waste of) effort of re-development of basic capabilities</td>
</tr>
</tbody>
</table>

A common, open-source simulation framework in MATLAB

• hard to debug or resolve

compact model developers and simulation people blame each other
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 `dioload.c` (SPICE3)

```c
#include SENSDEBUG
   printf("vd = %.7e \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 {
#endif PREDICTOR
   if (ckt->CKTmode & MODEINITPRED) {
```
Glimpse: Diode Model in MAPP

- executable (in Matlab)
- takes 10min to write
- works in all analyses

```
function MOD = diodeCapacitor_ModSpec_wrappe() % ModSpec description of an ideal diode in parallel with a capacitor
MOD = ee_model();
MOD = add_to_eo_model(MOD, 'external_nodes', {'p', 'n'});
MOD = add_to_eo_model(MOD, 'explicit_outs', {'ipn'});
MOD = add_to_eo_model(MOD, 'pars', {'C', 2e-12, 'Is', 1e-12, 'VT', 0.025});
MOD = add_to_eo_model(MOD, 'f', @f);
MOD = add_to_eo_model(MOD, 'q', @q);

end

function out = f(S)
    v2struct(S);
    out = Is*(exp(vpn/VT)-1);
end

function out = q(S)
    v2struct(S);
    out = C*vpn;
end
```

MOD.terminals
MOD.parms
MOD.explicit_outs
MOD.f: function handle
MOD.q: function handle
...
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) → Equation Engine (eg, MNA, Tableau, Multi-physics, etc.) → Differential Algebraic Equation (DAE)

\[ \frac{d}{dt} \vec{q}(\vec{x}(t)) + \vec{f}(\vec{x}(t), \vec{u}(t)) = \vec{0} \]

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

Analysis outputs: system-level structures, macromodels

Analysis results: waveforms, files, tables

User commands

(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: $L=80 nm$, $W=1 \mu m$

(d) MOS11 v2
default: $L=1 \mu m$, $W=1 \mu m$
MAPP: Multiphysics 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

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

Parameter sweep $n$ (reflective index)

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

- Two curves for different wavelengths:
  - $\lambda = 1550nm$
  - $\lambda = 1625nm$
Multiphysics Systems

potential/flow systems:

kinematic NIL:
“flow”: force
“potential”: position

magnetic NIL:
“flow”: magnetic flux
“potential”: magnetomotive force

thermal NIL:
“flow”: power flow
“potential”: temperature

Spintronic systems:

vectorized spin currents
vectorized spin voltages

Chemical reaction networks

rates and concentrations

“KCLs” at nodes have d/dt terms

Kerem Yunus Camsari; Samiran Ganguly;
Supriyo Datta (2013), “Modular Spintronics Library,”
https://nanohub.org/resources/17831.
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 simulation

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

- Line end voltage
- Arnoldi MOR: q=10
- Arnoldi MOR: q=1
- Arnoldi MOR: q=15
- Arnoldi MOR: q=3
- Arnoldi MOR: q=20
- Arnoldi MOR: q=5

Magnitude

Phase (degrees)

Frequency (Hz)
Homotopy Analysis on Goto Pair

\[ V_{dd} \]

\[ \text{diode current} = f(\text{diode voltage}) \]

\[ V_{dd} - v_1 \]

\[ + \]

\[ - \]

\[ + \]

\[ - \]

\[ v_1 \]

\[ n1 \]

\[ n2 \]

\[ \text{currents} \]

\[ \text{unstable} \]

\[ \text{stable} \]
Finding Folds with Homotopy

\[ i_{pn} = f_1(v_{pn}, s) \]
\[ \frac{d}{dt}s = f_2(v_{pn}, s) \]

**memristor**
(RRAM, CBRAM, PCM...)

**NC-FET**

**ESD snapback**
Phase-macromodel Simulation 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 TR AN

locking range phase dynamics

Observe Injection Locking properties

\( \Delta \phi(t) \) captures phase response nicely

details: Bhan s a l i / R o y c h o w d h u r y, "Gen-Adler: the Generalized Adler's equation for injection locking analysis in oscillators". Proc. ASPDAC, 2009.
Simulation Algorithms in MAPP: More Examples

Distortion Contribution Analysis on Gilbert cell

MAPP: Public Release

- Open Source download: [https://github.com/jaijeet/MAPP](https://github.com/jaijeet/MAPP)

- 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/Octave
  - C++ version to be released
  - mex interfaces to link C++ devices and circuit DAEs into MATLAB
- Help system (start with `help MAPP`)
  - quick start walk-through
- Automatic differentiation (vecvalder)
  - `help MAPPautodiff`
- Executable multiphysics device specification (ModSpec)
  - examples, tutorial: part of help
- DC, AC, transient analyses
  - also noise, homotopy, HB, shooting, PPV, MOR, etc.
    (initial version released at PHLOGON.eecs.berkeley.edu)
- Automated testing system exercising suite of tests
MAPP: Intended Uses

- **Developing simulation-ready device models**
  - including multiphysics 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

https://github.com/jaijeet/MAPP