NGSPICE: an Open Platform for Modeling and Simulation

Paolo Nenzi[1,2], Holger Vogt[1], Dietmar Warning[1], Robert Larice[1], Francesco Lannutti[2], Stefano Perticaroli[2]

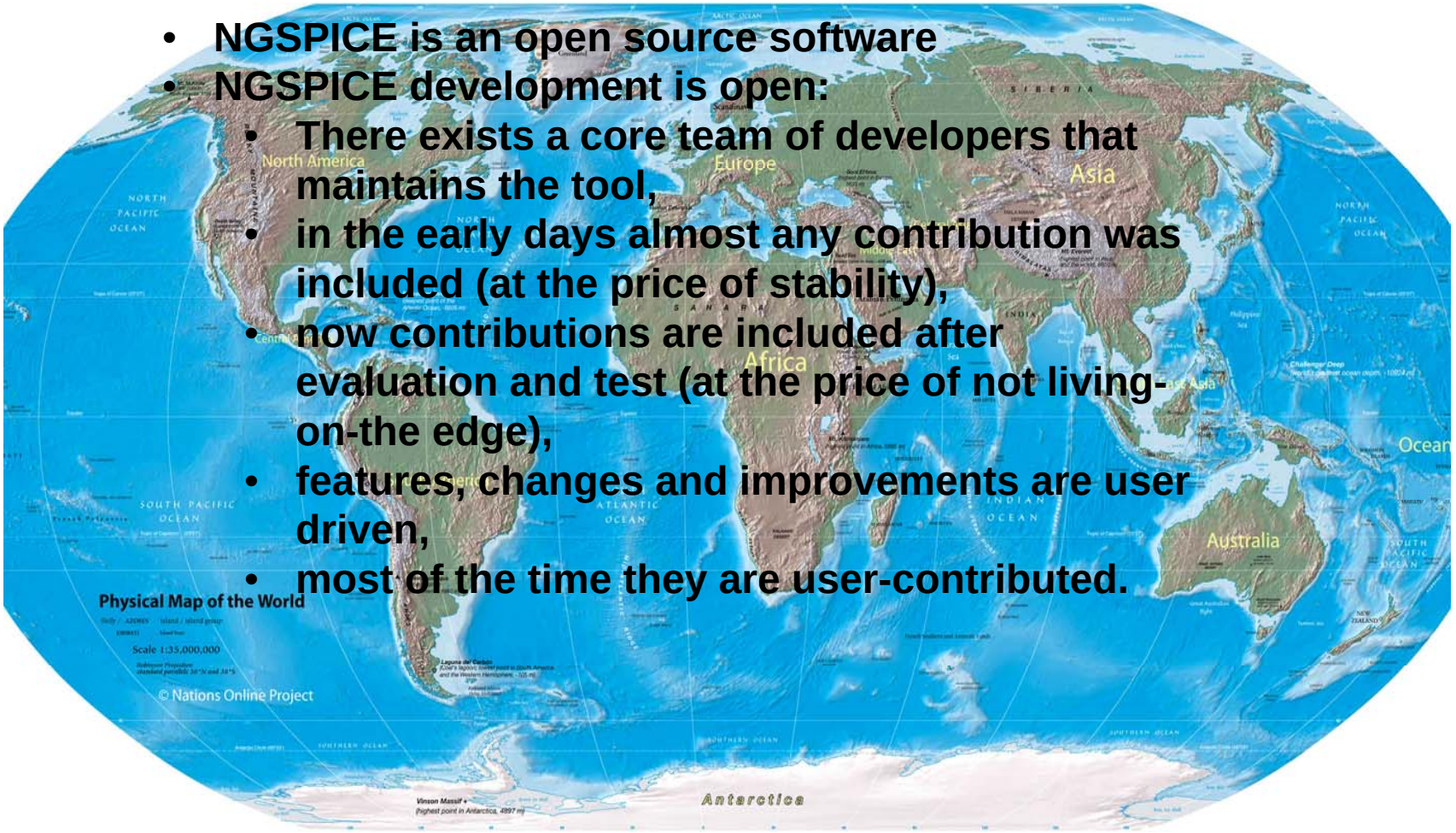
1) NGSPICE development team; 2) DIET, “Sapienza” – Università di Roma

http://www.ngspice.org/

MOS-AK/GSA Workshop, April 26th 2012 Dresden

NGSPICE development model

- NGSPICE is an open source software
- NGSPICE development is open:
  - There exists a core team of developers that maintains the tool,
  - in the early days almost any contribution was included (at the price of stability),
  - now contributions are included after evaluation and test (at the price of not living-on-the edge),
  - features, changes and improvements are user driven,
  - most of the time they are user-contributed.



Improvements in NGSPICE

Devices & Syntax

- Expression defined
  - resistance,
  - capacitance,
  - inductance.
- R,L or C can be a (non-linear) function of:
  - temperature,
  - node voltages,
  - branch currents.
- Improved “B” arbitrary source:
  - Added *time*, *temper* and *hertz* variables,
  - added ternary operator (a?b:c),
  - piecewise linear function source.
- Nonlinear VCCS and VCVS sources

Examples:

```
* Non–linear resistor
.param R0=1k Vi=1 Vt=0.5
* res. depending on control voltage V(rr)
R1 rr 0 r='V(rr) < {Vt} ? {R0} : {2*R0}'
* control voltage
V1 rr 0 PWL(0 0 100u {Vi})
```

```
* PWL arbitrary source
Bdio 1 0 I = pwl(v(A), 0,0, 33,10m,
+ 100,33m, 200,50m)
```

```
* Non linear VCVS (E element)
E41 4 0 vol = 'V(3)*V(3)–Offs'
```

```
* Non linear VCCS (G element)
G51 55 225 cur = 'V(3)*V(3)–Offs'
```

Improvements in NGSPICE

Transmission Lines

- Transmission lines:
  - Bug fixes and corrections on existing code
- KSPICE models:
  - Single Lossy Transmission Line (TXL)
  - Coupled Multiconductor Line (CPL)
  - NGSPICE supports up to 8 coupled lines

Example:

* TXL example
Y1 1 0 2 0 ymod LEN=2
.MODEL ymod txl R=12.45 L=8.972e−9 G=0
+ C=0.468e−12 length=16

* CPL example
P1 in1 in2 0 b1 b2 0 PLINE
.model PLINE CPL length={Len}
+ R=1 0 1 L={L11} {L12} {L22} G=0 0 0
+ C={C11} {C12} {C22}
.param Len=1 Rs=0
+ C11=9.143579E−11 C12=−9.78265E−12
+ C22=9.143578E−11 L11=3.83572E−7
+ L12=8.26253E−8 L22=3.83572E−7

Improvements in NGSPICE

Semiconductor devices

- Almost all device models have been modified to correct bugs.
- Some models (HFET and MES 2,3,4) comes from modeling books.

Type	Level	models
D	1	Original SPICE diode model
BJT	1	Original SPICE Gummel-Poon (SGP)
BJT	***2***	***Improved SGP model for lateral and vertical devices***
BJT	***4***	***VBIC model***
JFET	1	Original SPICE model
JFET	***2***	***Parker-Skellern model***
MES	1	Original Statz SPICE model
MES	***2,3,4***	***Models by Ytterdal***
MES	***5,6***	***HFET models***

Improvements in NGSPICE

Semiconductor devices (MOS)

- NGSPICE implements most recent industrial standard MOS models like BSIM3 and BSIM4
- EKV model has two implementations:
  - One Verilog-A 2.6 version is available in the standard distribution
  - The EPFL C-code implementation is available through NDA.
- Some old BSIM3 and BSIM4 implementations are retained for compatibility purposes (and to serve PDK's ☺)
- CMC Benchmark implemented

Type	Level	models
MOS	1	Spice3 level 1 model ("Shichman-Hodges")
MOS	2	Spice3 level 2 model ("Grove-Frohman")
MOS	3	Spice3 level 3 model
MOS	4	BSIM1
MOS	5	BSIM2
MOS	6	Spice3 level 6 model
MOS	***8***	***BSIM3 version 3.3.0 model***
MOS	9	MOS9 model (implementation by Alan Gillespie)
MOS	***10,58***	***BSIM4 SOI model version 4.3.1***
MOS	***14,54***	***BSIM4 model version 4.7.0***
MOS	***61***	***HISIM2 model version 2.5.1***
MOS	***62***	***HISIM_HV model version 1.2.2***
MOS	55	B3SOI-FD model
MOS	56	B3SOI-DD model
MOS	57	B3SOI-PD model
MOS	*44*	*EKV model*
MOS	49	BSIM3V1 model
MOS	49	BSIM3V1A model (Alan Gillespie)
MOS	49	BSIM3V0 model
MOS	62	STAG model

Improvements in NGSPICE

Semiconductor devices (ADMS)

- admsXml (L. Lemaitre) compiler needed to compile Verilog-A models to C source ready for link with NGSPICE
- Interface is still under development (run time, convergence problems, noise not implemented)
- CMC claimed Verilog-A as the standard model exchange format

Type	Level	models
BJT	*6*	*MEXTRAM 504 6.1*
BJT	*7*	*HICUM0 1.12*
BJT	*8*	*HICUM2 2.24*
MOS	*44*	*EKV 2.6*
MOS	*45*	*PSP 102.1*

Improvements in NGSPICE

Frontend improvements

- Memory usage:
  - Original spice3f5 had severe memory allocation problems that prevented multiple simulations without exiting from the software.
  - Most of the condition that caused NGSPICE to segfaults have been corrected.
- Parametric netlists:
  - NGSPICE supports the use of parameters and user defined functions in netlists.
- MEAS(ure):
  - This command analyze the output of DC, AC and transient analysis and reports quantities as propagation delay, rise and fall times, peak to peak voltage, etc.
- Monte Carlo analysis:
  - Absolute and relative probability functions (gauss, uniform) base on high quality random generator provided by NGSPICE
  - MC runs initiated on script level
  - Results can be investigated on plot level

Improvements in NGSPICE

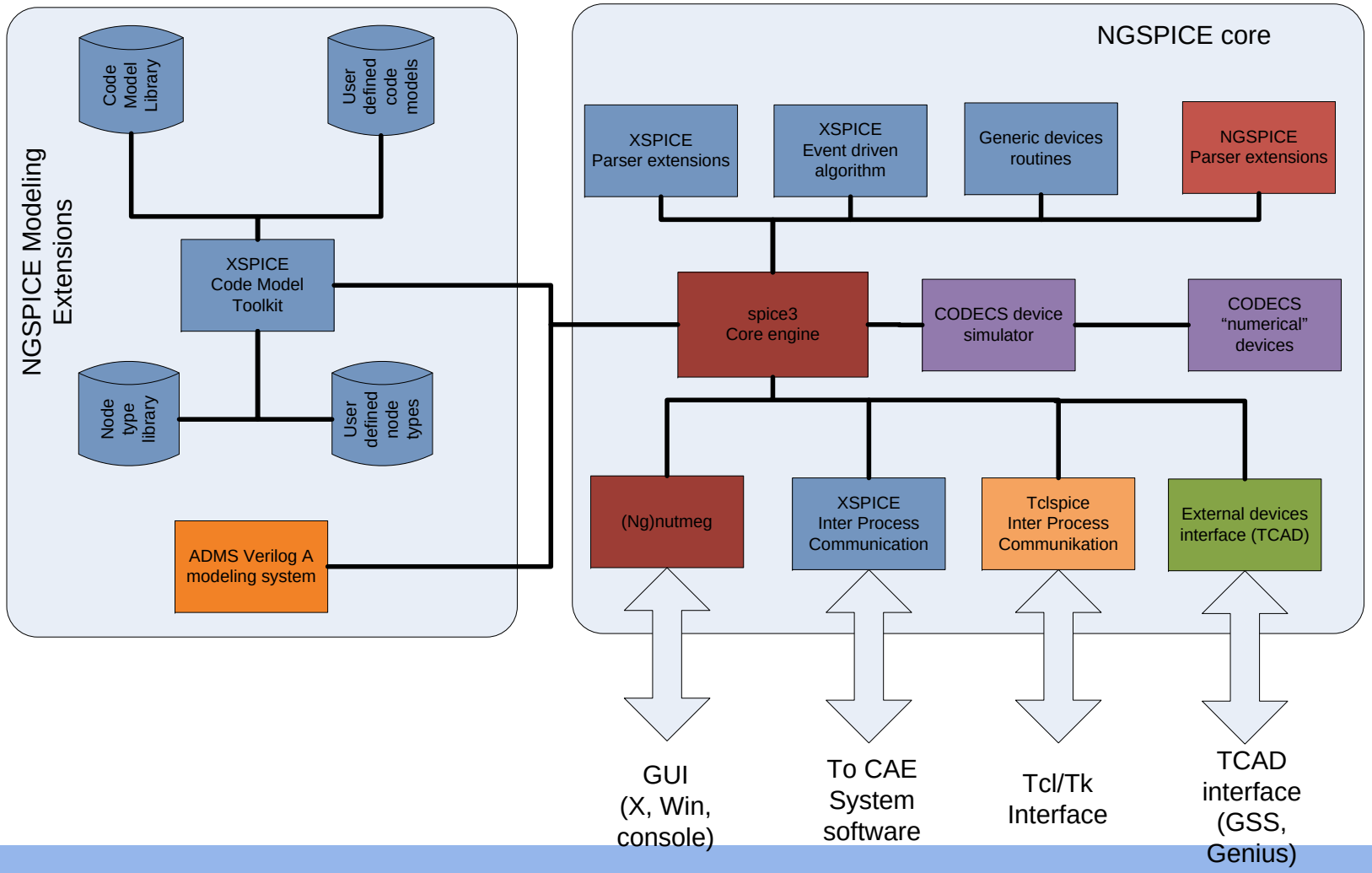
Parallel Computing

- OpenMP
  - The OpenMP Application Program Interface (API) supports multi-platform shared-memory parallel programming in C/C++ and Fortran on all architectures, including Unix platforms and Windows platforms.

- NGSPICE OpenMP implementation:
  - Actual BSIM3 and BSIM4 models can use OpenMP to exploit parallelism of multicore – multiprocessor machines in multithreading mode.
  - Very simple API – only few modifications in source is needed by usage of #pragmas
  - Tested under Linux (gcc4), Solaris and Windows (mingw64, VC++)
  - # of threads is selectable (e.g. set num_threads=6 quadcore CPU)
  - Performance rises with more than sqrt(threads)

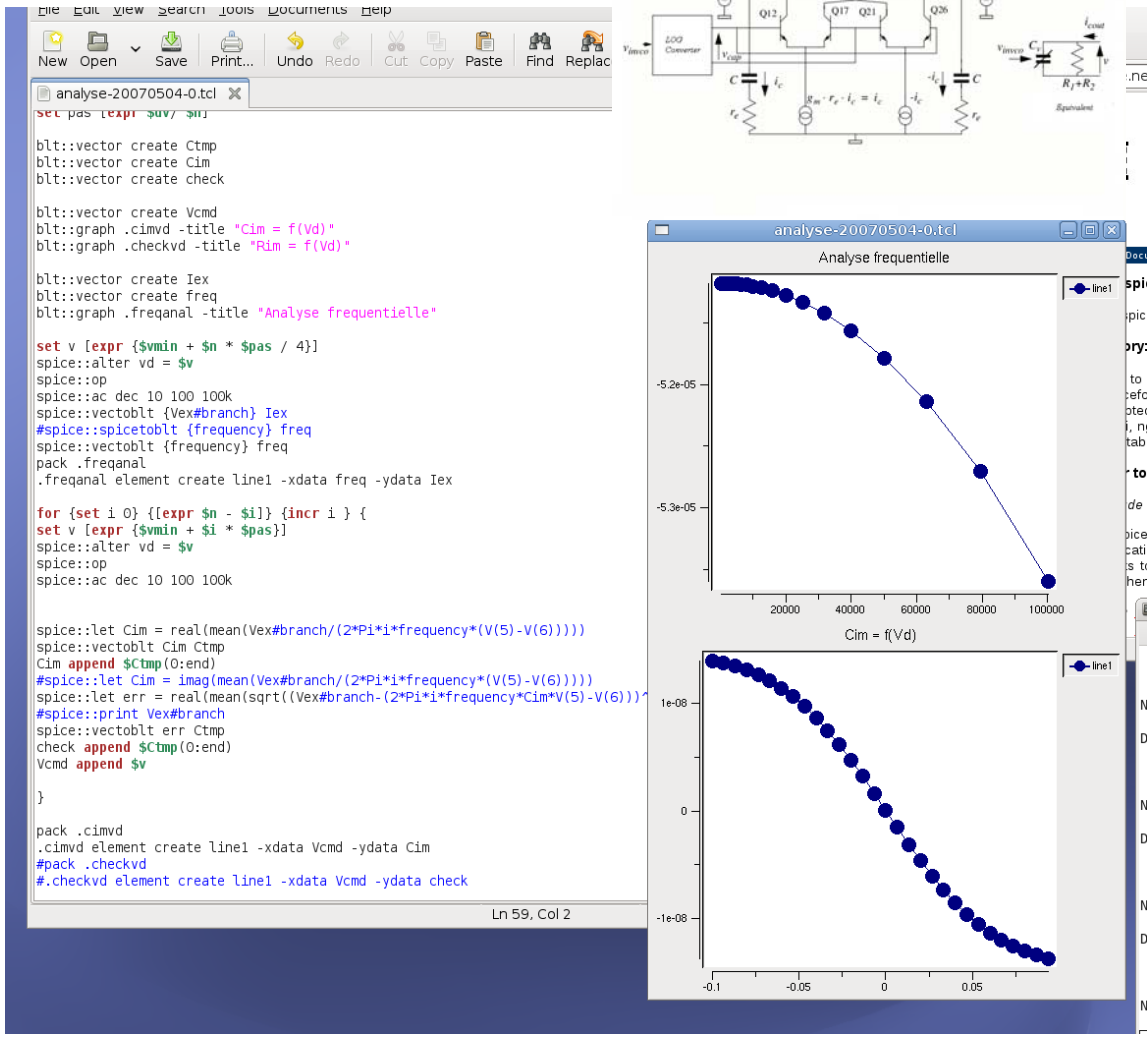
NGSPICE as simulator

simply another SPICE…



tclspice – another frontend to NGSPICE

- tclspice allows
  - Provides a modern IPC channel interface
  - To simulate circuits and export data to post-processing applications
  - Run simulations in optimization loop
- In TCL we define the SPICE vector:
  - **`SPICE::let Cim = real(mean(Vex#branch/(2*Pi*i*frequency*(V(5)-V(6)))))`**
  - Run analyses at different voltages.
  - Plot the results on fly by BLT package.
  - Vector can be exported to another application driven by the same TCL script.



Genius TCAD simulator

- Genius is a TCAD tool designed by Cogenda (www.cogenda.com) and, partially released as open source under GPL V3:
  - 2D simulations (no 3D)
  - Serial execution (no parallel code)
  - Triangle mesh generator only
  - Basic set of materials
- Genius TCAD can simulate circuits containing TCAD-devices with NGSPICE:
  - NGSPICE is loaded as a shared library (DLL) inside Genius

Mixed level simulation – CIDER Device simulator

- The idea of mixing TCAD and SPICE simulations is not new.
- 1994 CIDER 1b1:
  - Couples spice3f.2 to a C-based device simulator DSIM.
  - Poisson and continuity equation solver in 1D and 2D compatible with Stanford's PISCES.
  - Netlists can contains compact and "numerical" models.

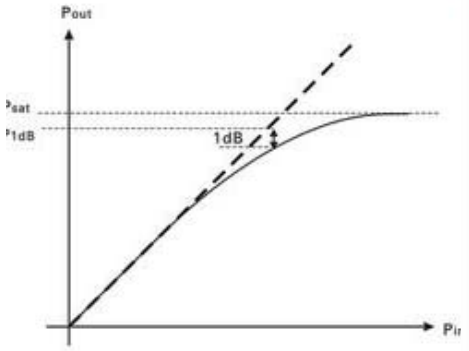
Model	Description
numd	1D junction diode model
numd2	2D junction diode model
numbjt	1D bjt model
numbjt2	2D bjt model
numos	2D mosfet model

- A fixed number of basic devices is defined.
- Only 1D and 2D device models support.

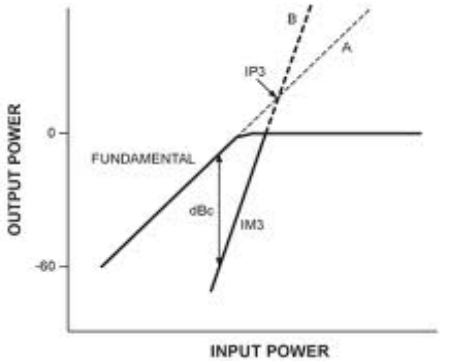
Implementing PSS in NGSPICE

Periodic Steady State (PSS) is the most accurate tool in simulated characterization of RF systems.

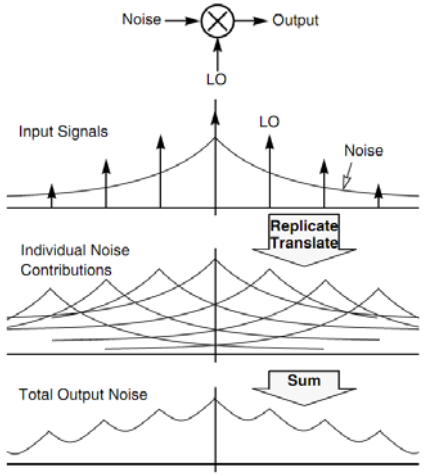
- Autonomous systems: Oscillators & PLL's
- Non-autonomous systems: Mixer, LNA's & PA's
- compression point, IP3 and other non linear harmonic content evaluations
- cyclo-stationary noise evaluation as folding in mixers and phase noise in oscillators



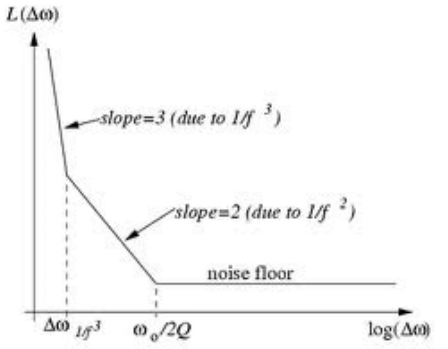
1dB compression point



IP3 evaluation



mixer noise



oscillator phase noise

PSS analysis: RTWO example

RTWO benchmark: 456 circuit equations with BSIM3v3 model

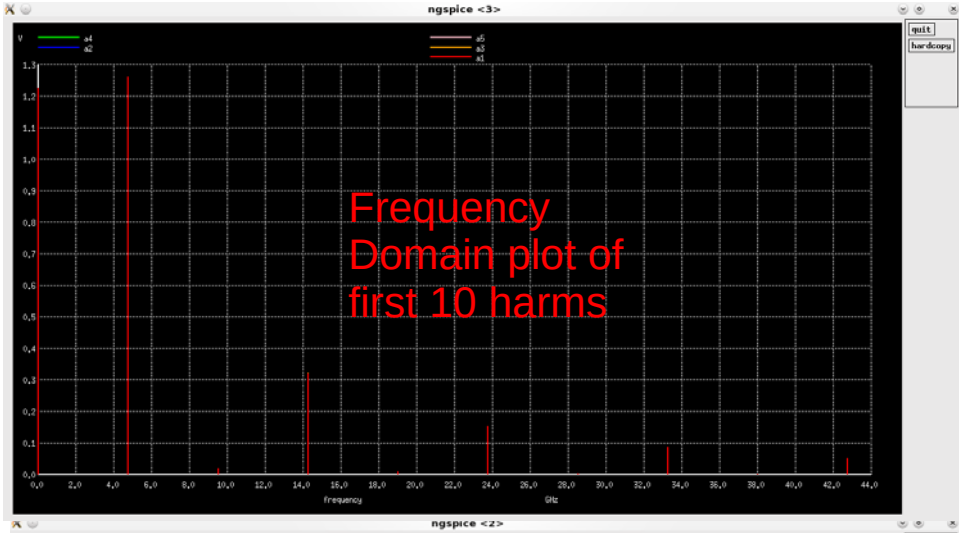
TITLE: Rotary Traveling Wave Oscillator

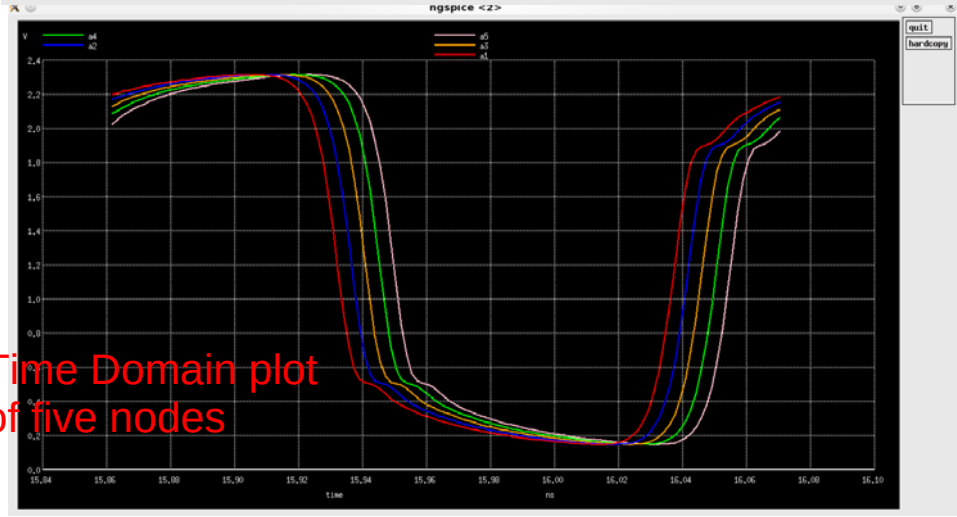
.OPTIONS TEMP=25 noacct

********** PSS Analysis Setup ************
.pss 4.45e9 15n 1 128 10 3 10e-3 uic
****

VSLEW_CONTROL VSLEW 0 (PULSE 0 1 0 1e-09)
EVLOGIC VRAMP 0 VSLEW 0 2.5
VDDPOWER VDD VRAMP DC 0
VARACTOR_V VARACTOR_V 0 DC 2.5

VDD0 VDD0 VDD DC 0
VSS0 VSS0 0 DC 0
VDD_A0 VDD_A0 VDD0 DC 0
VSS_A0 VSS_A0 VSS0 DC 0
VDD_B0 VDD_B0 VDD0 DC 0
VSS_B0 VSS_B0 VSS0 DC 0
LA0 A0 LCA0 3.69030941553353e-11
RA0 LCA0 A1 0.266535044422507
LB0 B0 LCB0 3.69030941553353e-11
[...]



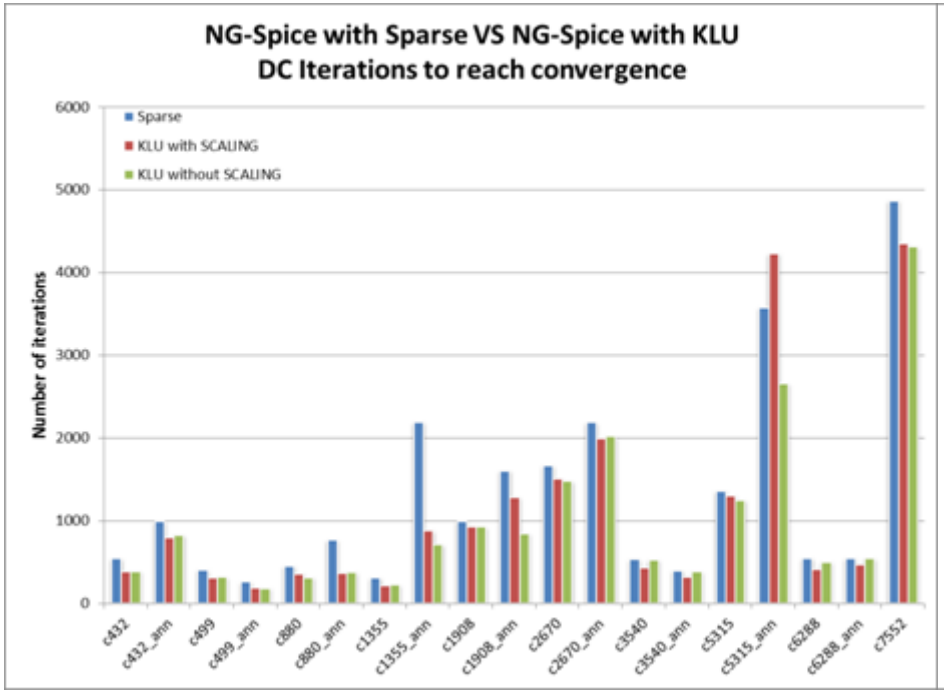


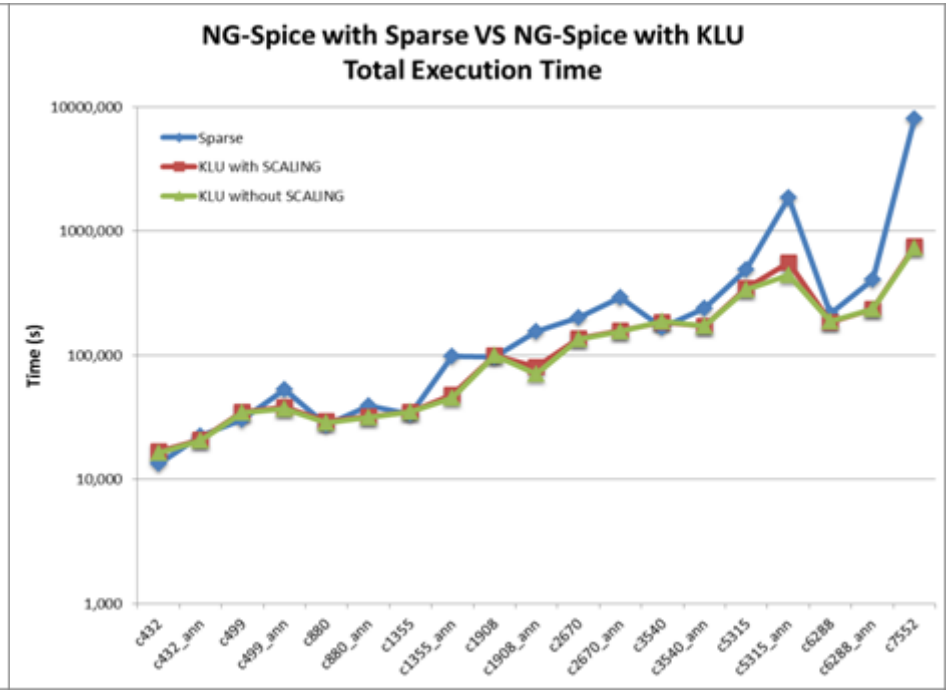
Alternative sparse linear solver: KLU

Can be applied to all kind of analysis (dc, transient, ac, pss)

Currently supports all devices models with exception of Cider dependent ones and urc model

Achieves up to 11x speed-up in transient analysis on ISCAS85 testbench c7552 with respect to Sparse (1.3 due to Kundert)





Roadmap: Netlist syntax and Model interface

- Steadily update of BSIM3 and BSIM4 according Berkeley releases
- Same for MEXTRAM and HICUM BJT's
- Re-animate the Verilog-A interface – improve convergence behavior of generated models
- HiSIM_HV is implemented in version 1.2.2, but now major release changed to 2.0.1: Is it also approved by CMC?
- MOSFET ACM option
- Implement more useful features in netlist syntax: temper in .param statements
- Better test-benches ! CMC-QA Benchmark?

Conclusions

- SPICE after 40 years are still invaluable tools for circuit design, analysis and verification.
- NGSPICE is a versatile platform integrable in higher-level applications to provide accurate analog simulation.
- sourceforge code repository is reorganized with **git** to allow better branched developments – open for new maintainer!
- Future development have been presented.
- Visit ***www.ngspice.org*** and ***http://sourceforge.net/projects/ngspice***