Formal Definition of Model Reduction

Before solving, ANSYS discretizes the computational domain. This includes not only meshing but also creating a system of equations to solve – first for each element, and then for the complete domain. In the case of a transient analysis of a second order system, the resulting ordinary differential equations are as follows:

\[ M\dot{x} + E\dot{x} + Kx = f \]  
\[ \text{(1)} \]

Eq (1) is shown schematically in the upper part of Fig 2. There are three global matrices, the mass matrix \( M \), the damping matrix \( E \) and the stiffness matrix \( K \), additionally the state vector \( x \), that includes all degrees of freedom (one node could have several degrees of freedom, for example in structural mechanics UX, UY, and UZ), and the load. The goal is to reduce the number of degrees of freedom in Eq (1) in the state vector (and hence in the matrices), in order to solve it fast at system level. For a thermal problem, the mass matrix does not exist \( (M = 0) \), the damping matrix is the heat capacity matrix, the stiffness matrix is the heat conductivity matrix, and, in this case, the number of nodes is equal to the number of degrees of freedom (a node has just TEMP).

In the last 20 years, this problem has been extensively researched by mathematicians (see Fig 3). Before we consider this, let us introduce in Eq (1) inputs and outputs, as these terms are not present directly at the level of the finite element modeling. Let us start with the load that could be time dependent \( f(t) \). The goal would be to split it to a combination of constant vectors \( b_i \) and time dependent functions \( u(t) \)

\[ f(t) = \sum b_i u(t) = Bu \]  
\[ \text{(2)} \]

The constant vectors transfer the time dependent function to different degrees of freedom. For example, a time dependent force on a surface could be expressed as a function in time that acts on nodes in the surface and the vector bi distributes this force accordingly. When we need to have several independent time dependent functions at system level, we introduce several input functions, and finally we have an input matrix \( B \) and a vector of scalar input function \( u \) (see Fig 2, bottom left side).

At system level, we are not interested in the complete state vector \( x \), but rather we need to know just a part of the information from it and this could be defined by means of the output matrix \( C \)

\[ y = Cx \]  
\[ \text{(3)} \]

where the vector \( y \) collects only information that we really are interested in (see Fig 2, bottom right). The notation of inputs and outputs allows us to rewrite Eq (1) as follows

\[ M\dot{x} + E\dot{x} + Kx = Bu \]  
\[ \text{(4)} \]

Now the formal model reduction could be defined as a mathematical problem to reduce the dimension of \( x \) while preserving the dynamical relationships between inputs \( u \) and outputs \( y \) within prescribed accuracy.

Model Reduction as Projection

Formally speaking, all degrees of freedom in the state vector \( x \) in Eq (4) are independent and it is impossible just to eliminate some of them based on pure mathematical manipulations. Theoretically, one could reduce degrees of freedom in \( x \) by introducing its higher derivatives, that means, by increasing the number of system matrices. However, this does not reduce the computational complexity, it even brings more computational problems. Thereafter, model reduction is always some approximation.
In Fig 3 there are different methods for model reduction. All of them share the idea of a projection that is expressed as follows (see also Fig 4, top)

\[ x = Vz + e \] (5)

Eq (5) states that, provided the error vector is negligible, the state vector \( x \) can be formally described by a few degrees of freedom \( z \). Mathematically one speaks about a low dimensional subspace that captures the system dynamics. Provided we neglect the error, the state vectors \( x \) during the transient simulation remain in the low dimensional subspace expressed by the matrix \( V \). If so, one can project Eq (4) on that subspace defined by Eq (5) and obtain the reduced model as follows

\[ V^T MVz + V^T EVz + V^T KVz = V^T Bu \] (6)

(see Fig 4 bottom). The input function vector \( u \) and the output vector \( y \) are the same in Eq (6) and in Eq (4) with an exception of the approximation error between \( y \) in both equations:

\[ |y_r - y| = \delta \] (7)

where \( y \) are outputs in the original model and \( y_r \) are in the reduced one.

Now let us consider Fig 3 where model reduction methods are classified according to Prof. Antoulas [7]. From a theoretical viewpoint, the best are the Singular Values Decomposition (SVD)-based methods (Fig 3 left), as they have global error estimates and produce an optimal reduced model where the approximation error is minimal. It is worth noting, that although model reduction may look like an optimization method to minimize an approximation error, then the required dimension of a reduced model is chosen based on the global error estimates. Unfortunately, computational time in this case grows cubically with the dimension of the state vector. Practically, it means that with modern hardware the number of degrees of freedom to reduce is limited to ten thousands. SVD-based methods with better scaling computational properties (Fig 3, right) are still under development and, at the moment, implicit moment matching (Fig 3, in the middle) is the only option that can be employed for industrial applications right now.

The idea of moment matching is to transform the dynamic system (4) into the Laplace domain, and then to find such a low-dimensional system that has the same first derivates in the Taylor expansion around some point as the original model.

The direct implementation of this idea is numerically very unstable but mathematicians have found that, by means of the generation of a particular Krylov subspace, one finds such a projection subspace that the reduced model definitely matches first moments. The drawback of moment matching is the absence of global error estimates and some engineering Know How is required to determine the dimension of the reduced model.

**Engineering Model Reduction**

Interestingly enough that mathematicians (see for example [7]) hardly consider model reduction methods already available in ANSYS, such as mode superposition, Guyan reduction, and component mode synthesis (CMS). The reason is that these methods are not based on a strict mathematical theory but rather on engineering intuition. To this end, it is very instructive to compare Antoulas’ book [7] with a ty-
tional book written by a mechanical engineer [8]. While one finds theorem and their proofs in [7], the book [8] looks more like a recipe book. Mathematicians develop model reduction in the general case of any dynamical system but methods from mechanical engineers are applicable to structural mechanics only.

Said that, I would like to state that in structural mechanics the methods developed by mechanical engineers are applicable to structural mechanics only.

Let us look at Fig 5 where mode superposition is compared with the Arnoldi-based model reduction. In Fig 5 (bottom, left), the harmonic response of a hard disk drive actuator/suspension system is compared for the original model in ANSYS, the reduced model of the dimension 80 by MOR for ANSYS, and mode superposition reduced model (80 modes have been used). The difference between three models is smaller than a line thickness and to see the difference, the relative error between reduced models and the original model is presented in Fig 5 bottom right. One can clearly see that mathematicians apply model reduction better than mechanical engineers yet, on the other hand the approximation error of a mode superposition model is already good enough from an engineering viewpoint (less that 1%).

Thereafter, the next strategy could be recommended. If you are working with a mechanical model, a good starting point would be the model reduction already available in ANSYS. In this case, everything is available within a single environment and it is necessary just to employ it in practice. The command SPMWRITE (available also under GUI in Mechanical) creates a Simulor model directly after mode analysis. Alternatively MOR for ANSYS (a part of the CADFEM Toolbox) can be used in case of other ANSYS models (thermal, piezoelectric, etc.). In any case by means of model reduction, one can bring a finite element model into system level analysis (see for example Fig 6 in case of a battery pack [4]).

**InfoAuthors**
Dr. Tamara Bechthold, Albert-Ludwigs-University, Freiburg i.Br., Department of Microsystems Engineering IMTEK
Dr. Evgeny Rudnyi, CADFEM GmbH

**InfoContact**
Dr. Evgeny Rudnyi, CADFEM GmbH
Tel. +49 (0)8092-7005-82
erudny@cadfem.de

**InfoReferences**

**MoreInformation**
http://ModelReduction.com