Ngspice: Recent progresses and future plans

Paolo Nenzi$^{1,2}$, Francesco Lannutti$^{1,2}$, Robert Larice$^2$, Holger Vogt$^2$, Dietmar Warning$^2$

1) DIET – University of Roma “Sapienza”; 2) Ngspice development team

11th MOS-AK/GSA ESSDERC/ESSCIRC Workshop, September 20th 2013 - Bucharest (RO)
Outline

• Introduction
  – A spicy history
  – Ngspice development model
  – Ngspice (actual) structure

• Recent progresses
  – Ngspice as a shared library
    • Future plans: parallel execution of multiple Ngspice instances
  – Ngspice statistical simulations
    • Future plans: VAI-SPICE
  – Ngspice frontend improvements
    • Conditional netlists, Snapshot file, matrix dump
  – Ngspice in mobility

• Conclusions
A spicy history

• ‘70s:

• ‘80s:

• ‘90s:
  – **1993** T. Quarles, SPICE3f
  – **1997** T. Quarles, SPICE3f5
  – **1999** NGSPICE
Ngspice development model

• Ngspice is an open source software released under the BSD license.
• Ngspice development is carried on by a core team of developers that improves and maintains the tool:
  • git source code management system is used as version control system:
    git://git.code.sf.net/p/ngspice/ngspice
  • contributions are included after evaluation and test,
  • improvements in the simulator’s code are kept in separate branches (available in repository) and merged when stable.
• Ngspice developers try to release an update per year:
  • latest available release if ngspice-25 (Jan 3rd, 2013)
• Ngspice improvement efforts concentrates on three directions: models, analyses, frontend.
Ngspice actual structure

Ngspice: Recent progresses and future plans
ngspice release 25, January 3rd, 2013

As always: new features have been added to ngspice-25, improving its applicability and stability.

- Bug fixes: Memory leaks removed, especially for several XSPICE devices, improved code compliance.
- New features: CMC model qa check for actual BSIM and hisim devices; additional parameters accessible in bsim3; new or updated commands: 'snsave', 'snload'; new code model 'memristor'; hisim models updated; 'measure' speed enhanced; port for Mac OS X.
- Documentation: Updated pdf manual and other documentation.

Ngspice’s

ACTUAL PROGRESSES
2010 Ngspice as simulation kernel - TCLspice

Ngspice: Recent progresses and future plans

9/20/13
TCLspice

- TCLspice started as an independent project by Stefan Jones and now is fully integrated into ngspice.
- TCLspice allows:
  - to simulate circuits and export data to post-processing applications
  - perform post synthesis of digital ASICs getting data from the TCL shell included in several digital design tools.
  - run simulations in an optimization loop.
Ngspice as a shared library

- Ngspice can be compiled as a shared library (or a DLL).
- An external application can control the simulation and exchange data with it.
- Advantages:
  - Embeddable circuit simulator.
  - Provide a user interface or IDE to ngspice without modifying simulator’s code.
- Small API:

```c
int ngSpice_Init(SendChar*, SendStat*, ControlledExit*, SendData*, SendInit- Data*, BGThreadRunning*, void);
int ngSpice_Init_Sync(GetVSRCData* , GetISRCData* , GetSyncData* , int*, void*);
int ngSpice_Command(char*);
bool ngSpice_running (void);
pvector_info ngGet_Vec_Info(char*);
int ngSpice_Circ(char**);
char* ngSpice_CurPlot(void);
char** ngSpice_AllPlots(void);
char** ngSpice_AllVecs(char*);
bool ngSpice_SetBkpt(double);
```
Ngspice as a shared library

• Example:
  • Initialize ngspice,
  • load a netlist from a file,
  • write output to a file.

... 

/* Initialize ngspice */
res = ngSpice_Init(NULL, NULL, cnt_exit, NULL, NULL, self);

/* Load a netlist */
ngSpice_Command("source ../examples/adder_mos.cir");

/* Execute ‘run’ command */
ngSpice_Command("run");

/* Write out results */
ngSpice_Command("write testout.raw v(2)");

...
Ngspice as a shared library

Example:
- Initialize ngspice,
- write netlist element by element,
- background execution
- results printout.

```c
/* Initialize ngspice */
res = ngSpice_Init(print_fnct, NULL, cnt_exit, NULL, NULL, self);

/* Input a netlist */
ngSpice_Command("circbyline DC test ");
ngSpice_Command(“circbyline V1 1 0 1”);
ngSpice_Command("circbyline R1 1 2 1");
ngSpice_command("circbyline R2 2 0 1");
ngSpice_Command("circbyline .dc V1 0 0 0.1");
ngSpice_Command("circbyline .end");

ngSpice_Command("bg_run");    /* Execute ngspice in a separate thread */
ngSpice_Command("bg_halt");    /* Halt ngspice */
ngSpice_Command("bg_resume"); /* Resume simulation */

/* Print results via the print_fnct callback function */
ngSpice_Command("print v(2) ");
```

...
Ngspice as a shared library

- **Future extension: parallel execution.**
- Allows to run several synchronized instances of ngspice,
- no yet “ready to use” implementation”,
- only for TRANsient simulation and no XSPICE support,
- uses `int ngSpice_Init_Sync(GetVSRCDat...`,
- `GetSyncData`, `int*`, `void*`);
- Instances exchanges data via current and voltage generators:
  - the callback functions `GetVSRCDat` `getvdat` and
  `GetISRCData` `getidat` to retrieve external voltage or current inputs,
  - timestep advances when all instances have converged.
Ngspice statistical simulations

- Monte Carlo simulation of circuits has been implemented into Ngspice:
  - Random number generators:
    - gauss(nom, rvar, sigma)
    - agauss(nom, avar, sigma)
    - unif(nom, rvar)
    - aunif(nom, avar)
    - limit(nom, avar)
  - Random seed can be set to reproduce pseudorandom sequences:
    - set rndseed=<some integer>
  - Implementation uses front-end parameters and ngspice scripting language
  - Function returning random vectors have been introduced:
    - rnd(vector), sgauss(vector), sunif(vector), poisson(vector), exponential(vector)
Ngspice statistical simulations

Perform Monte Carlo simulation in ngspice (LC bandpass filter)
V1 N001 0 AC 1 DC 0
R1 N002 N001 141
*
C1 OUT 0 1e-09
L1 OUT 0 10e-06
C2 N002 0 1e-09
L2 N002 0 10e-06
L3 N003 N002 40e-06 C3 OUT N003 250e-12
*
R2 0 OUT 141
*
.control
let mc_runs = 100
let run = 1
set curplot = new          // create a new plot
set scratch = $curplot      // store it’s name to ‘scratch’
*
define unif(nom, var) (nom + nom*var * sunif(0))
define aunif (nom, avar ) (nom + avar * sunif (0))
define gauss (nom, var , sig ) (nom + nom*var / sig * sgauss (0))
define agauss (nom, avar , sig ) (nom + avar / sig * sgauss (0))
*
Ngspice statistical simulations

... dowhile run <= mc_runs
alter c1 = gauss(1e-09, 0.1, 3)
alter l1 = agauss(10e-06, 2e-06, 3)
alter c2 = agauss(1e-09, 100e-12, 3)
alter l2 = gauss(10e-06, 0.2, 3)
alter l3 = agauss(40e-06, 8e-06, 3)
alter c3 = gauss(250e-12, 0.15, 3)
ac oct 100 250K 10Meg
*
set run =$&run" $ create a variable from the vector
set dt = $curplot $ store the current plot to dt
setplot $scratch $ make 'scratch ' the active plot
* store the output vector to plot 'scratch '
let vout{$run}={$dt}.v(out)
setplot $dt $ go back to the previous plot
let run=run+1 end
plot db({$scratch}.all)
.endc
.end
Ngspice statistical simulations

- Example:
  - Ngspice: analysis of a statistical Wien bridge oscillator
  - Ngspice can generate a gnuplot plot
  - GNUplot: fitting with a gaussian curve (f1 fitted curve, f2 starting curve)
Ngspice statistical simulations

- Future extension: VAI-SPICE
  - Variability-aware Inherently Statistical SPICE Simulation Engine
  - DIET VLSI group (Prof. M. Olivieri) is working on this

Theoretical formulation of the method has been developed in:

Conditional netlists

- IF-ELSE condition to control the element in a netlist.
- Actual implementation does not support nested IF-ELSE, only one ELSEIF is allowed.

```plaintext
device instance in IF-ELSE block

.param ok=0 ok2=1
v1 1 0 1
R1 1 0 2
.if (ok&&ok2)
R11 1 0 2
.else
R11 1 0 0.5 ; this is selected
.endif
```
Snapshot file (load/save)

- Ngspice can now save/load the snapshot file of a transient simulation
- XSPICE and polynomial I/V sources not supported

Example input file for snsave / snload
* load a circuit (including transistor models and .tran command)
* starts transient simulation until stop point

.include adder_mos_circ.cir
.control
* cd to where all files are located
.cd ngspice/examples/snapshot
.set noaskquit
.set noinit
* interrupt condition for the simulation
.stop when time > 500n
* simulate
.run
* store snapshot to file
 snsave adder500.snap
.quit
.endc
.end
Snapshot file (load/save)

• ... and load the snapshot file in another instance of ngspice

* script: Example input file for snsave / snload
* reload the circuit and continue the simulation

.control
* cd to where all files are located
cd ngspice/examples/snapshot
snload adder_mos_circ.cir adder500.snap
* continue simulation
resume
* plot some node voltages
plot v(10) v(11) v(12)
.endc
.end
Matrix dump

- Ngspice can dump the system matrix and the RHS vector to a file or to console.
- The `mdump` and `mrdump` commands implement this feature.
- Matrix and vector dump is a valuable tool for developers to verify their code.

```
*Dump matrix and RHS values after 10 and 20 steps
*of a transient simulation
source rc.cir
step 10
mdump m1.txt
mrdump mr1.txt
step 10
mdump m2.txt
mrdump mr2.txt
* just to continue to the end
step 10000
```
The next step

NGSPICE IN MOBILITY
Ngspice for mobile devices

• Piero Belforte and Ed Pataki developed iSchematics, a ngspice fronted for ngspice that runs on iOs devices.
• Schematic capture and data output are available on the mobile device.
• Simulation engine runs on remote servers.
• It integrates SWAN an Digital Wave Filter simulator developed by Piero.
Ngspice for mobile devices (Android)

- Calin Andrian develops WeSpice, an implementation of Ngspice running on Android devices **locally**.
- No Internet connection is necessary to analyze circuits.
- It allows schematic capture and data visualization.
Conclusions

• Some of the recent improvements in ngspice have been presented:
  – Shared library implementation
  – Statistical analysis

• *Third party* implementation of ngspice on mobile devices have been introduced:
  – they represent the continuing interest on this software after 40 years

• All of the examples in this presentation are available on the Ngspice Users Manual (version 25plus):