Computational Lab "Model Reduction of a Thermal Model"

Evgenii B. Rudnyi, 2005, http://Evgenii.Rudnyi.Ru/, http://www.imtek.uni-freiburg.de/simulation/

Goals:

1) Learning by doing how to use mor4ansys (version 1.8) and Post4MOR.

2) Comparing computational time as well as accuracy for transient simulation of a full-scale thermal model in ANSYS and a reduced model obtained by mor4ansys.

Software required:

1) ANSYS (starting version 6.1) to build the finite element model and make its transient simulation.

2) mor4ansys to make a reduced model directly from the ANSYS finite element model.

The code and binaries for Sun, Linux and Windows as well as documentation are available at

`http://www.imtek.uni-freiburg.de/simulation/mor4ansys/`

3) Mathematica (starting version 5.0) to make simulation of a reduced model and compare results.

4) Post4MOR to provide useful functions in Mathematica.

The code and documentation are available at

`http://evgenii.rudnyi.ru/programming.html#post4mor`

Important notice. A reduced model as well as the result of ANSYS transient simulation is written in the Matrix Market format (plain ASCII files, http://math.nist.gov/MatrixMarket/). You can use another environment to simulate a reduced model and make plots. For example, you can read Matrix Market files into Matlab. However, in this case you should create your own functions to reach the lab goal.

Files:

`runansys` is a sample script to run ANYS in the batch mode.

`model.ans` is a ANSYS script to build the model.

`matrices.ans` is a ANSYS script to generate matrices for mor4ansys.

`simulation.ans` is a ANSYS script to run transient simulation.

`fixstep.ans`, `adaptive.ans` and `logtime.ans` are ANSYS scripts to use different strategies for transient simulation. They use `simulation.ans`.

`plots.nb` is a Mathematica notebook with the command to run simulation and make plots.

Computational Time:

The lab has been made for a 500 MHz Sun workstation with 256 Mb RAM. It takes about 10 min to build the model in ANSYS and then about 1 hour for each transient simulation. If you use a modern computer, you may afford to change `elem=0` to `elem=1` in the beginning of `model.ans`. This will increase the number of degrees of freedom from about 20000 to about 80000.

The computational time on the Mathematica side is within seconds.

Background for Thermal Model:

The thermal model of a microthruster unit is used in the lab. The ANSYS script `model.ans` has been developed by Tamara Bechtold. In the context of the lab, this is quite a generic thermal device.

There is a heater and the task is to simulate temperatures as at some points as a function in time during the step response. You can find more information about the original engineering problem in

E. B. Rudnyi, T. Bechtold, J. G. Korvink, C. Rossi, *Solid Propellant Microthruster: Theory of Operation and Modelling Strategy*, Nanotech'02, September 2002, AIAA paper 2002-5755. Preprint is at http://www.imtek.uni-freiburg.de/simulation/pyros/rudnyi02.pdf

T. Bechtold, E. B. Rudnyi, J. G. Korvink and C. Rossi, *Efficient Modelling and Simulation of 3D Electro-Thermal Model for a Pyrotechnical Microthruster*. International Workshop on Micro and Nanotechnology for Power Generation and Energy Conversion Applications PowerMEMS 2003, Makuhari, Japan, 4-5 December 2003. Preprint is at http://www.imtek.uni-freiburg.de/simulation/mor4ansys/pdf/bechtold03PowerMEMS.pdf

Running ANSYS in the batch mode:

You can execute scripts from within ANSYS GUI: `File | Read Input From`. However in my view, when scripts are done it is more convenient to run them in the batch mode from the command line. A sample script to do it is `runansys`. Yet, it should be modified accordingly: change the path to ANSYS bin directory and the name of the license server as required.

The script takes a single argument, `filename`, it assumes that the script to read is in `filename.ans` and redirect ANSYS output to `filename.log`. It starts ANSYS as a background job. Use `ps` or `top` to watch the job status and check `filename.log` to see what ANSYS is doing.

The script will not work on Windows. Sorry.

Step 1. Building Model

Make sure that the ANSYS current directory contains lab files. This happens automatically if you run from this directory

```
./runansys model
```

Alternatively, execute the script `model.ans` from GUI.

The script should produce:

`file.db` is the ANSYS database for the model,

`file*.jpg` are model pictures,

`output.txt` is the file with the output nodes for mor4ansys. They are points in the middle of the model. The first is directly under the heater; the next nodes are below along the central vertical line. You can find the node definitions at the end of `model.ans`.

Record the number of nodes in the model and time to build the model.

Step 2. Running Transient Simulation in ANSYS

This step is not required to perform model reduction. We need it to compare the results of a reduced model with the original ANSYS model.

Integration in time numerically leads to numerical errors. It is important to know the magnitude of numerical integration errors when you compare results with the reduced model. To this end, we will use three integration strategies: 1) a fixed timestep, 2) adaptive integration, 3) a logarithmic timestep. The latter is widely used in electro-thermal simulation.

We assume step response. At time 0, the initial temperature is zero and electrical power to the heater becomes equal to 150 mW.

Run

```
./runansys fixstep
./runansys adaptive
./runansys logtime
```

Wait until the previous ANSYS job finishes, before you start the next script. Scripts need `file.db` from step 1 as well as the script `simulation.ans` in the current directory.

The scripts should create files `fixstep.txt, fixstep.txt.names, adaptive.txt, adaptive.txt.names, logtime.txt, logtime.txt.names`. They contain temperatures for the output nodes. File `*.txt` contains the numerical values in the Matrix Market format and `*.txt.names` contain output names.

Record the simulation time and the number of equations that have been solved. Why the number of equations is different from the number of nodes?

Step 3. Model Reduction with mor4ansys

Run the script `matrices.ans` to generate matrices for mor4ansys. Note that `file.full` after the transient simulation cannot be used for model reduction. It is necessary to generate files again.

```
./runansys matrices
```

Run the next command to perform model reduction

```
runmor4ansys file.full file.emat -C output.txt
```

On Windows the command is `mor4ansys`. You should receive files `mor.*` that contain the reduced model of dimension 30 in the Matrix Market format.

Record the time required for model reduction and the contribution to the total time from different steps. Record the dimension of the original system and compare with the number of equation in ANSYS.

Step 4. Simulation of the reduced model and plotting

Open `plots.nb` in Mathematica. Make the current directory contain the files made during previous steps (`SetDirectory`). Load functions from `Post4MOR.m`. Execute commands (command by command) in the notebook.

Record the time for simulation of the reduced system. Compare the time to perform model reduction and simulate the reduced system with the simulation time in ANSYS.

Step 5 (Options). Challenges

a) Play with ANSYS options to reduce time for transient simulation: 1) Play wtih memory options. 2) Try different solvers. 3) Re-use the factorization matrix in the case of the fixed timestep.

b) Try model reduction available in ANSYS (the Guyan method and mode superposition). You can find the comparison of the Arnoldi and Guyan in

T. Bechtold, E. B. Rudnyi and J. G. Korvink, Automatic Generation of Compact Electro-Thermal Models for Semiconductor Devices. IEICE Transactions on Electronics, 2003, v. E86C, N 3, pp. 459 - 465. Preprint is at http://www.imtek.uni-freiburg.de/simulation/mor4ansys/pdf/bechtold03IEICE.pdf, final paper is at http://search.ieice.org/2003/files/e000c03.htm#e86-c,3,459.

c) Try another input function of your choice. You do not need to perform model reduction again as it does not depend on the input function.