MAPP: The Berkeley Model and Algorithm Prototyping Platform

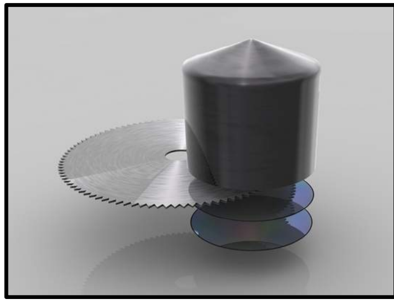
Tianshi Wang, A. Gokcen Mahmutoglu,
Archit Gupta, Aadithya V. Karthik*,
Jaijeet Roychowdhury

EECS Department
University of California, Berkeley

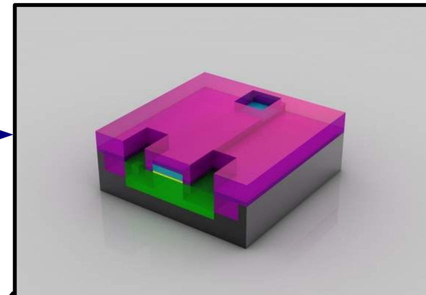
*Sandia National Laboratories
Albuquerque, NM

Compact Modelling

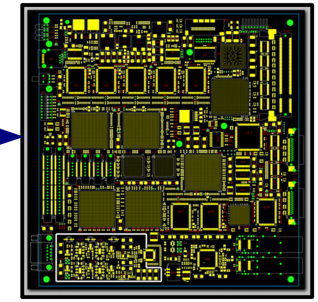
EDA: Electronic Design Automation



Process



Device



Circuit

<u>Analysis/Modelling for Individual Devices</u>
("device simulation")

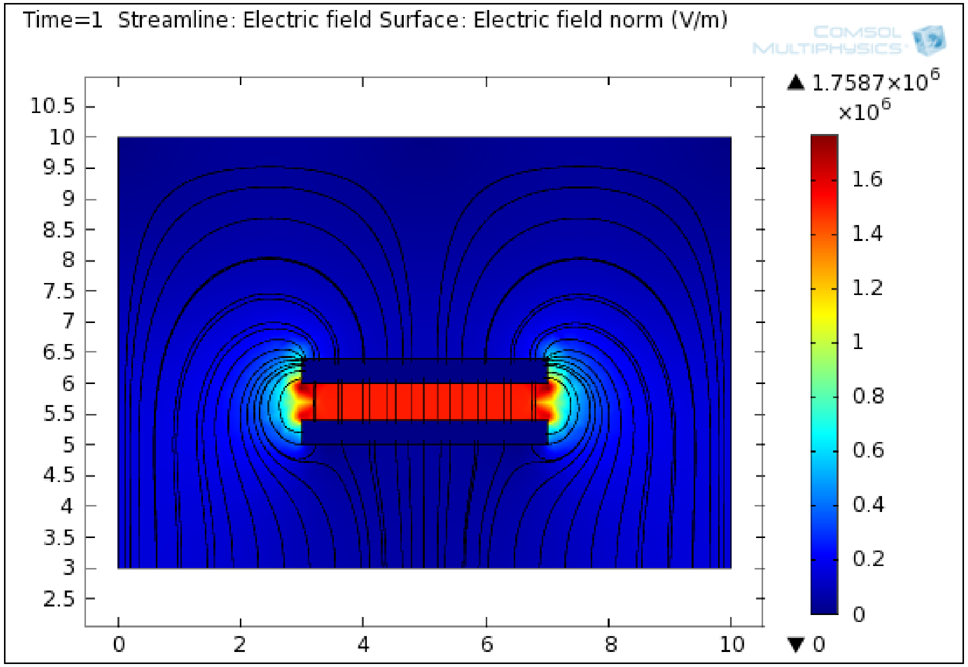
- Provides detailed information about device operation & characteristics
- <u>Computationally intensive</u>
  - » EM simulation, drift-diffusion eqns., numerical solution of PDEs, etc.

<u>Compact Model</u> 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 **<u>circuit simulation</u>**

Images from <u>www.google.com/images</u>

Compact Modelling



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

voltage-current relationships

$$q(v) = Cv \qquad i(t) = \frac{d}{dt}q(v(t))$$

charge capacitance

electrical parameter

$$C = \frac{\epsilon_r \epsilon_0 L \cdot W}{d}$$

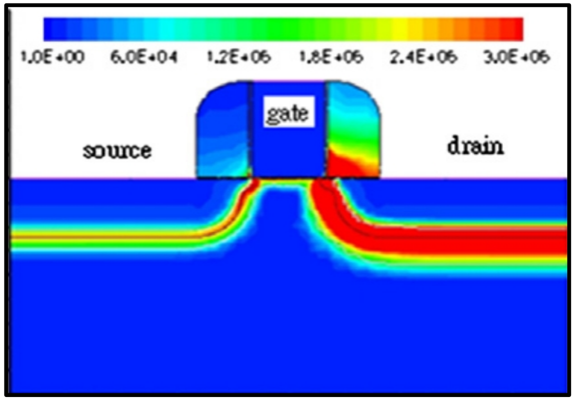
geometrical parameters

or

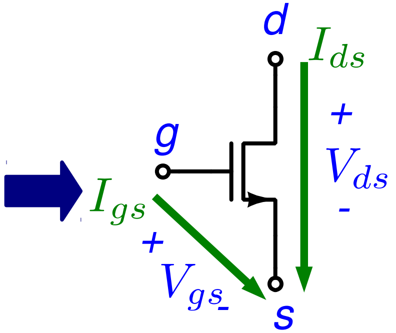
$$C = \frac{\epsilon_r \epsilon_0 L \cdot W}{d}\left[1 + \frac{d}{\pi W} + \frac{d}{\pi W}\ln\left(\frac{2\pi W}{d}\right)\right] + \cdots$$

H. B. Palmer 1927

Compact Model



S.L. Jang, J.S. Yuan, S.D. Yen, E. Kritchanchai, G.W. Huang "Experimental evaluation of hot electron reliability on differential Clapp-VCO" Microelectronics Reliability, Volume 53, Issue 2, February 2013, Pages 254–258



MOSFET

- Schichman-Hodges (core)

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

- BSIM/EKV/PSP/MVS...

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**

develop good compact models

- **many pitfalls:**
-
- hard to debug or resolve

compact model developers and simulation people blame each other

prototype simulation algorithms

you will need:

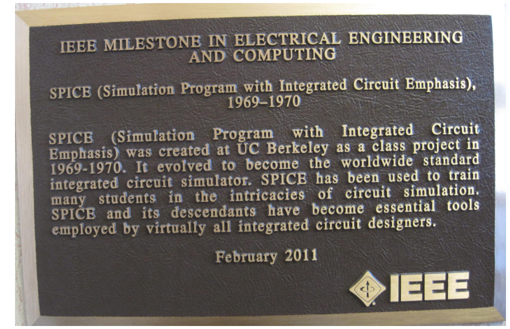
- parsing, equation formulation, output, …

huge (waste of) effort of re-development of basic capabilities

A common, open-source simulation framework

in MATLAB

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: **<u>modular, well-structured, flexible</u>**
  - » separated models, algorithms, numerics, I/Os
  - » simple, clean interfaces
  - » short, easy to read, easy to modify

Excerpt from *dioload.c* (SPICE3)

```
#ifdef 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 {
#ifndef PREDICTOR
         if (ckt->CKTmode & MODEINITPRED) {
```

Sensitivity analysis code

AC analysis code

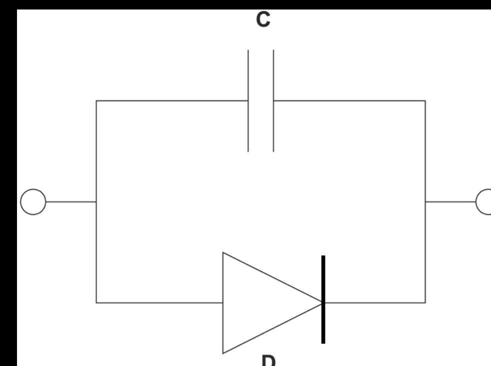
Transient analysis related code

Glimpse: Diode Model in MAPP

```
function MOD = diodeCapacitor_ModSpec_wrapper()
% ModSpec description of an ideal diode in parallel with a capacitor
   MOD = ee_model();
   MOD = add_to_ee_model(MOD, 'external_nodes', {'p', 'n'});
   MOD = add_to_ee_model(MOD, 'explicit_outs', {'ipn'});
   MOD = add_to_ee_model(MOD, 'parms', {'C',2e-12, 'Is',1e-12, 'VT',0.025});
   MOD = add_to_ee_model(MOD, 'f', @f);
   MOD = add_to_ee_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
"diodeCapacitor_ModSpec_wrapper.m" 19L, 548C written          1,1          All
```



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)

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

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

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

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

object-oriented

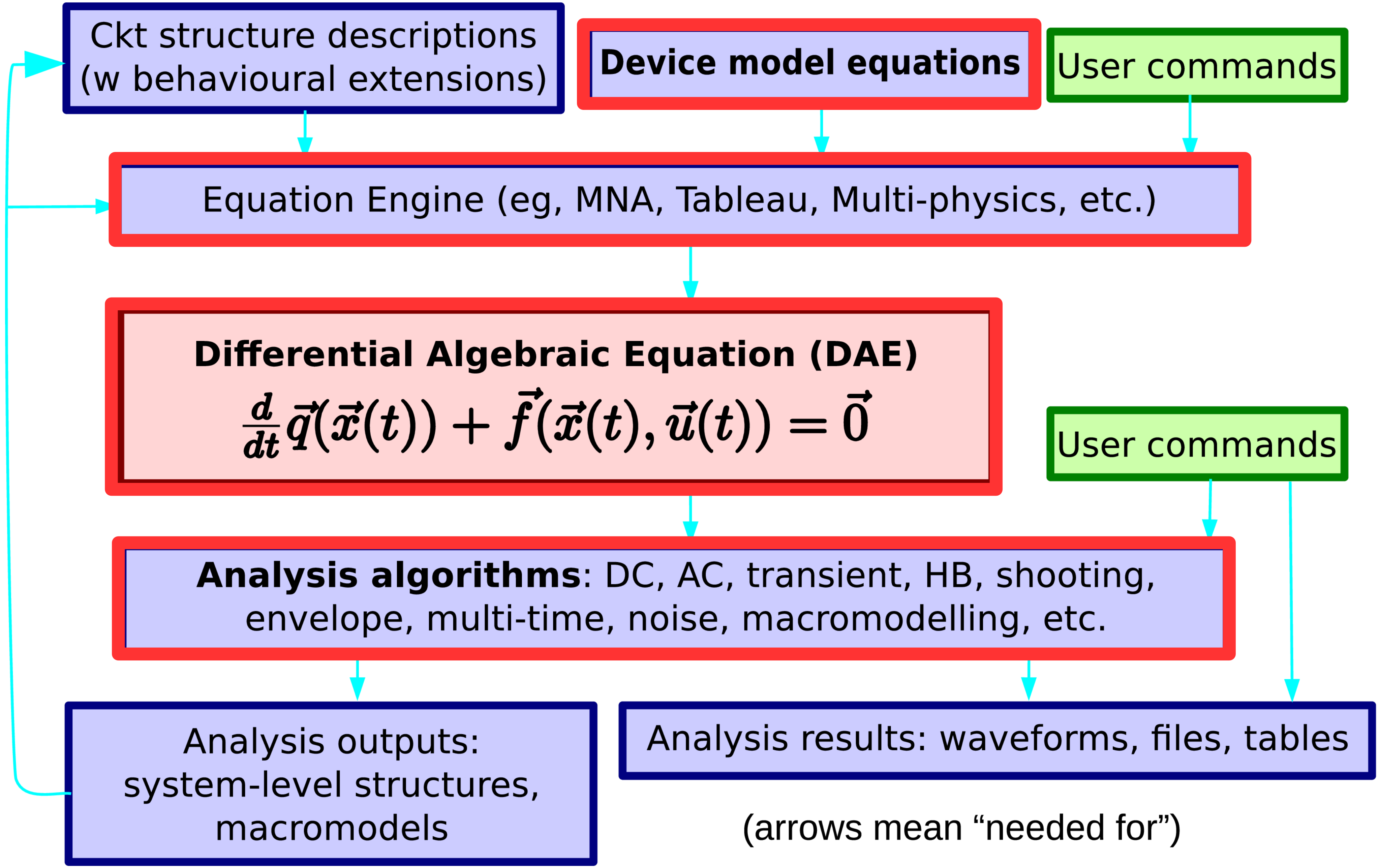
reuses LMS (transient) code

150 lines of code

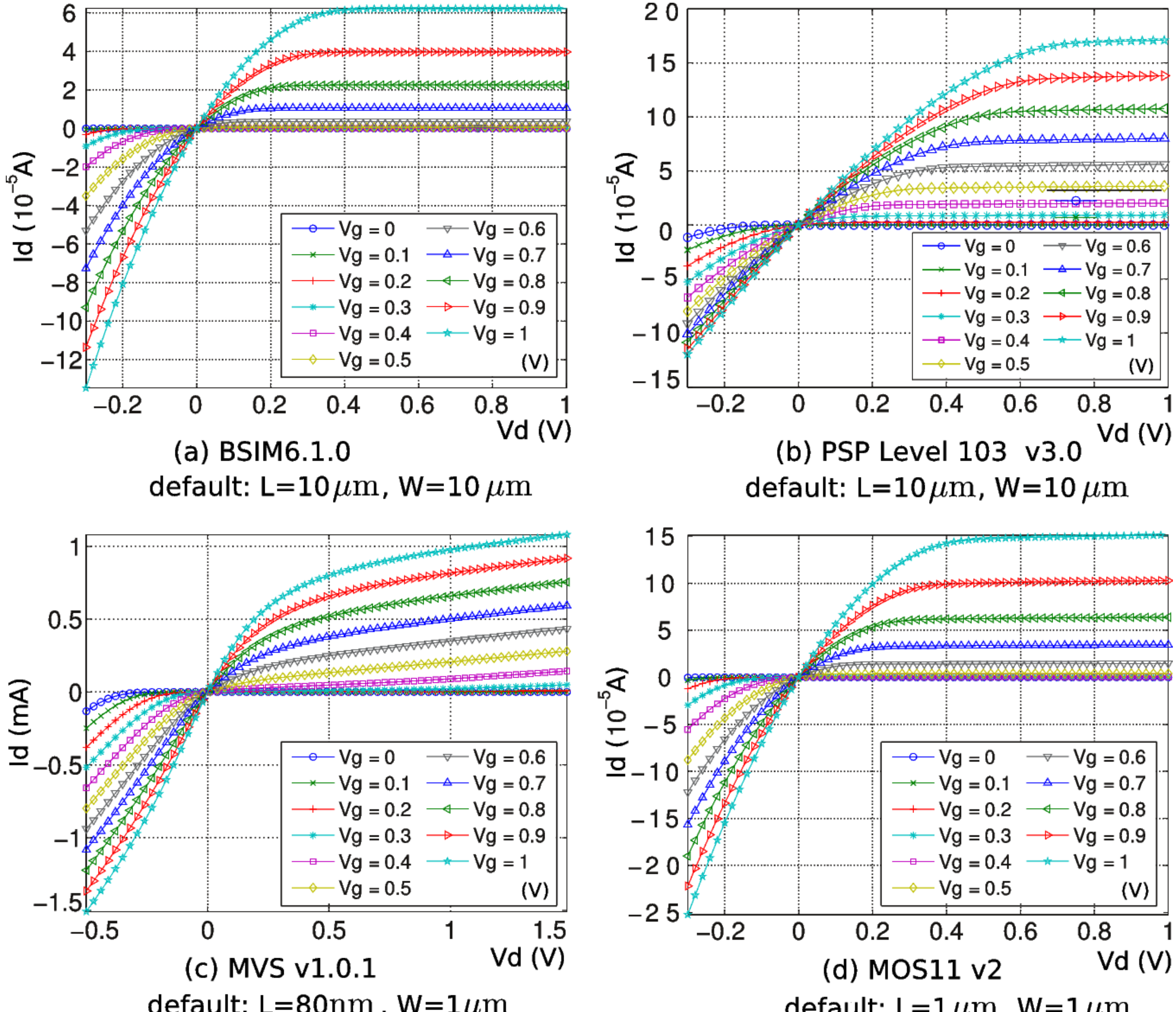
works with all devices, circuits, domains

a pleasure to write (you too can do it)

Code Structuring of MAPP



MAPP: Compact Model Prototyping



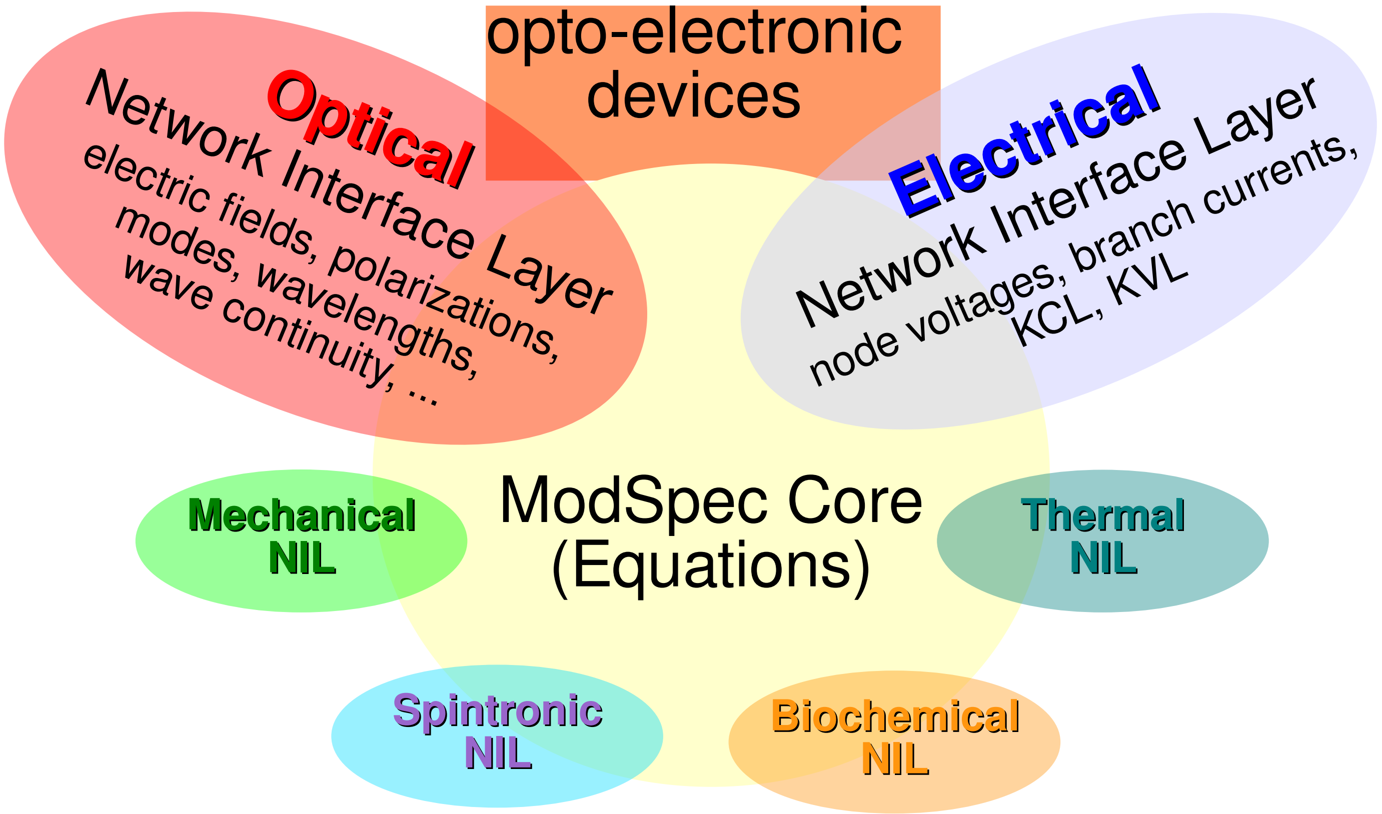
(a) BSIM6.1.0
default: L=10μm, W=10 μm

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

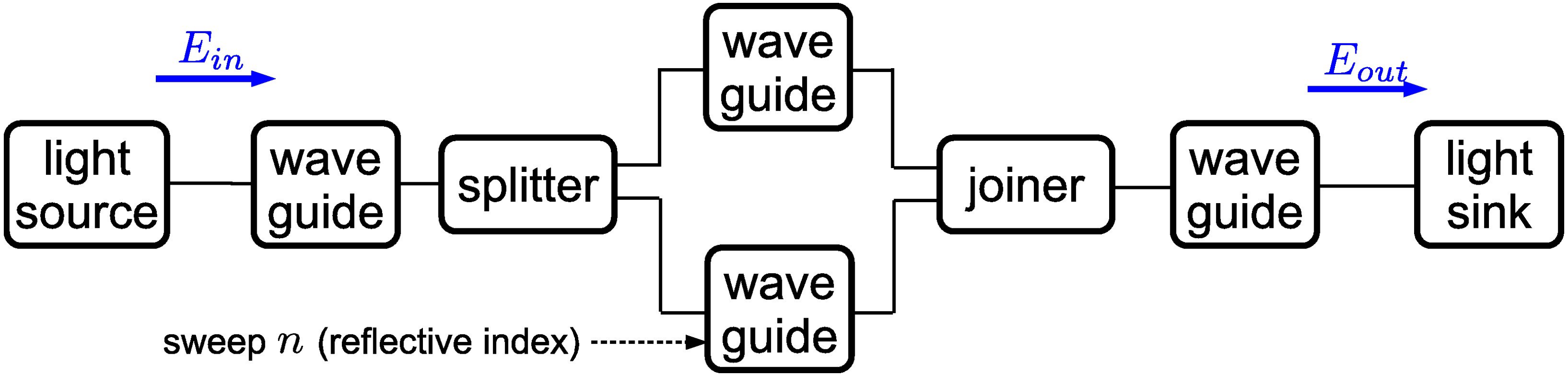
(c) MVS v1.0.1
default: L=80nm, W=1μm

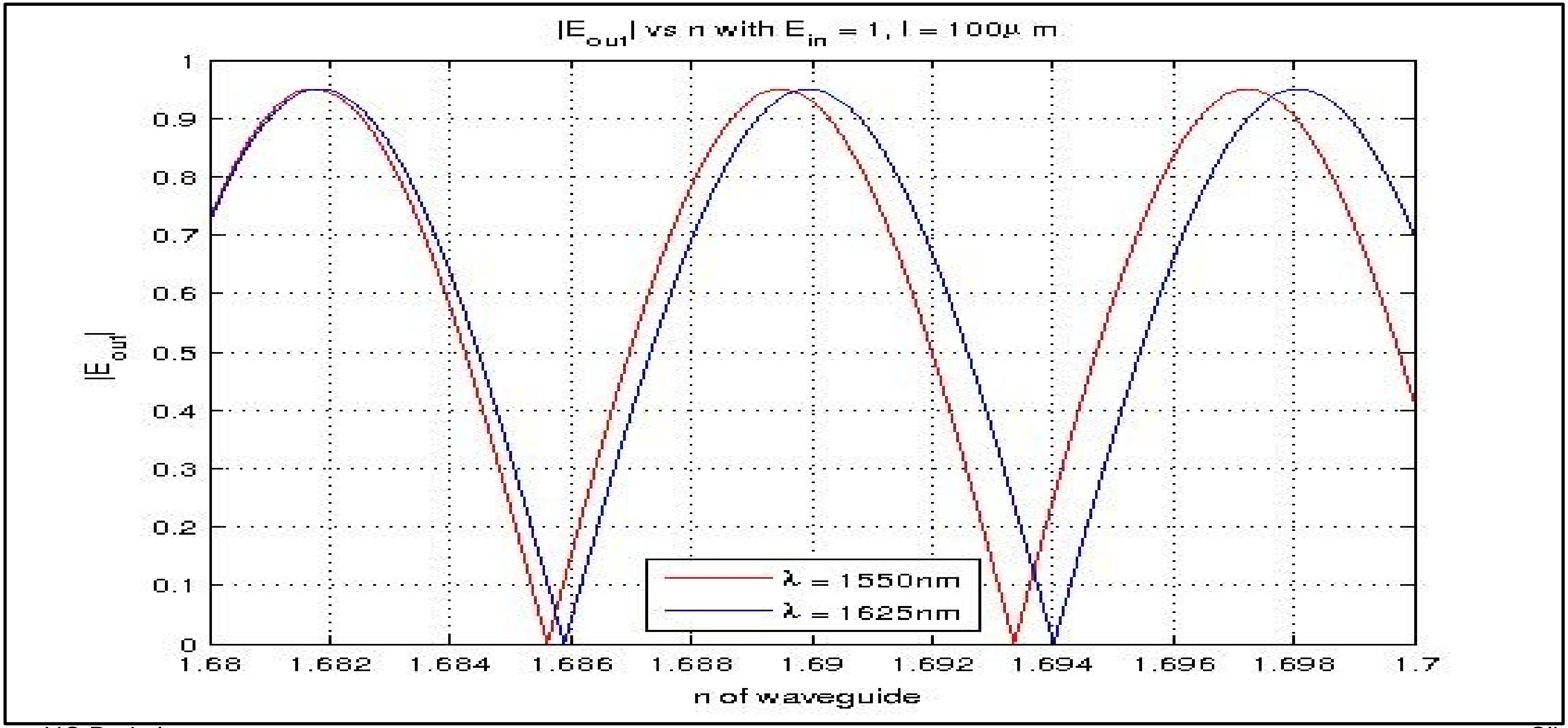
(d) MOS11 v2
default: L=1μm, W=1μm

MAPP: Multiphysics Support



Optical System Modelling/Simulation Example

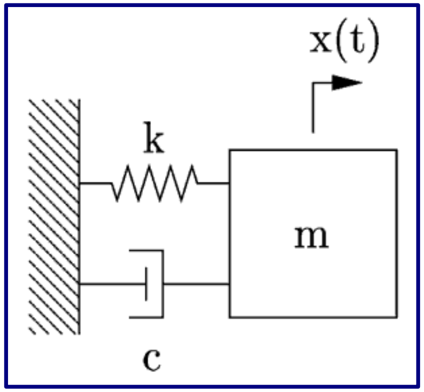


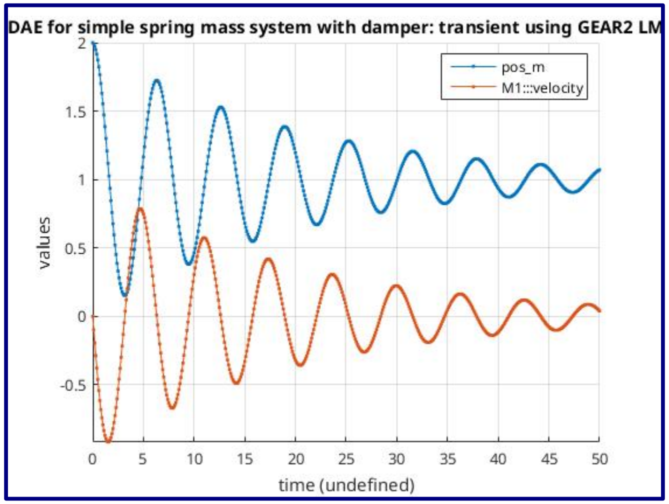


Multiphysics Systems

potential/flow systems:

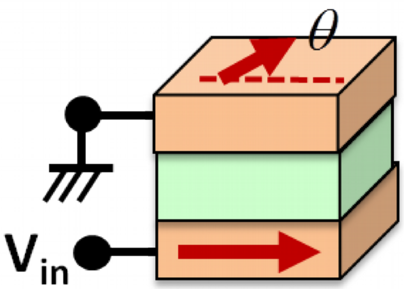
kinematic NIL:
"flow": force
"potential": position





magnetic NIL:
"flow": magnetic flux
"potential": magnetomotive force

thermal NIL:
"flow": power flow
"potential": temperature



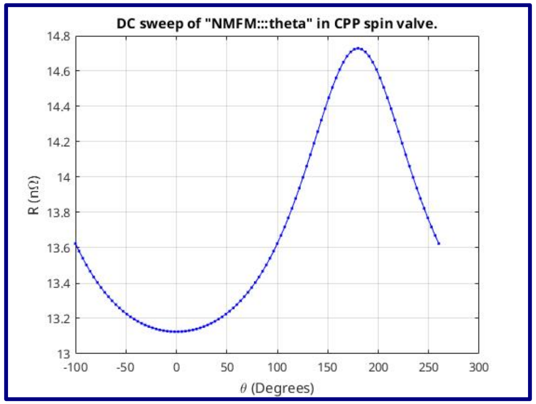
Chemical reaction networks

rates* and *concentrations

"KCLs" at nodes have d/dt terms

Spintronic systems:

vectorized spin currents
vectorized spin voltages

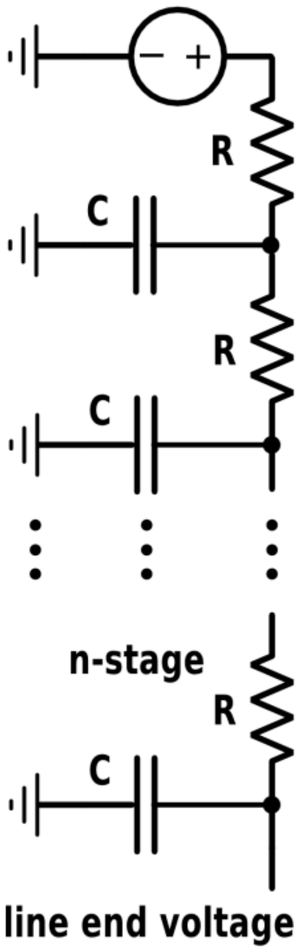


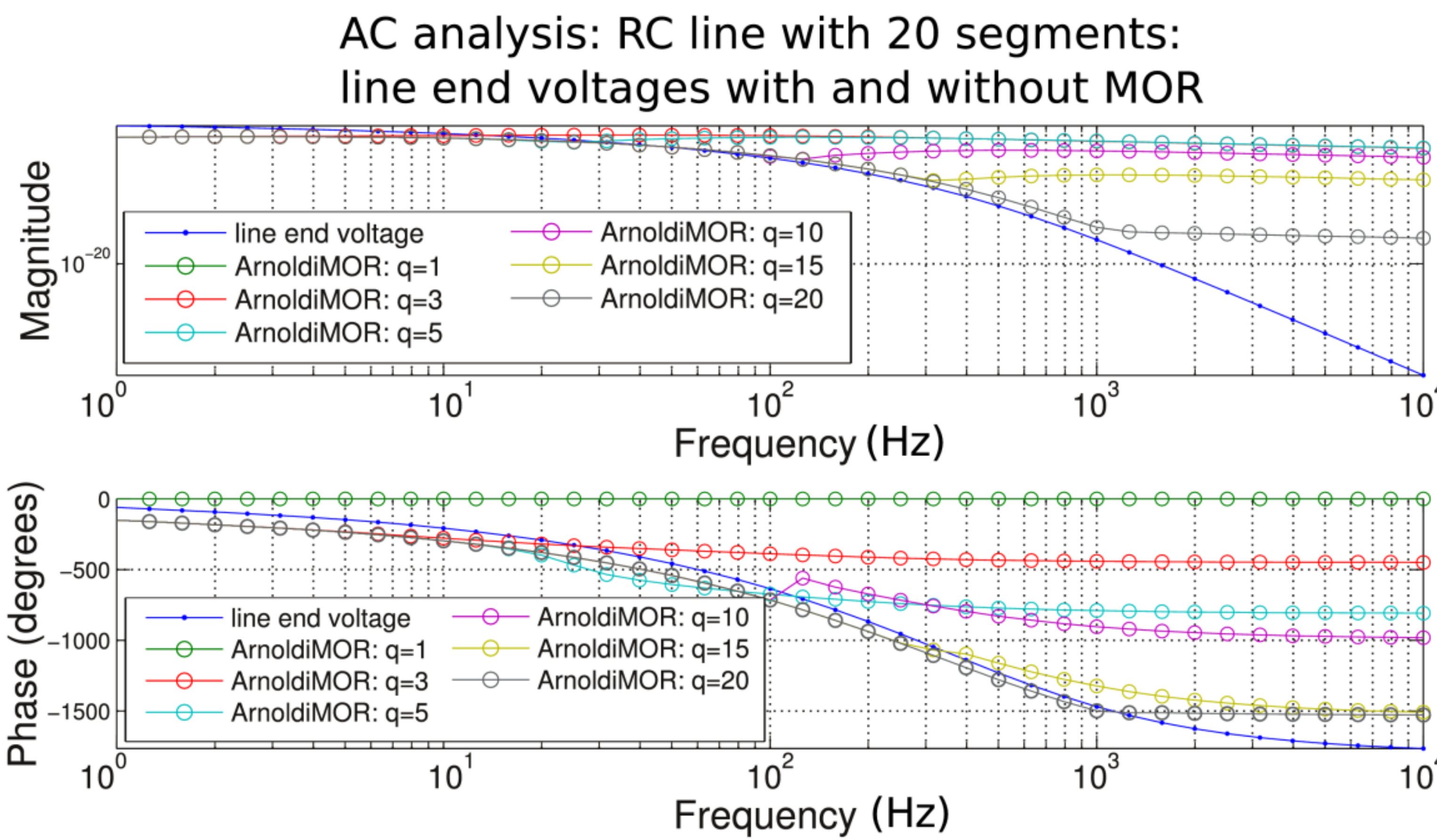
Kerem Yunus Camsari; Samiran Ganguly; Supriyo Datta (2013), "Modular Spintronics Library," https://nanohub.org/resources/17831.

LTI MOR Example in MAPP

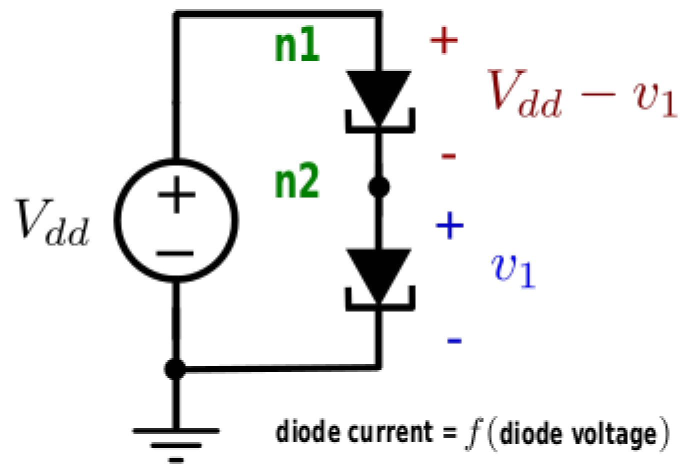
Equation formulation for the RC line circuit → **Extract C/G matrices, Arnoldi MOR** → **Equation formulation for the reduced-order system**

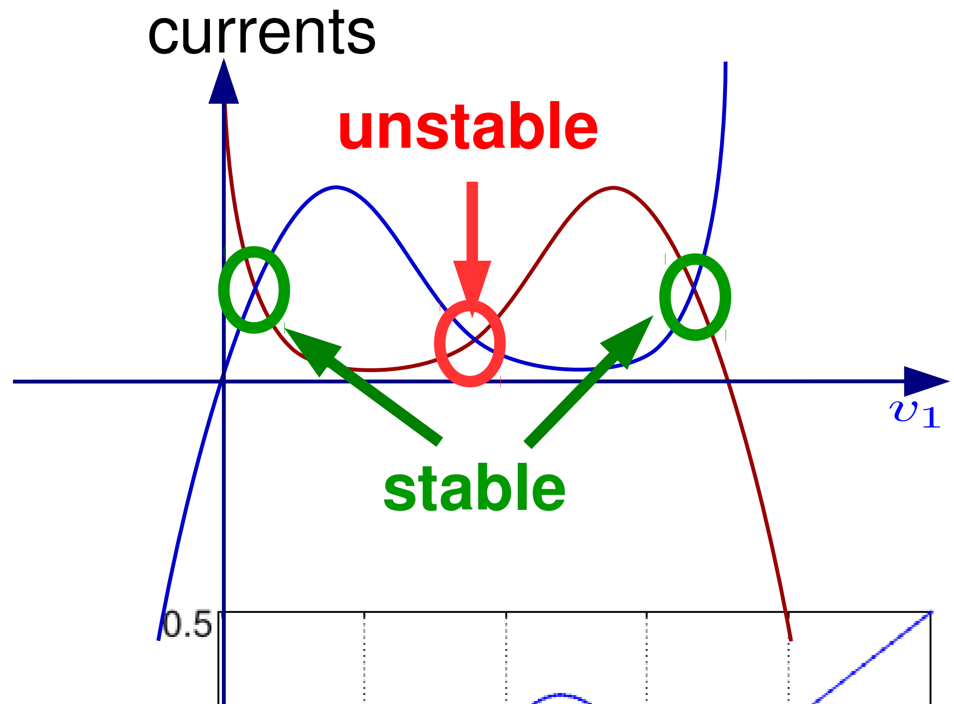
AC simulation ↔ **AC simulation**

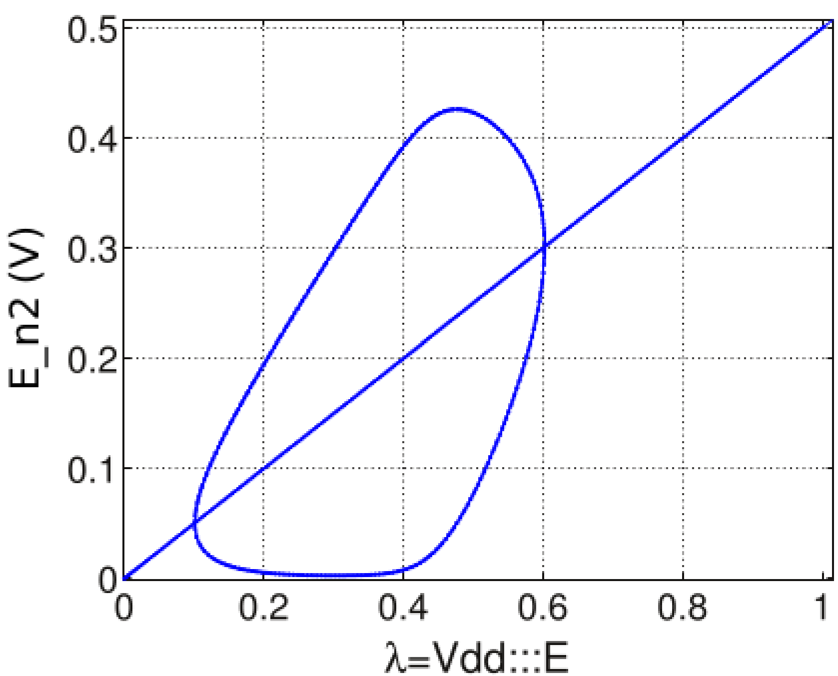


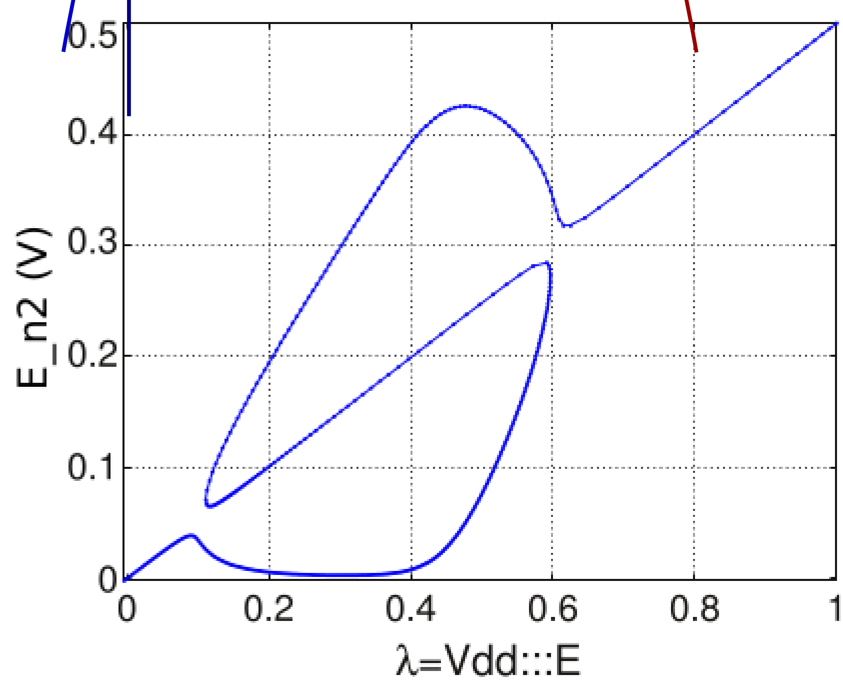


Homotopy Analysis on Goto Pair

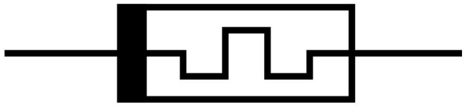






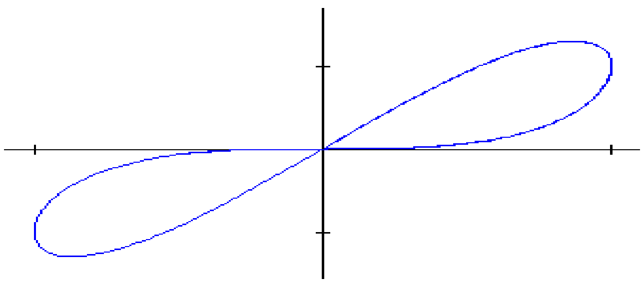


Finding Folds with Homotopy

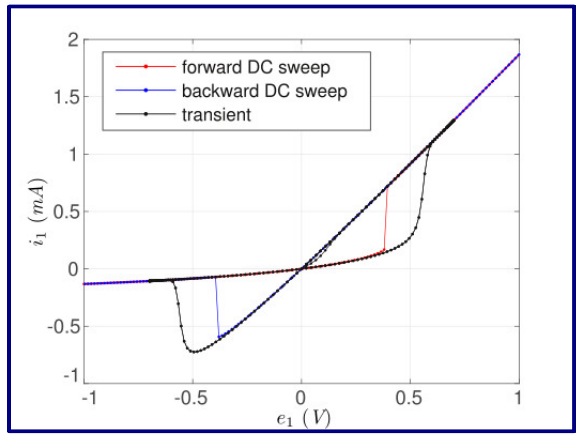


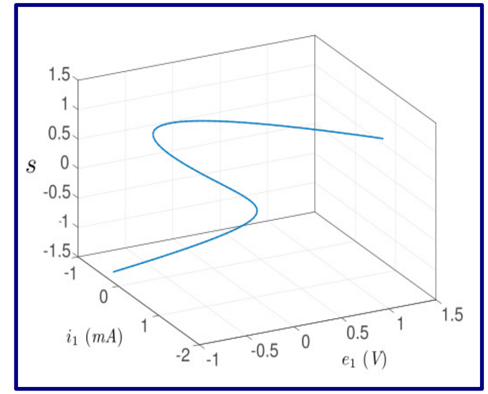
memristor

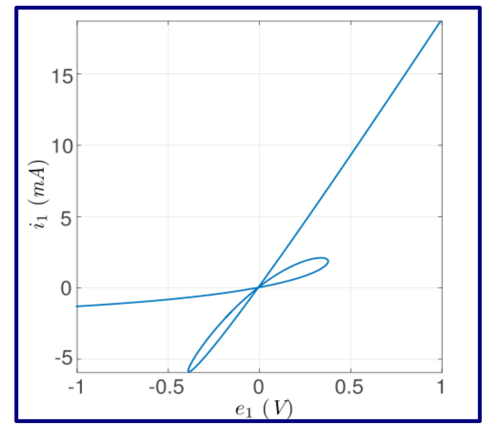
(RRAM, CBRAM, PCM...)



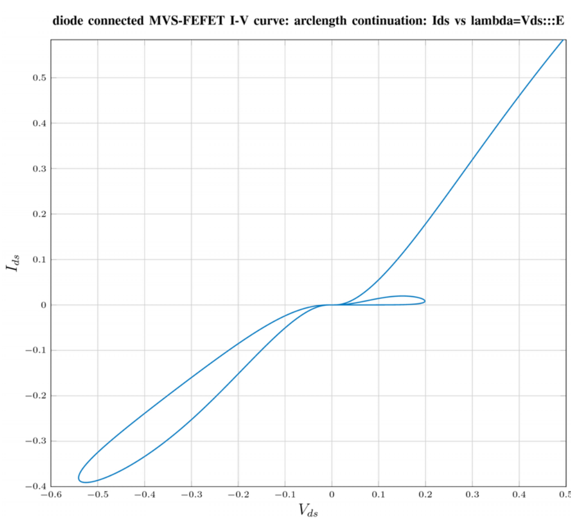
$$\color{red}{\mathbf{ipn}} = f_1\left(\color{blue}{\mathbf{vpn}},\ \mathbf{s}\right)$$
$$\frac{d}{dt}\mathbf{s} = f_2\left(\color{blue}{\mathbf{vpn}},\ \mathbf{s}\right)$$



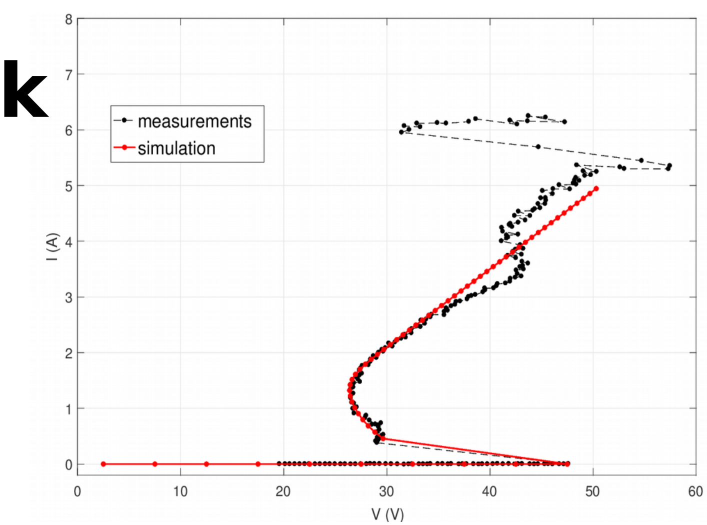




NC-FET



ESD snapback



Phase-macromodel Simulation in MAPP

Model osc. in MAPP

not just electrical ones

Simulate PSS: shooting or HB

Extract PPVs

Gen-Adler Analysis

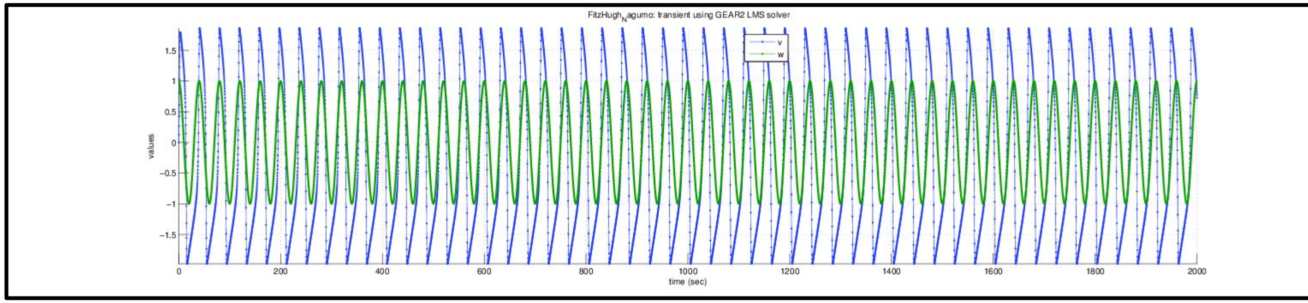
$$\frac{d}{dt}\Delta\phi(t) = f_0 - f_1 + f_0 \cdot g(\Delta\phi(t))$$

DC — **locking range**

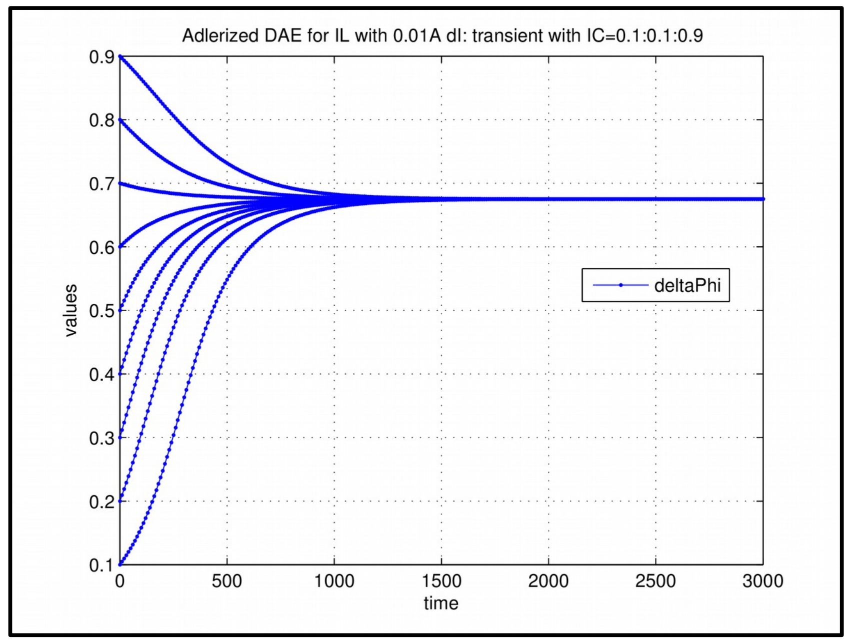
TRAN — **phase dynamics**

Observe Injection Locking properties

Standard TRAN simulation:



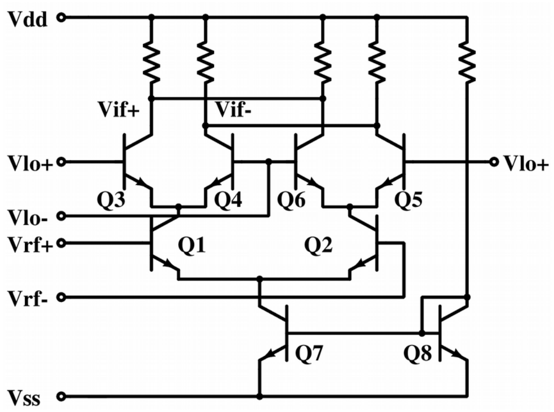
Phase-based TRAN:

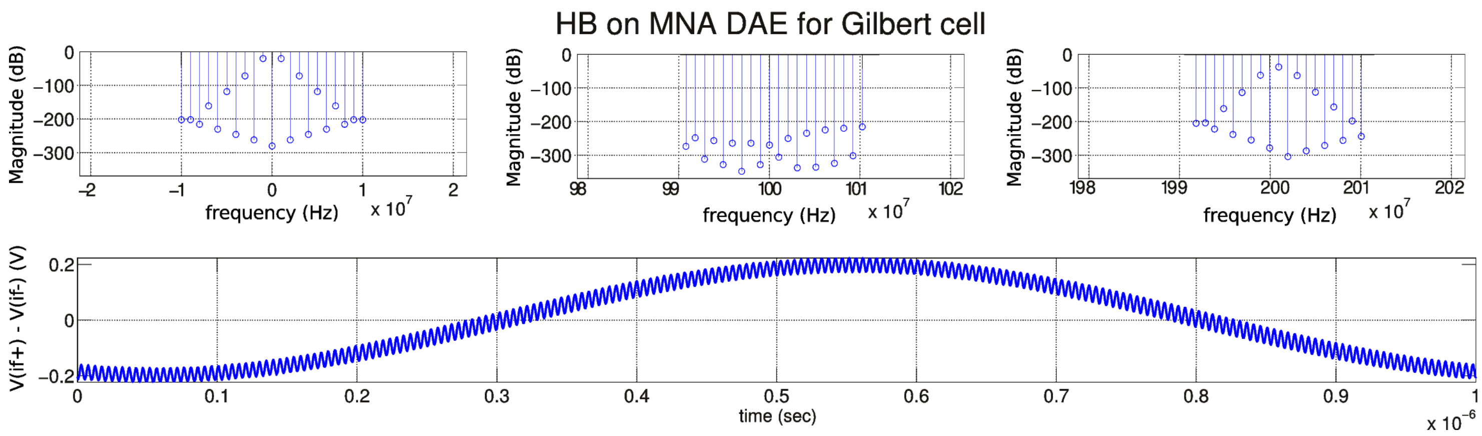


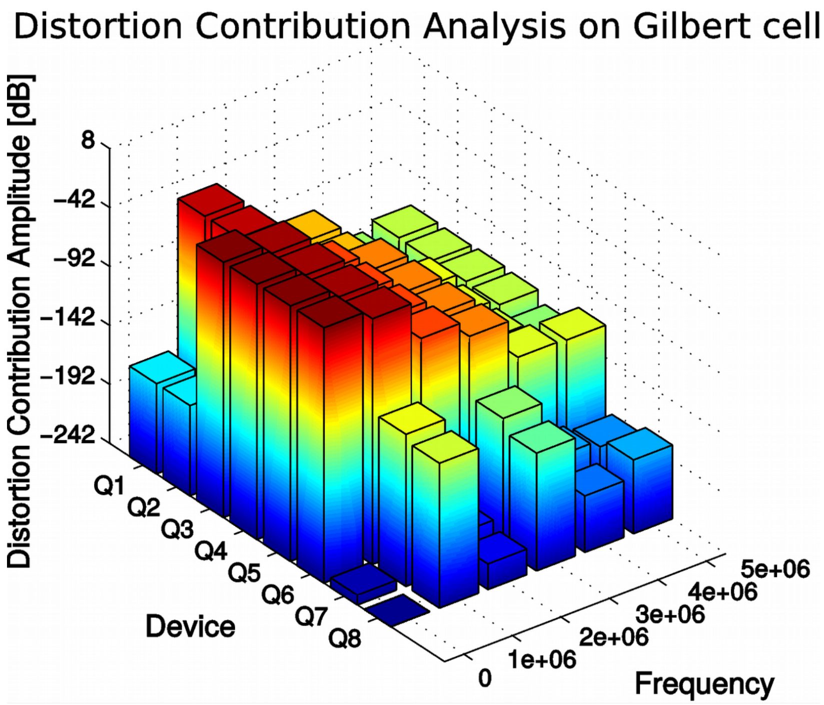
$\Delta\phi(t)$ captures phase response nicely

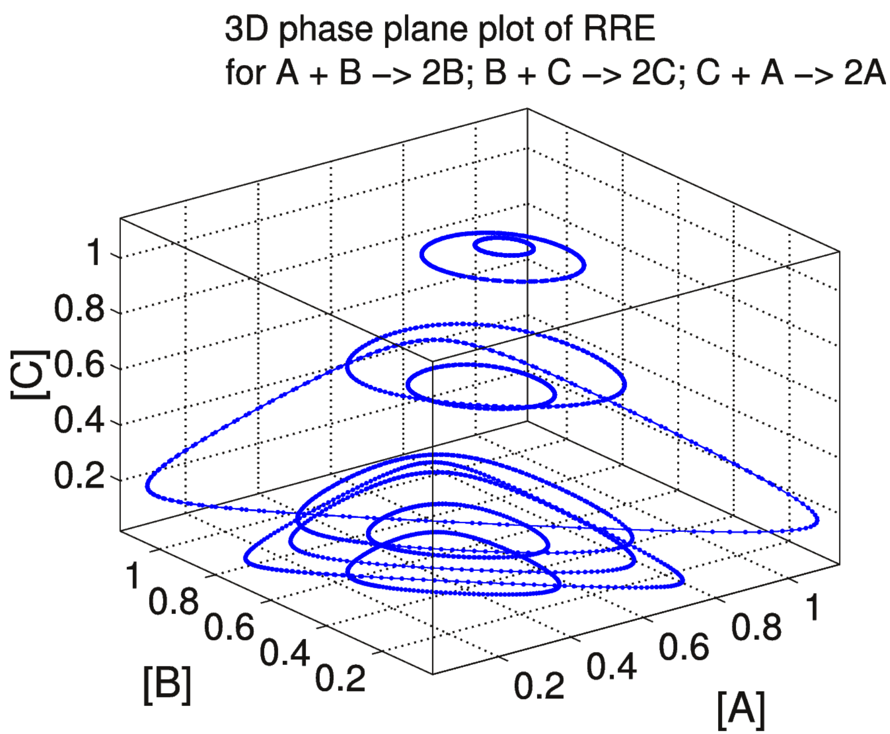
details: Bhansali/Roychowdhury, "Gen-Adler: the Generalized Adler's equation for injection locking analysis in oscillators". Proc. ASPDAC, 2009.

Simulation Algorithms in MAPP: More Examples









details: Wu/Roychowdhury, "Efficient per-element distortion contribution analysis via Harmonic Balance adjoints". Proc. CICC 2014.

MAPP: Public Release

- Open Source download: 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.

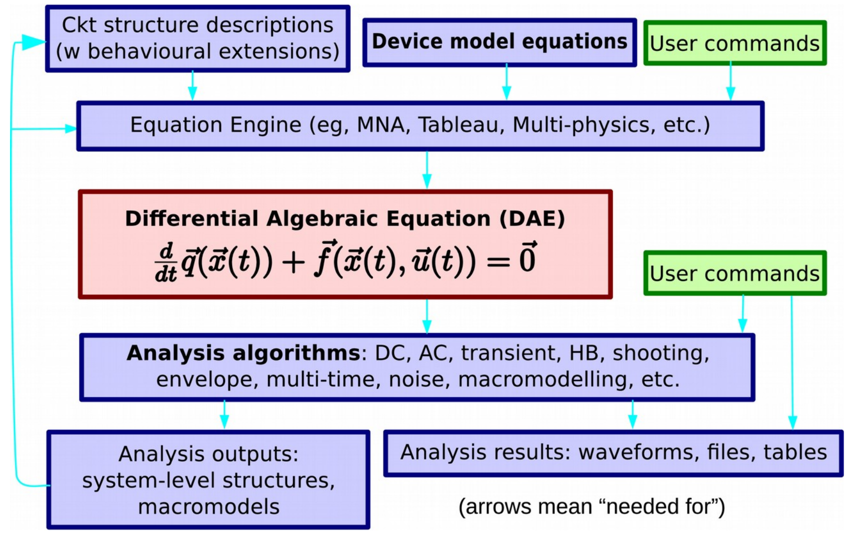
    (initial version released at PHLOGON.eecs.berkeley.edu)
- Automated testing system exercising suite of tests

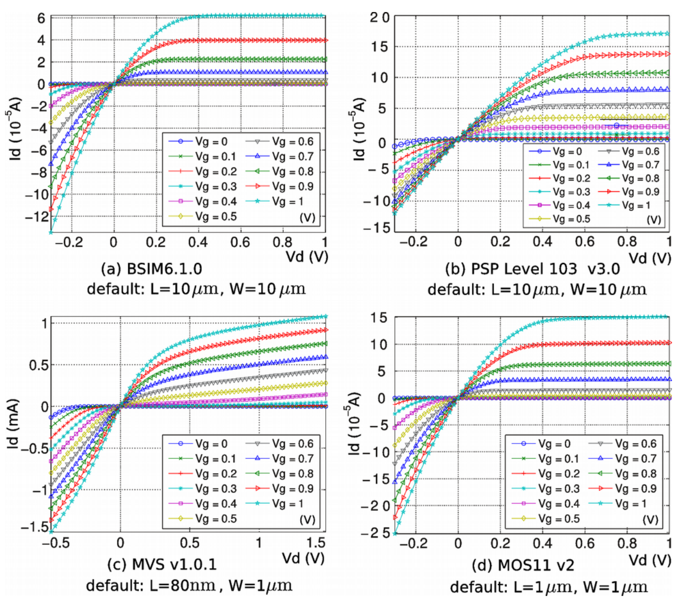
MAPP: Intended Uses

- 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

```
#ifdef 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 {
#ifndef PREDICTOR
        if (ckt->CKTmode & MODEINITPRED) {
```



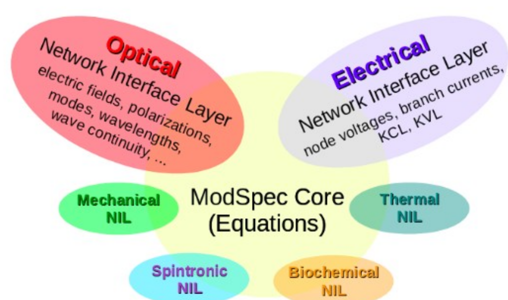


(a) BSIM6.1.0 default: L=10μm, W=10μm

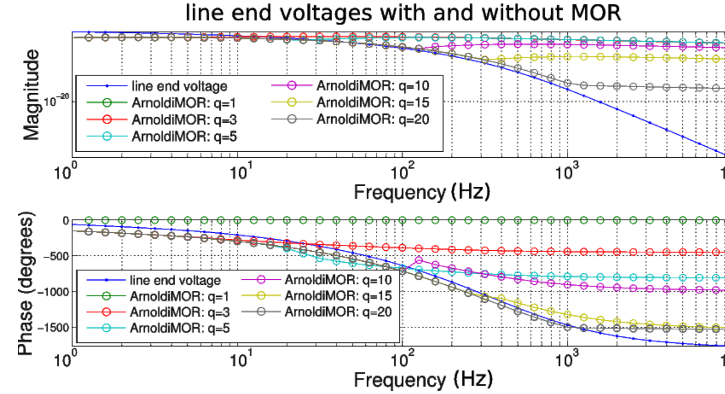
(b) PSP Level 103 v3.0 default: L=10μm, W=10μm

(c) MVS v1.0.1 default: L=80nm, W=1μm

(d) MOS11 v2 default: L=1μm, W=1μm



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



https://github.com/jaijeet/MAPP