Experimenting OMC and --daeMode with large-scale grid models: status and perspectives

Francesco Casella
(francesco.casella@polimi.it)

POLITECNICO MILANO 1863

Dynamica

Adrien Guironnet
(adrien.guironnet@rte-france.com)

Rte

Background

Simulation of Power Transmission and Distribution Grids

- Flexibility
  - New models
  - Innovative components
  - Different solvers
  - Multi-domain models

- Ease of model development
  - Declarative equation-based approach
  - Inheritance for basic setups and housekeeping

- Open-Source environments
  - OS component models
  - OS system models
  - OS solvers

Known Issues with Current Modelica Tools

Sparse DAE, Dense ODE

↓

daeMode required for
efficient performance!

Large numbers of repeated components
+
Flattening to scalar equations

↓

Huge code generation time and size

Requirements

Advanced Modelica tools
using daeMode and
array-based code generation

Open Benchmarks
to assess the performance
of such tools in this area

Previous Work

2016-2017 OpenModelica

- Sparse solver support (Willi Braun)
  - UMFPACK, KLU (linear sparse solvers)
  - KINSOL (Nonlinear Newton solver from Sundials, supports sparse solvers)
  - IDA used as sparse ODE stiff solver instead of DASSL

- DAEMode (Willi Braun)
  - Direct use of IDA for DAE integration after index reduction
  - IDA solves differential and implicit algebraic loops simultaneously
  - Native sparse solver support

W. Braun, F. Casella and B. Bachmann, "Solving large-scale Modelica models: new approaches and experimental results using OpenModelica". Proc. 12th International Modelica Conference, Prague, Czech Republic, May 15-17, 2017, pp. 557-563

2016-2017 Politecnico MI / Dynamica / CESI / TERNA

- Feasibility study of national transmission grid modelling using Modelica and OpenModelica 1.11.0
  - Prototype Modelica library with basic component models
  - Four test cases of HV trasmission systems

Network	Nodes	Gens	Lines	Trafos	Equations
GRID_C	751	74	369	583	56386
GRID_E	1817	267	1458	1202	157022
GRID_D	8376	2317	1946	2489	579470
GRID_G	8113	407	6833	2824	593886

Network	Flattening	C gen.	Compilation	Simulation
GRID_C	24	24	13	12
GRID_E	73	67	35	44
GRID_D	334	315	123	111
GRID_G	318	303	144	186

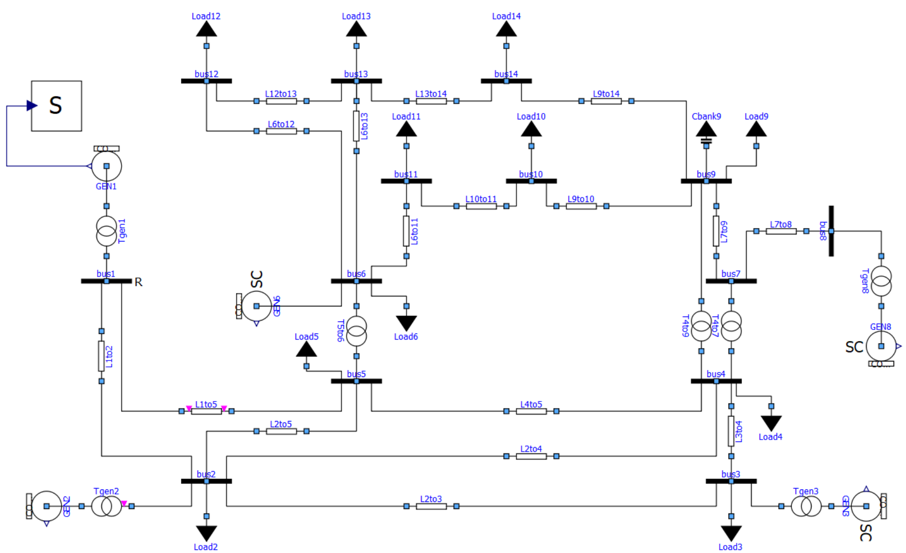
- Proprietary library / Confidential example cases

F. Casella, A. Bartolini, S. Pasquini and L. Bonuglia, "Object-Oriented Modelling and Simulation of Large-Scale Electrical Power Systems using Modelica: a First Feasibility Study". Proc. IEEE IECON 2016, Firenze, Italy, Oct. 24-27, 2016

F. Casella, A. Leva and A. Bartolini, "Simulation of large grids in OpenModelica: reflections and perspectives". Proc. 12th International Modelica Conference, Prague, Czech Republic, 2017, pp. 227-233.

2018-2020 Politecnico MI / Dynamica / RTE

- PowerGrids library
  - Open Source (**https://github.com/PowerGrids/PowerGrids**)
  - Main goals: demonstrator / teaching / research for power grid models
  - Advanced Modelica concepts / Easy model development
  - Small test cases (up to 14-bus IEEE benchmark)




```
model LineConstantImpedance "Transmission line with constant impedance"
  extends BaseClasses.PiNetwork(
    UNomA = UNom,
    UNomB = UNomA);
  extends Icons.Line;

  parameter Types.Voltage UNom(start = 400e3) "Nominal/rated voltage";
  parameter Types.Resistance R "Series resistance";
  parameter Types.Reactance X "Series reactance";
  parameter Types.Conductance G = 0  "Shunt conductance";
  parameter Types.Susceptance B = 0 "Shunt susceptance";
equation
  k = Complex(1);
  Y = 1/Complex(R, X);
  YA = Complex(G/2, B/2);
  YB = Complex(G/2, B/2);
annotation( ...);
end LineConstantImpedance;
```

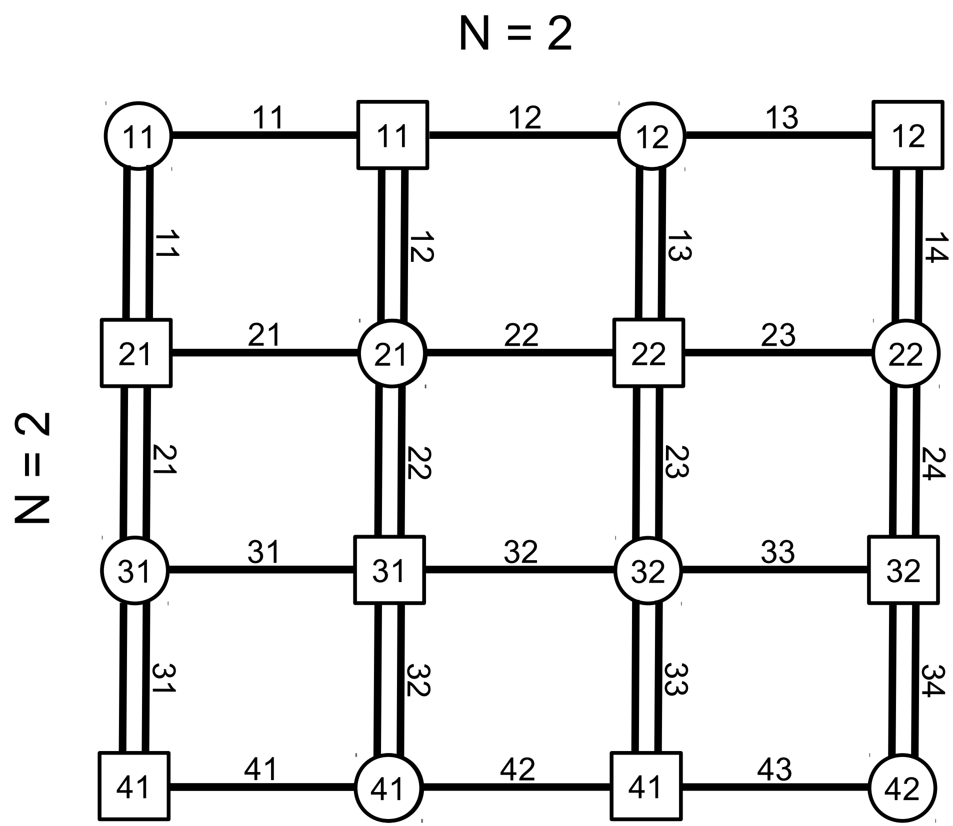
A. Bartolini, F. Casella and A. Guironnet, "Towards Pan-European Power Grid Modelling in Modelica: Design Principles and a Prototype for a Reference Power System Library". Proc. 13th International Modelica Conference, Regensburg, Germany, Mar 4-6, 2019, pp. 627-636

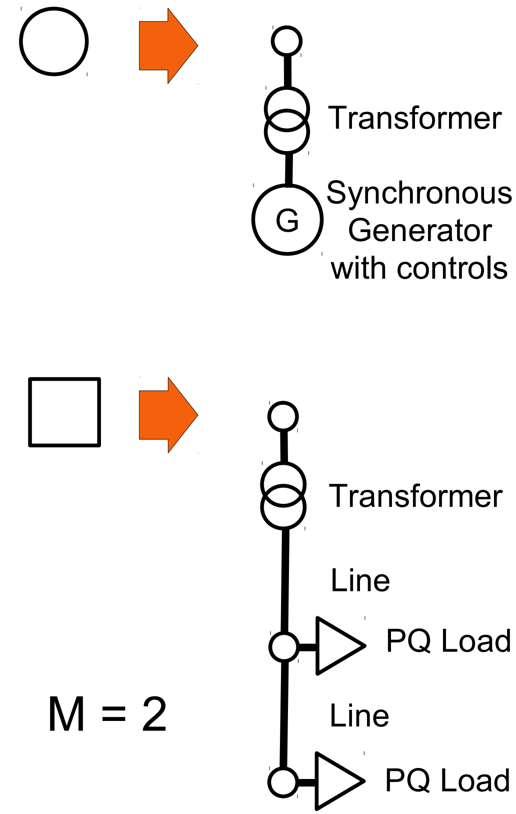
Current Work

2020 Politecnico MI / Dynamica / RTE

- ScalableTestGrids library
  - Open Source (`https://github.com/PowerGrids/ScalableTestGrids`)
  - Provide open-source scalable power grid benchmark models in Modelica
  - Assess the performance of existing Modelica tools
  - Drive the development of next-generation Modelica tools
- Features
  - Representative power grid structure
  - Scaled by Integer parameters
  - Includes
    - Generators with voltage and frequency controllers
    - Transformers
    - Transmission lines,
    - PQ loads
  - Large number of individually instantiated components (mimicks grid models generated from netlists by scripts)
  - Modelica functions to generate arbitrarily scaled models
- Simulation
  - Steady-state initialization, using homotopy
  - Step change load of upper half of the grid, causing global and inter-area damped frequency oscillations.

Model Structure (N = 2, M = 2)





$2N^2$ Generators
$2N^2M$ Loads
$4N^2$ Transformers
$6N(2N-1) + 2N^2M$ Lines

Simulation with OpenModelica

- Simulation with `--daeMode`
- Sparse Kinsol/KLU solver for initialization
- Fixed-step homotopy solver, separate $\lambda = 0$ handling
- No tearing
  - Faster code generation
  - Sparse solvers recover efficiency
  - Avoids losing strategic start values of nonlinear variables vs. irrelevant start values of linear variables
- Currently set by `__OpenModelica` vendor annotations
- Should be automatically set up by the tool in the future, based on system structure analysis
- Test machines
  - Intel Xeon E5-2650 2.3 GHz 20 virtual cores 72 GB RAM Ubuntu 18.04 LTS
  - Intel i7 8550U 1.8 GHz 8 virtual cores 16 GB RAM Windows 10 Pro
- OMC v1.17.0-dev-356-gc2c52350cb

Problems Highlighted by the Tests

- Symbolic Jacobians for initialization equations were not generated if homotopy was used (✓*fixed*)
- Separate set of initial equation for $\lambda = 0$ was not generated if homotopy was used (✓fixed)
- Symbolic Jacobians for simulation are not generated in --daeMode (✗*pending*)
- Memory access error in Linux for larger systems, probably due to old version of sparse algebra library (✗*pending*)
- Proper code generation and simulation options are not selected automatically (✗*pending*)

Results with OpenModelica

							Xeon E5-2650, Ubuntu 18.04					i7 8550U, Windows 10				
N	M	# equations	# generators	# transformers	# lines	# loads	code gen time / s	C compile time / s	exec size [MB]	sim time / s	# solver steps	code gen time / s	C compile time / s	exec size [MB]	sim time / s	# solver steps
2	4	12174	8	16	14	32	18.8	3.6	13.5	0.9	297.0	19.4	12.7	43.9	0.8	297.0
3	4	28284	18	36	25	72	41.6	8.4	31.2	2.2	319.0	48.8	24.6	59.5	2.0	319.0
4	4	51078	32	64	36	128	84.6	14.8	56.2	4.0	315.0	95.3	35.7	81.6	3.4	315.0
6	4	116718	72	144	58	288	*170.0*	*30.0*	*110.0*	*8.0*	*300.0*	219.9	66.0	145.3	8.9	293.0
8	4	209094	128	256	80	512	*350.0*	*60.0*	*220.0*	*16.0*	*300.0*	*500.0*	*130.0*	*300.0*	*20.0*	*300.0*
11	4	397788	242	484	113	968	*700.0*	*120.0*	*440.0*	*32.0*	*300.0*	*1,000.0*	*260.0*	*600.0*	*40.0*	*300.0*
16	4	~800000	512	1024	168	2048	*1,500.0*	*240.0*	*900.0*	*64.0*	*300.0*	*2,000.0*	*500.0*	*1,200.0*	*80.0*	*300.0*
23	4	~1600000	1058	2116	245	4232	*3,000.0*	*480.0*	*1,800.0*	*128.0*	*300.0*	*4,000.0*	*1,000.0*	*2,400.0*	*160.0*	*300.0*
32	4	~3200000	2048	4096	344	8192	*6,000.0*	*1,000.0*	*3,600.0*	*256.0*	*300.0*	*10,000.0*	*2,000.0*	*4,800.0*	*320.0*	*300.0*
45	4	~6400000	4050	8100	487	16200	*12,000.0*	*2,000.0*	*7,200.0*	*512.0*	*300.0*	*20,000.0*	*4,000.0*	*10,000.0*	*640.0*	*300.0*
64	4	~12800000	8192	16384	696	32768	*24,000.0*	*4,000.0*	*14,400.0*	*1,024.0*	*300.0*	*40,000.0*	*8,000.0*	*20,000.0*	*1,300.0*	*300.0*

Experimental results

Projected results – Reasonable performance

Projected results – Not yet reasonable performance

Outlook

- Simulation performance good for size up to about 4000 components
- Improvements needed above that:
  - Symbolic Jacobians for simulation
  - More efficient and streamlined run-time code
  - Faster hardware (latest AMD Ryzen expected to be 3X faster)

- Code generation and compilation OK up to about 200 components
- Vectorized backend and code generation required above that level
- Interesting to test with other Modelica tools

Conclusions

- Growing interest for large-scale power grid simulation in Modelica
- ScalableTestGrid library available as open-source
- Allows testing Modelica tool performance with power grid models of arbitrary size
- Current status with OpenModelica
  - Usable without problems up to 200 components
  - Could be used with some patience up to 1000/2000 components
- Further size increase requires breakthrough in vectorized code generation
- On-going work:
  - FH Bielefeld: new backend
  - Linköping University: LargeDyn project
  - Politecnico di Milano: HiPerMod project

Thank you for your kind attention!