mor4ansys (version 1.8): Compact Behavioral Models from ANSYS by Means of Model Order Reduction

E. B. Rudnyi, J. G. Korvink,
IMTEK, Freiburg University, Germany, {rudnyi,korvink}@imtek.uni-freiburg.de
http://www.imtek.uni-freiburg/simulation/mor4ansys/

21.02.2005

What is new in 1.8

All information from the FULL file can be read into mor4ansys. This simplifies the transfer of information from ANSYS. As result, options -D and -T are removed.

Two additional methods to deal with the second order systems: the transformation to the first order system and the SOAR algorithm.

The code is rewritten. Now the computational routines make the mor4ansys library.

What is new in 1.6

Numerous bugs are fixed.

Model with constraint equations (CE command family) can be handled by mor4ansys. The implementation has not been optimized yet but seems to work.

Introduction

The goal of the manual is to describe the software `mor4ansys` (pronounced "more for ANSYS") that generates compact models directly from ANSYS models by means of model order reduction (see [1] for an overview). The advantages of this approach are as follows:

- ANSYS models are quite accurate. They can be created for a realistic CAD geometry of the device by taking into account small details that are important to make an accurate model.
- The software allows us to reduce the dimension of an ANSYS model significantly. From our experience [2], for thermal and mechanical problems the dimension of a reduced model of less than 30 allows us to have an accurate representation of original ANSYS models up to a dimension of 500 000.
- In order to run the software, only minimal knowledge about the background theory is sufficient to proceed. The generation of a compact model is almost automatic.
- The computational cost of model reduction is relatively small. Usually, the model reduction time is comparable to the solution of an according stationary problem in ANSYS. The analysis of the computational complexity is in Ref [3].

Let us explain what we mean by almost automatic. This means that there are a few important decisions to be taken by the engineer responsible for the simulation, namely:

- The software expects the ANSYS model to be linear. This means that if material properties and depend on temperature, one has to choose an appropriate operation point to linearize the model. Note that the nonlinearity in the input function, however, **is** allowed.
- Currently, it is not possible to choose the best order of the reduced system a priory. A user should choose the order of the reduced system manually.

We believe that these two conditions are quite reasonable considering the huge reduction in model dimension that you can achieve by using `mor4ansys`.

Note that we have research results (see publications at the mor4ansys site) that will allow us to overcome these limits in the future releases.

`mor4ansys` is a command-line tool. It reads input files, performs model reduction and then writes output files. We will start with basic theory, then we describe what inputs files are necessary and how one can prepare them in ANSYS. After than, we explain options that influence the model reduction process.

Basic Theory

After discretization, a system of ordinary differential equations is obtained

$$\begin{aligned} M\ddot{\boldsymbol{x}} + E\dot{\boldsymbol{x}} + K\boldsymbol{x} &= B\boldsymbol{u} \\ \boldsymbol{y} &= C\boldsymbol{x} \end{aligned} \tag{1}$$

where $\boldsymbol{x}$ is the vector of unknowns that includes all degrees of freedom, M, E, K are the mass, damping and stiffness matrices, B is the input, C the output matrix, $\boldsymbol{u}$ is the vector of inputs and $\boldsymbol{y}$ is the vector of outputs. Note that our naming convention [4] is slightly different from the usual finite element convention of using C as the damping matrix. First order systems are the special case of Eq (1) when the mass matrix is zero but we use a different naming convention [4] and first order systems are written as follows

$$\begin{aligned} E\dot{\boldsymbol{x}} &= A\boldsymbol{x} + B\boldsymbol{u} \\ \boldsymbol{y} &= C\boldsymbol{x} \end{aligned} \tag{2}$$

where $M = 0$ but $A = -K$.

`mor4ansys` comprises two modules: the first to assemble Eq (1) or (2) from ANSYS files (see the next section) and the second to run the Arnoldi process [5] to generate the reduced model.

The basic idea of model reduction is to find a low-dimensional subspace V that reasonably well approximates the transient behavior of $\boldsymbol{x}$, that is

$$\boldsymbol{x} = V\boldsymbol{z} + \varepsilon \tag{3}$$

where the approximation error ε is assumed to be small even though the number of columns of V (the dimension of $\boldsymbol{z}$) is much less the number of rows (the dimension of $\boldsymbol{x}$). `mor4ansys` employs moment matching via Arnoldi process [5] to construct V. The compact model is obtained by the projection of Eq (1) as follows

$$\begin{aligned} M_r\ddot{\boldsymbol{z}} + E_r\dot{\boldsymbol{z}} + K_r\boldsymbol{z} &= B_r\boldsymbol{u} \\ \boldsymbol{y} &= C_r\boldsymbol{z} \end{aligned} \tag{4}$$

where $M_r = V^T M V$, $E_r = V^T E V$, $K_r = V^T K V$, $B_r = V^T B$, $C_r = CV$. The meaning of inputs $\boldsymbol{u}$ and $\boldsymbol{y}$ outputs is the same as in Eq (1). Note that it is unnecessary to specify inputs $\boldsymbol{u}$ in order to run model reduction as the reduced system (4) is valid for any input. The transient and harmonic simulation of Eq (4) is much faster than those of the original high-dimensional Eq (1).

The approximation error clearly depends on the dimension of the vector $\boldsymbol{z}$, generally the higher the better but it does not depend on number of outputs. The reduced model is produced by an iterative procedure. This means that if we have produced the reduced model of the order 30, we can obtain all the reduced models of the lower order just by discarding the last columns in the project matrix V and, as result, the last rows and columns in the reduced matrices. This can be used in the recommended strategy to find an optimal dimension of the reduced model: to generate a reduced model of a chosen maximum dimension (30 is a good starting point) and then to compare solution predicted by reduced models with the original ANSYS simulation. Note that the original ANSYS simulation it not required for the model reduction process by itself.

The reduced matrices are written in the Market Matrix format [6] and can be easily read to any simulation software (Matlab, Mathematica, etc.). Each matrix is written in each own file. The files have names `base_name.matrix_name` where `base_name` can be specified with the option `-o` (by default `base_name` is equal to `mor`) and `matrix_name` follows convention of Eq (1) for the second-order systems or (2) for the first order systems. Additionally, file `base_name.C.names` contain text strings describing different outputs when each string is written on a separate line.

Preparing ANSYS files

Output Matrix

The output matrix is mostly for convenience. The model reduction during the Arnoldi process does not use the output matrix. There are two options in order to specify C. It is possible to have the complete output as one can restore the complete state vector by Eq (3). In this case, use option `-f` and `mor4ansys` will generate the output matrix C automatically by including all degrees of freedom. In this case, C will be equal to a permuted identity matrix in order to reorganize the state vector $\boldsymbol{x}$ to more natural node ordering. The information about on how degrees of freedom are written can be found in the file `mor.C.names`.

It may be more convenient to choose the output degrees of freedom explicitly. The second option allows this and you should write a file as follows. In the case, when there is only a single degree of freedom per node, you write a text file where each line contains

```
output_name node_number
```

where `output_name` is any text without spaces (it will be written to the file `mor.C.names`) and `node_number` is the number of the desired output node. You can use `node(x, y, z)` in your ANSYS script to find it.

When the number of degrees of freedom per node is more than one, the degree of freedom should be also specified.

```
output_name dof_name node_number
```

`dof_name` should be one of those listed in the header of the EMAT file (UX, UY, UZ, ROTX, ROTY, ROTZ, AX, AY, AZ, VX, VY, VZ, PRES, TEMP, VOLT, MAG, ENKE, ENDS, EMF, CURR).

Use option `-C` to specify the file name with the output degrees of freedom for `mor4ansys`.

System Matrices

In the current version of `mor4ansys`, the two files (FULL and EMAT) are used in order to read the system matrices. You generate them as follows:

```
/solu
allsel
antype,static
eqslv,sparse
nsubst,1
wrfull,1
ematwrite,yes
solve
fini
```

The WRFULL command is only available since ANSYS 8. It makes ANSYS write FULL and EMAT file only and not perform a real solve. In previous versions of ANSYS the command is not available. Without this command, the time to extract the system matrices will be equal to that of the stationary solution.

If the original model is nonlinear than keep in mind that the element matrices are evaluated for the current state vector. With WRFULL this operation is quite fast as ANSYS does not perform a real solution, it only writes assembled element matrices to the binary file.

A FULL file generated for the sparse solver as shown above contains the load vector, the stiffness matrix, the Dirichlet and equation constraints. Unfortunately, I have not found a reliable way to have all system matrices written in the FULL file for any case. For example, the modal analysis makes this for a mechanical system but not for a thermal one. As result, mor4ansys uses a EMAT file to read other matrices.

The FULL file should be specifed first as follows

```
mor4ansys -C outdofs.txt file.full file.emat
```

or

```
mor4ansys -f file.full file.emat
```

when you would like to have all degrees of freedom as outputs.

Multiple Load Vectors

Input matrix B is made up of load vectors. In the simplest case, it comprises a single load vector and $\boldsymbol{u}$ is a scalar input function. In this case, you do nothing else. As model reduction does not depend on $\boldsymbol{u}$, you can take a step input function in ANSYS. This produces a load vector that can be used later on for any input function.

A compact model obtained by model reduction depends on the load vector. One can change the input function but not the load vector itself. This means that the load vector can be easily scaled but its rotation is not allowed. When this is inappropriate, one can make the original model in the multiple input form and to run model reduction with several input vectors simultaneously. ANSYS allows us to specify several loads but unfortunately it saves only current load in the EMAT or FULL file. In order to generate several independent input vectors it is necessary to write several EMAT or FULL files. Note that a EMAT file may not contain the load vector in all the cases. As result, I would recommend to use FULL files.

To use this strategy you could

```
!apply first load
!generate full and emat file as written above
/filname, second
!remove first load and apply the second load
!generate emat or full file
/filename, third
!remove the previous load and apply the third load
!generate emat or full file
...
```

Then you can run provided that the system matrices are in `file.emat`

```
mor4ansys -C outdofs.txt file.full file.emat second.full third.full
```

Several rules to keep in mind.

- The first two files must be a FULL and EMAT files as explained in the previous section. `mor4ansys` read system matrices and the first load vector from them.
- Files must be consistent: they must be generated from the same model.
- If a current load vector is zero, mor4ansys skips it and does not include to the matrix B. Check mor4ansys output to understand the relationship between files and columns of B.

By default, load vectors are written to B as they are. The option `-c` may allow you to simplify the generation of the load vectors. If it is specified, mor4ansys corrects columns of B as follows. Let us assume the C numbering, that is, that the first vector is 0, the last is `NumberOfInputs - 1`. Then

```
for (int i = NumberOfInputs - 1; i > 0; --i)
```

$$\boldsymbol{b}_i = \boldsymbol{b}_i - \boldsymbol{b}_{i-1}$$

This means that one can apply the loads consecutively

```
!apply first load
!generate full and emat file as written above
/filname, second
!generate emat or full file
/filename, third
!generate emat or full file
...
```

and `mor4ansys` corrects them by assuming that each generated load includes all the previous ones.

Performing Model Reduction

The Arnoldi process [5] takes as input the two matrices, a square matrix with the dimensionality equal to the state vector $\boldsymbol{x}$, and an equivalent of the input matrix with number of columns equal to the number of different inputs. Let us denote them as R and S. In the case of a single input, S is the vector. After n iterations, the Arnoldi process generates the projection matrix V with n columns and a Hessenberg matrix H such as

$$H = V^T R V \tag{5}$$

The Hessenberg matrix is discarded after the process is completed and the matrix V is used to make a reduced model. See Eq (4) for the second order system, for the first order it is quite similar.

For the first order system $R = A^{-1}E$ and $S = A^{-1}B$. The Arnoldi process is based on matrix vector multiplication. This means that we do not have to compute the system matrix inverse explicitly and thus to obtain very efficient algorithm. The matrix inverse is replaced by a solution of system of linear equations. The system matrix is factorized once before the process and then the corresponding system of linear equations is solved quite efficiently by means of back substitution.

For the second order system, there are three choices. By default, $R = K^{-1}M$, $S = K^{-1}B$ and the damping matrix is not used during model reduction and projected afterwards [9]. This method allows us to preserve the Rayleigh damping coefficients as parameters. Option -1 transforms the second order system to the first order system before the model reduction. Option -2 chooses the use of the Second Order ARnoldi (SOAR) algorithm [10].

There are several options to control model reduction process.

-N specifies the dimension of the reduced model. By default it is 30.

-t set up the tolerance to deflate the vector. By default it is 1e-12.

-s is to choose a solver. By default it is TAUCSlltmf.

-p is the solver parameter. By default it is metis.

Default values should be good enough for most models. The main problem is that it is difficult to predict what dimension of reduced model will produce the required accuracy. In our experience, a dimension of 30 is satisfactory for many thermal and structural mechanics problems but the user should always check if this is enough.

When a new vector is generated, mor4ansys checks its norm. If it is less then the tolerance specified with -t option, it is deflated, that is, it is removed because it is assumed to represent a zero vector within rounding errors. When there are no more vectors to continue Arnoldi process, it stops.

There are many solvers to solve a system of linear equations. At present, mor4ansys uses the TAUCS

[7] and UMFPACK [8] libraries. Our recommendations are as follows.

For symmetric and positive definite matrices `TAUCSlltmf` with `metis` is the best choice. If the matrix is symmetric but indefinite try `TAUCSldlt` with `metis` is not bad although UMFPACK may be faster in this case. For non-symmetric matrices use UMFPACK. If you run out of memory, choose out-of-core solvers. 4 Gb of RAM is highly recommended for high dimensional models.

Choices available for a solvers are as follows.

- `TAUCSlltmf` - Multifrontal supernodal Cholesky decomposition.
- `TAUCSlltll` - Left-looking supernodal Cholesky decomposition.
- `TAUCSlltooc` - Out-of-core sparse Cholesky decomposition.
- `TAUCSllt` - Cholesky decomposition column by column (slow).
- `TAUCSldlt` - LDL^T factorization.
- `TAUCSlu` - Out-of-core sparse pivoting LU decomposition.
- `UMFPACK` - Multifrontal LU decomposition.

TAUCS solvers for symmetric matrices can take a parameter with the reordering method. Allowable values for a parameter are listed below. In our experience, metis is the best.

- `metis` - hybrid nested-dissection minimum degree ordering.
- `genmmd` - multiple minimum degree ordering.
- `md` - minimum degree ordering.
- `mmd` - multiple minimum degree ordering.
- `amd` - approximate minimum degree ordering.
- `treeorder` - no-fill ordering code for matrices whose graphs are trees.

More information about methods is in the library manuals [7][8].

Advanced Options

Writing Original High Dimension Model

An option `-w` allows us to write the original high-dimensional system in the Matrix Market format [6] (see [4]). The specified string that follows the option `-w` is assumed to be the base name. Each matrix is written in the file `base_name.matrix_name` where `matrix_name` follows convention of Eq (1) for the second-order systems or (2) for the first order systems. Additionally, file `base_name.C.names` contain text strings describing names of different outputs when each string is written on a separate line.

mor4ansys can perform model reduction for models written in the Matrix Market format as follows

```
mor4ansys -MM base_name
```

Specifying Expansion Points

By default, `mor4ansys` uses the expansion point zero. This can be modified with the option -x as follows

```
mor4ansys -C output.txt -x 1e5 file.emat
```

where the number after the option sets the expansion point. Note that if the expansion point is different from zero, the reduced model does not preserve the stationary state.

Additionally, one can perform a multi-point expansion. It is necessary to create a text file with each line containing

```
expansion_point number_of_vectors
```

The file name is then specified as an argument to the option `-x`.

Non-zero Initial State

The initial state for the transient simulation may be different from zero. In this case, one can compute the initial state for the reduced model by projecting the initial state x_0 to the low-dimensional subspace

$$z_0 = V^T \boldsymbol{x}_0 \quad (6)$$

This is an approximation as

$$\boldsymbol{x}_0 \neq V V^T \boldsymbol{x}_0 \quad (7)$$

If the difference between the initial state and its projection is too big, there are two solutions. The first is to increase the dimension of the reduced model. Second is to use the transformation

$$\boldsymbol{x}_{new} = \boldsymbol{x} - \boldsymbol{x}_0 \quad (8)$$

`mor4ansys` supports both methods. Eq (6) is used during postprocessing and in this case one employs `mor4ansys` as usual. In order to use Eq (8), one can choose an option -0 followed by a file name that refers to a file where x_0 is written (numbers separated by white space). In this case, the reduced model will contain an additional file with an extension `init` where the components of x_0 will be written.

Running mor4ansys: Summary

`mor4ansys` is a command-line tool. A command contain options and file names.

`mor4ansys [ options]  file_names`

A command without arguments

`mor4ansys`

lists all the options and their defaults.

Option `-f` forces `mor4ansys` to generate a complete output matrix.

Option `-o` file_name allows us to specify a base name for the reduced model. By default, it is `mor`.

Option `-0 file_name` allows us to specify the initial vector to make transformation (8).

Option `-c` forces `mor4ansys` to correct load vectors.

Option `-w` file_name allows us to write the original ANSYS model in the Matrix Market format. By default, this file is not written. Note that for high dimensional models it can take a lot of disk space.

Option `-C file_name` allows us to specify outputs.

Option `-t number` specifies a deflation criterion. When a new vector norm is less than this number, it is excluded. If this is the last vector, the whole process stops. By default, a deflation criterion is 10^{-12}.

Option `-s` allows us to specify a solver. By default, it is `TAUCS_lltmf`.

Option `-p` allows us to specify a solver parameter. By default, it is `metis`.

Option `-N number` sets a dimension for the reduced model. As was mentioned above, you are able to restore all the reduced models with lower dimensions. By default, it is set to 30.

Option `-x file_name` allows us to specify expansion points.

Option -1 forces the transformation of the second order system to a first order system.

Option -1 forces the use of the SOAR algorithm for the second order system.

Postprocessing Compact Model in Mathematica

We have functions in Mathematica to postprocess results:

- To import results from ANSYS and `mor4ansys`.
- To plot results from ANSYS and reduced models of different dimensions.
- To compute an error between results produced by reduced models and ANSYS.
- To choose the most appropriate reduced model.
- To make transient and harmonic simulation with different input functions.

The functions are available at http://evgenii.rudnyi.ru/programming.html#post4mor.

References

[1] E. B. Rudnyi, J. G. Korvink, *Automatic Model Reduction for Transient Simulation of MEMS-based Devices*, Sensors Update, 2002, 11, 3-33.

[2] J. G. Korvink, E. B. Rudnyi, *Model Order Reduction of MEMS for efficient computer aided design and system simulation*, Sixteenth International Symposium on Mathematical Theory of Networks and Systems, Belgium, July 5-9, 2004. http://www.imtek.uni-freiburg.de/simulation/mor4ansys/

[3] E. B. Rudnyi, J. G. Korvink. *Model Order Reduction for Large Scale Engineering Models Developed in ANSYS*. PARA'04, Workshop On State-of-the-art In Scientific Computing, Technical University of Denmark, June 20-23, 2004. http://www.imtek.uni-freiburg.de/simulation/mor4ansys/

[4] Oberwolfach Model Reduction Benchmark Collection, http://www.imtek.uni-freiburg.de/simulation/benchmark/.

[5] R. W. Freund, *Krylov-subspace methods for reduced order modeling in circuit simulation*, Journal of Computational and Applied Mathematics vol. 123, pp. 395–421, 2000.

[6] R. F. Boisvert, R. Pozo, K. A. Remington, The Matrix Market Exchange Formats: Initial Design, NISTIR 5935, http://math.nist.gov/MatrixMarket/

[7] S. Toledo, D. Chen, V. Rotkin, TAUCS – A library of sparse linear solvers, http://www.tau.ac.il/~stoledo/taucs/

[8] T. A. Davis, UMFPACK, http://www.cise.ufl.edu/research/sparse/umfpack/

[9] E. B. Rudnyi, J. Lienemann, A. Greiner, and J. G. Korvink. mor4ansys: Generating Compact Models Directly from ANSYS Models. In Technical Proceedings of the 2004 Nanotechnology Conference and Trade Show, Nanotech 2004, March 7-11, 2004, Bosten, Massachusetts, USA.

[10] Z. J. Bai, K. Meerbergen, Y. F. Su. Arnoldi methods for structure-preserving dimension reduction of second-order dynamical systems. In: P. Benner, G. Golub, V. Mehrmann, D. Sorensen (eds), Dimension Reduction of Large-Scale Systems, Lecture Notes in Computational Science and Engineering. Springer-Verlag, Berlin/Heidelberg, Germany, 2005.