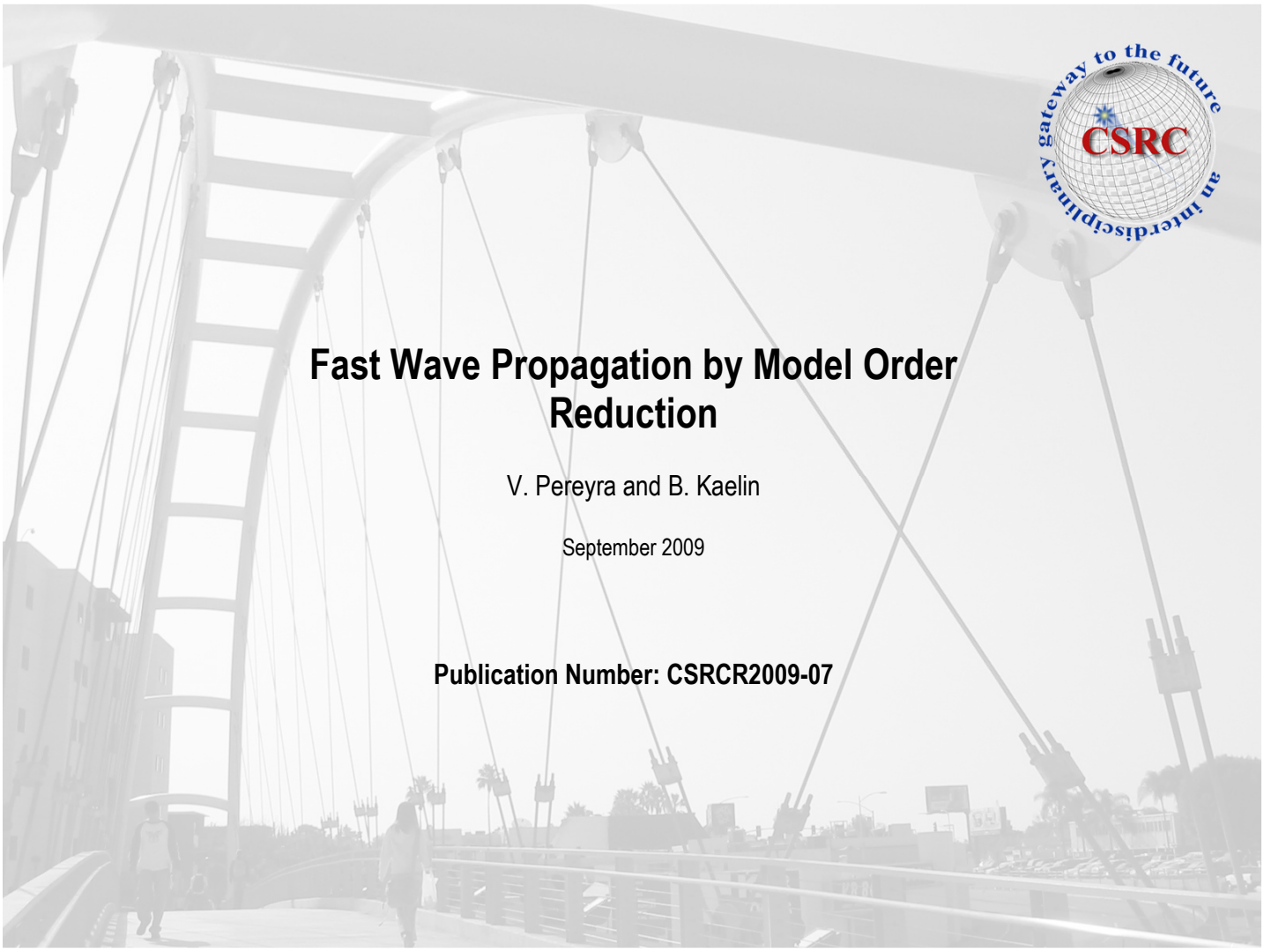


Fast Wave Propagation by Model Order Reduction

V. Pereyra and B. Kaelin

September 2009

Publication Number: CSRCR2009-07

Computational Science & Engineering Faculty and Students Research Articles

Database Powered by the Computational Science Research Center Computing Group

COMPUTATIONAL SCIENCE & ENGINEERING

SAN DIEGO STATE UNIVERSITY

Computational Science Research Center
College of Sciences
5500 Campanile Drive
San Diego, CA 92182-1245
(619) 594-3430

FAST WAVE PROPAGATION BY MODEL ORDER REDUCTION

V. PEREYRA[1] and B. KAELIN[2]

Abstract. Large scale wave propagation simulation is currently achievable in reasonable turnaround times by using distributed computing in multiple cpu clusters. However, if one needs to perform many such simulations, as is the case in optimization, tomography, or seismic imaging, then the resources required are still prohibitive. Model order reduction of large dynamical systems has been successfully used in several application domains to paliate that problem and in this paper we explore one of its manifestations, Proper Orthogonal Decomposition, for wave propagation. We describe the method and show how it can be easily interfaced with two different high fidelity simulators. We exemplify its use on several problems of increasing complexity and size.

Key words. Wave propagation; Model Order reduction; Proper Orthogonal Decomposition;
AMS subject classification. 65M99

1. Introduction. There are many applications that require the repeated transient simulation of acoustic, elastic or electromagnetic wave propagation. To name a few: structural analysis, blast on structures, vibrations of Navy vessels, sonar, design of piezoelectric transducers for medical ultrasound, medical imaging and therapeutics uses of ultrasound, earth seismic imaging for the Oil Industry and Earthquake Seismology, Optimization driven by Simulation for material identification and optimal design. As such, any significant improvement in the performance of numerical simulators would be very important.

Model Order Reduction (MOR) refers to a collection of techniques to reduce the number of degrees of freedom of the very large scale dynamical systems that result after space discretization of time-dependent partial differential equations. Some of these techniques have been successfully employed in the simulation of VLSI circuits, computational fluid mechanics, real-time control, heat conduction and other problems [1, 3, 5, 7]. Not much has been done for wave propagation, although it does not seem that there are fundamental difficulties for its application [2].

However, since none of these techniques are trivial to interface with existing large scale high fidelity codes, it is important to be able to select wisely the correct approach in order to minimize development costs. At this time we have centered our attention on the class of methods that go by the name of **Proper Orthogonal Decomposition (POD).** We start from the premise that it is possible to run a few full simulations within the domain of interest. POD uses snapshots from these simulations to form an orthogonal basis for the solution space. This can be thought of as a problem-dependent modal decomposition, as opposite to the use of artificial basis functions (Fourier expansions, wavelets). By using truncated Singular Value Decompositions it is possible to reduce even further the size of this basis without sacrificing accuracy and also to prevent the introduction of high frequency noise. The dynamic behavior of a new problem is calculated by solving projected collocation equations for the time dependent coefficients of a linear combination of the natural basis functions.

A different class of methods, tailored to problems where even a few high fidelity simulations are not an option, is based on Krylov subspace machinery for large-scale matrix computations [5]. These methods generate reduced-order models that are in a certain sense optimal, directly from the large-scale data matrices describing the given linear system. Interfacing these techniques with high-fidelity codes is less trivial, and would require major modifications. Therefore, we will focus first on POD-type methods. In a later stage we will explore hybrid approaches that combine the easy of use of POD methods with the powerful approximation properties of Krylov subspace-based order reduction.

We show numerical results in one and two dimensions displaying compression rates from 701 to 34,222 and with overall acuracy between 1% and 10%.

2. Model Order Reduction. The purpose of Model Order Reduction (MOR) is to replace a large dynamical system by a smaller one that still captures the dynamics of interest with sufficient accuracy. For wave propagation, when is possible to perform some high-fidelity calculations using existing finite difference or finite element codes, the approach that we will discuss here is called Proper Orthogonal Decomposition (POD), the Karhunen-Loeve Transform, Principal Components Analysis or, in more modern terms, the

[1]Weidlinger Associates Inc., 399 W. El Camino Real, Suite 200 Mountain View, CA 94040 USA. vpereyra@yahoo.com
[2]3DGeo Corp., 4633 Old Ironside Dr. Suite 401, Santa Clara CA 95094 USA. bruno@3dgeo.com

Singular Value Decomposition. This technique will allow us to analyze a complex spatio-temporal dynamic behavior and extract from it a (small) set of dominant components (data driven modes), separating them from noise and inessential underlying dynamical behavior, while still giving a sufficiently accurate description of the dynamics of interest.

It is similar to a mode analysis using Fourier, wavelets or other artificial bases, but in the approach under discussion we will use snapshots extracted from a number of high-fidelity simulations that have appropriate inputs, in order to extract the most important problem specific modes. The ideal application is one in which we have a parametrized model that needs to be calculated many times, such as in optimization, inversion, parametric studies, multiple inputs or source wavelets.

Another important application occurs in imaging using full wave solvers and reflection data. This requires forward simulation from the sources and integration backward in time from the receivers. Snapshots of these two calculations need to be correlated to form an image in space of the materials sensed by the imaging process. That requires generating, keeping and accessing a large number of very large 3D snapshots. This is done currently in massively parallel super-computers and requires considerable network traffic that slows down the process. An intriguing possibility is to perform the model order reduction described below, employing a smaller number of snapshots and using the reduced system to generate the finer mesh of snapshots on the fly. This application would require only one simulation per source and if the snapshots can be taken reasonably far apart, POD would considerably reduce network traffic and access to secondary storage (the two weak components of large distributed systems, which improve considerably slower than what Moore's law postulates).

The procedure consists of the following steps:

- A pre-processing step in which a few large scale high-fidelity calculations are performed. In all the examples below we have used just one simulation to extract snapshots.
- An SVD of the matrix whose columns are spatial snapshots extracted from those simulations is calculated and truncated at the required error level.
- The space-time approximate solution is written as a linear combination of the k selected modes (left singular vectors) with (unknown) **time dependent** coefficients.
- This Ansatz is replaced in the original equations in a Ritz-Galerkin collocation approach and due to the orthogonality of the modes, a reduced system of ODE's will result. Solving for the coefficients of the linear combination for a problem with new inputs, a very economical procedure results - compared to the original high-fidelity calculation.

3. Model Order Reduction by Proper Orthogonal Decomposition. Let us consider a first-order hyperbolic system already discretized in space:

$$\begin{aligned} w' &= Aw + Bu(t), \\ v &= Cw, \end{aligned} \tag{3.1}$$

where $w(t), B \in R^M$ and A, C are appropriate matrices and ' represents time differentiation. Matrix A is sparse in the finite element or finite differences case, but full if an spectral method is used. The vector u contains the inputs (forcing function, time dependent boundary conditions), while the vector v contains the desired outputs (for instance, seismograms at a few locations). The vector B distributes the time dependent forcing function over the desired spots in the spatial mesh. For the state vector w, M is the number of degrees of freedom in space, generally very large.

We assume that we either can observe (measure) the system for various inputs at different times or that we can numerically simulate it. Let $\Phi = \{\phi_i\}$ $i = 1, ..., l$ $(l << M)$, be the $M \times l$ matrix whose columns are these spatial snapshots, and let $\Phi = U\Sigma V^T$ be its Singular Value Decomposition, where U, V are orthogonal matrices and Σ contains the singular values σ_i in its diagonal, sorted in descending order of magnitude. Since the vectors in U, V have norm l_2 equal to 1, the Frobenius norm is given by the sum of squares of the singular values:

$$E^2 = \sum_{i=1}^{l} \sigma_i^2.$$

If we truncate the SVD at the mth term, with $m \leq l << M$ then the error (or left-over energy) is:

$$\delta_m^2 = \sum_{i=m+1}^{l} \sigma_i^2.$$

Thus if we want to preserve a certain fraction of the total information, say $0 < p \leq 1$, then m must be chosen so that:

$$\delta_m^2 \cong (1 - p^2)E^2.$$

Let the truncated set of left singular vectors of Φ be called U_m. We now seek solutions of system (3.1) (with the same spatial discretization), of the form:

$$w(x,t) = U_m \; a(t), \tag{3.2}$$

where $a(t)$ is a vector of time dependent coefficients of dimension m to be determined. The coefficients $a(t)$ for a new input are determined via Galerkin collocation. We replace in system (3.1) the Anzatz (3.2), obtaining:

$$\begin{aligned} U_m \; da/dt &= AU_m \; a(t) + Bu(t), \\ v &= CU_m \; a(t). \end{aligned}$$

Multiplying by U_m^T the differential equation and since the columns of U_m are orthogonal, we get:

$$\begin{aligned} da/dt &= U_m^T A U_m \; a(t) + (U_m^T B) \; u(t), \\ v &= (CU_m) \; a(t), \end{aligned} \tag{3.3}$$

which is the reduced system of ODE's of dimension m, whose solution will produce the time dependent coefficients $a(t)$. Combining these coefficients with the spatial modes U_m as in (3.2) produces the full solution for a new problem. The matrix of the reduced system $A_m = U_m^T A U_m$, is not sparse.

Summary. The steps to follow then are:

1. Run s full simulations with the same spatial mesh (for instance, changing the source location).
2. Extract k snapshots from each simulation, for a total of $l = k * s$ columns in Φ.
3. Calculate the SVD of Φ (complexity of the SVD for a $M \times l$ matrix is $O(M \times l^2)$).
4. Truncate at level $p < 1$.
5. With the resulting m modes construct the matrices of the reduced system:

$$A_m = U_m^T A U_m, \quad B_m = U_m^T B(x), \quad C_m = C(x) U_m.$$

6. To solve a new problem (say with the source at a different position, or a diferent input source), we solve the reduced systems of ODE's for the coefficients $a_j(t), j = 1, ..., m$, in the representation 3.2 of the solution. Of course, we can also solve the same problem, with the object of producing intermediary time snapshots between the selected sparse ones.
7. To obtain the solution in the original space do: $w = Ua$.
8. Validation: compare reduced results with full high fidelity results (at the sensors!).

Comments. In the previous algorithms there are some undetermined quantities, namely: the number of full simulations s, the number of snapshots b and the level p. A possible way of deciding the proper number of simulations and snapshots (besides some experimentation) would be to start with $s = 1$, and increment it if necessary. A good indicator that we have enough snapshots would be when small (normalized, i.e., divided by the largest one) singular values start showing. Using an updating algorithm for the successive SVD's would be an efficient way to proceed [6].

Since the real expense is in the simulation, one can take b reasonably large to start with, and let the SVD analysis decide if some snapshots are not contributing information to the reduced transfer function. In this way there is no *a priori* guess and we would stop as soon as there is enough information content in our data set of snapshots.

The use of a high-order method provides already a beneficial reduction in the initial number of spatial degrees of fredom (by a factor of up to 10,000 in 3D, when compared with a second order finite element method). For realistic problems, the original system will still be too large and too time consuming for wholesale real-time simulation, and thus we need to be able to speedup the calculation further by using these order reduction techniques.

4. Example: Scalar Wave Equation . We consider as a simple test problem to validate these ideas, the 1D scalar wave equation in a semi-infinite homogeneous half space, writen in first order form:

$$\begin{aligned} v' &= \rho^{-1} p_x, \\ p' &= K v_x, \end{aligned}$$

where v, p are the velocity and the vertical component of the stress, respectively, while ρ, K are density and the bulk modulus respectively. The initial and boundary conditions are:

$$\begin{aligned} v(0,x) &= 0, \\ p(0,x) &= 0, \\ p(t,1) &= 0, \\ p(t,0) &= Ricker(t), \end{aligned}$$

where the forcing function is a 50 Hz Ricker wavelet with amplitude 1. We take for this experiment, $\rho = 2000\ k/m^3,\ c_p = 3000\ m/s,\ K = 1.8 \times 10^{10} P$.

Once the problem is discretized in space (on a staggered mesh using second order accurate centered differences) we obtain the following block structure:

$$\begin{bmatrix} w_1' \\ w_2' \end{bmatrix} = \begin{bmatrix} 0 & A_{12} \\ A_{21} & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} + 2/\rho\ B\ R(t), \tag{4.1}$$

where the vectors w_1, w_2 contain the discretized values of v *and* p respectively, $A_{12},\ A_{21}$ are bi-diagonal and B is a vector with all zeroes except for the first component that is equal to 1. The 2 in the forcing term comes from the top and bottom free surface conditions. To advance in time we use leapfrog, assuming that w_1 is available at t and w_2 is available at $t + dt/2$:

$$\begin{aligned} w_1(t+dt) &= w_1(t) + dt\ A_{12} w_2(t+dt/2), \\ w_2(t+3dt/2) &= w_2(t+dt/2) + dt\ A_{21} w_1(t+dt). \end{aligned}$$

4.1 is the full system of ODE's that we want to reduce. Due to the special structure it is convenient to continue the reduction in block form. Thus, let Φ_1, Φ_2 be the matrices of snapshots for v, p respectively, and let

$$\Phi_1 = U_1 \Sigma_1 V_1^T,\ \Phi_2 = U_2 \Sigma_2 V_2^T,$$

be their Singular Value Decompositions. Introducing the Ansatz:

$$w_1 = U_1 a_1(t),\ w_2 = U_2 a_2(t)$$

and replacing in the differential equation, after some additional manipulations we obtain the reduced system:

$$\begin{bmatrix} a_1' \\ a_2' \end{bmatrix} = \begin{bmatrix} 0 & U_1^T A_{12} U_2 \\ U_2^T A_{21} U_1 & 0 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} + 2/\rho \begin{bmatrix} U_1^T B_1 \\ 0 \end{bmatrix} R(t).$$

Observe that we can choose a different number of modes for each of the two sets of variables.

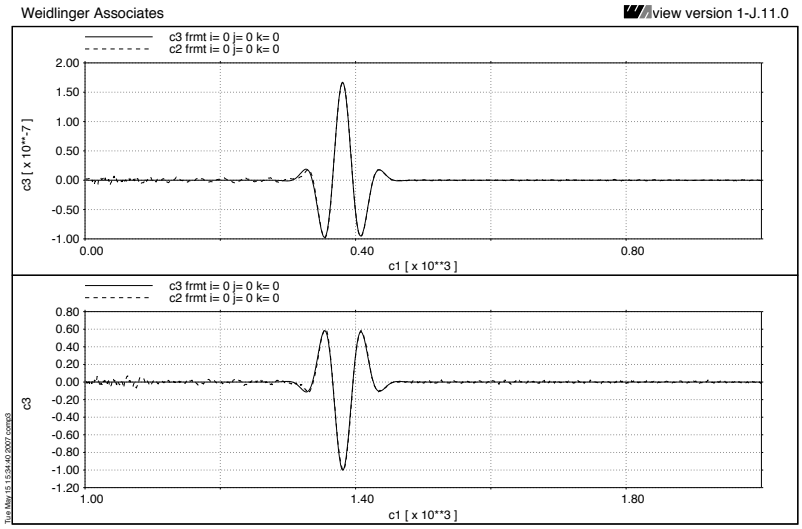


FIGURE 5.1. *Comparison of FLEX (solid curves) and MOR (dashed curves) results. Ricker source at top (left end), t=0.4125. Top figure: velocity; bottom figure: vertical component of the stress.*

5. Numerical Results. We run the second order finite elements code FLEX for 5000 time steps, with $\delta t = 0.00033$, corresponding to a CFL condition of 0.99 for the problem above and collect 100 equally spaced time snapshots. FLEX uses leapfrog, a second order explicit integrator in time and essentially symmetric differences (on an staggered mesh) in space. For the reduced system we use as time integrator the code SVODE of Brown, Hirschman and Byrne [4] in its stiff option.

The first experiment simply tries to reproduce the results of FLEX by solving the same problem but with the reduced system. In Figure 5.1 we cross-plot the results of the 2 codes for a snapshot at the 1250th time step. The results are good to eye-ball accuracy. Observe that the two sets of variables differ in about 7 orders of magnitude.

In the second experiment (Figure 5.2) we solve the reduced system with a Ricker source at $x = 500$, with a frequency of 40 Hz and amplitude equal to 2 and show the snapshot at the 750th time step. We still cross-plot with the results for FLEX with the original source in order to verify visually if there are changes in wave form or amplitude. Now we see wave pulses propagating in both direction from the center for the velocity component, some extraneous results for the vertical component of the stress and substantial high frequency noise. Observe that the expected vertical stress amplitude is still 1, because of the way in which we apply this forcing function.

Finally, we repeat the second experiment but taking only 66 left singular vectors (i.e., we drop the 34 vectors associated with the smallest singular values, see Figure 5.3). Now, as hoped, we get much cleaner results and the system has 132 variables instead of 2000, a factor greater than 15 order reduction! (Figure 5.4).

These results will not be totally surprising to anyone familiar with least squares fitting. The bad results obtained when using too many basis functions are just another manifestation of the phenomenon of over-fitting; i.e., we are approximating very faithfully spurious noise and amplifying it as we integrate along. Thus, it is doubly beneficial to filter out these highly oscillatory modes associated with the small singular values, since we also get an additional reduction in the size of the problem, i.e., enhanced data compression plus high frequency noise filtering.

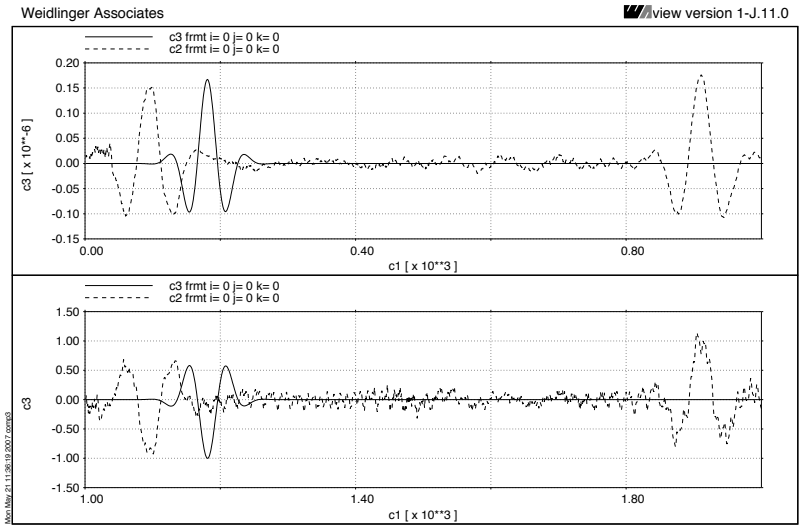


FIGURE 5.2. *Source for MOR (dashed curves) at x=500; Ricker wavelet, frequency = 40 Hz, amplitude = 2. FLEX results (solid curves) are still for problem with source at x=0; they are shown only for wave shape comparison.*

6. Wave Equation in Second Order Form. Some solvers keep the equations in second order form:

$$w_{tt} = v^2 \triangle w + Bu(t),$$

or, after space discretization:

$$w'' = Aw + Bu(t). \tag{6.1}$$

We follow the same procedure, by proposing the Ansatz $w(x,t) = U_m a(t)$, and replacing in the differential equation 6.1 obtaining:

$$a''(t) = U_m^T A U_m a(t) + U_m^T B u(t).$$

Introducing the auxiliary variables: $a_1(t) = a(t)$; $a_2(t) = a'(t)$, then we obtain the first order form of the reduced equations for use in a conventional ODE integrator:

$$\begin{aligned} a_1' &= a_2, \\ a_2' &= R a_1 + U_m^T B u(t), \end{aligned}$$

where $R = U_m^T A U_m$.

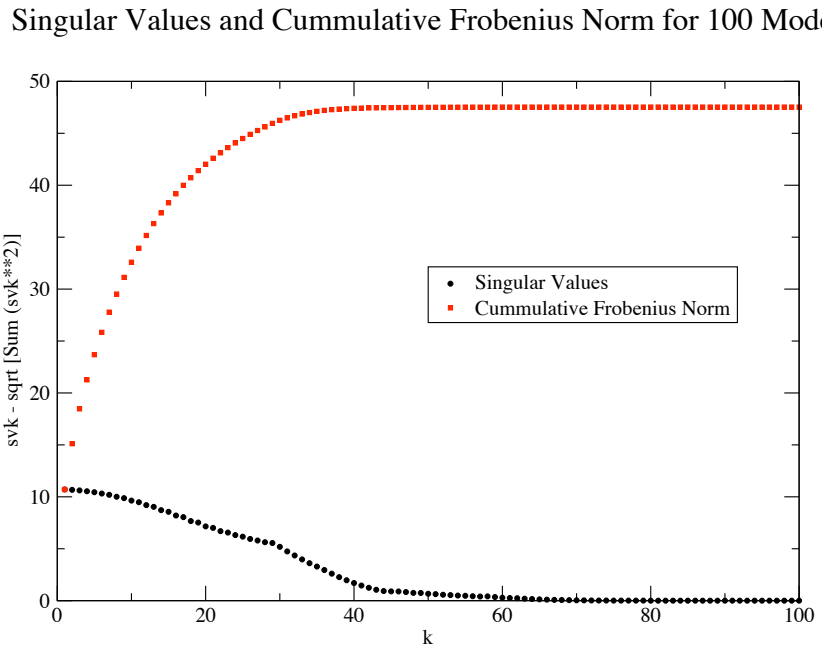


FIGURE 5.3. *Singular values and cummulative Frobenius norm*

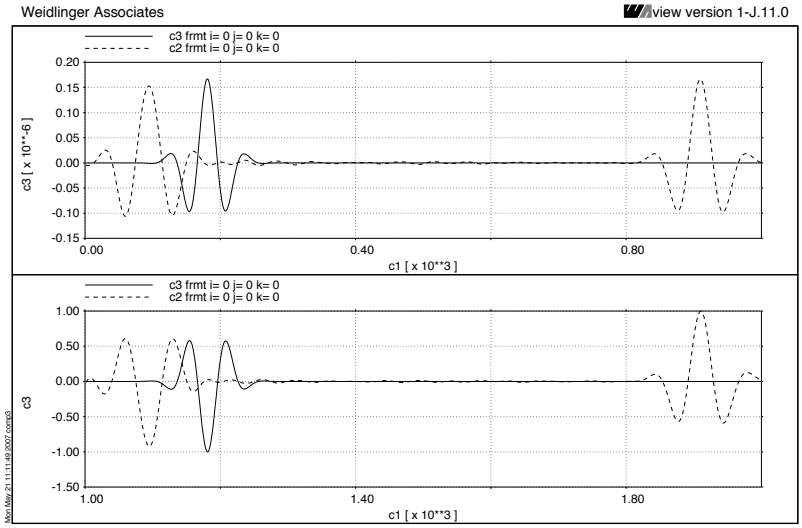


FIGURE 5.4. *Same as in Figure 5.2, but MOR uses only 66 modes, instead of 100*

$\frac{-1}{560}$	$\frac{8}{315}$	$\frac{-1}{5}$	$\frac{8}{5}$	$\frac{-205}{72}$	$\frac{8}{5}$	$\frac{-1}{5}$	$\frac{8}{315}$	$\frac{-1}{560}$

TABLE 7.1
Coefficients for 8th order aproximation to the second derivative

7. Interfacing with a high order finite difference solver. We consider now a finite difference solver using a high order spatial approximation. This is a a valuable approach when either high accuracy or long integrations are required, but also as a way to decrease the number of spatial degrees of freedom required, a significant problem in 3D.

For nodes in the interior of the mesh (i.e., at least 4 nodes away from any boundary), we shall use an 8th order approximation in the space finite difference solver. The discretization formula for the second spatial derivatives is centered and symmetric:

$$[u_{xx}]_{i,j,k} \simeq \sum_{l=-4,4} a_l u_{i+l,j,k}/\delta x^2,$$

where the coefficients a_k are given in Table 7.1 [9, 8]:

The approximation is used for all the coordinate directions. For instance, for the Laplacian in 3D we would have:

$$\Delta u(x_i, y_j, z_k) \simeq v(x_i, y_j, z_k) \sum_{l=-4,4} a_k [u_{i+l,j,k}/\delta x^2 + u_{i,j+l,k}/\delta y^2 + u_{i,j,k+l}/\delta z^2],$$

where v is the velocity of propagation.

In order to compare the accuracy of the reconstruction we will integrate the reduced system so as to reproduce all the time steps of the high fidelity code (which are provided and from where we extract one every $nsjump$ ones for the reduction phase). Thus, if Ψ is the $M \times k$ matrix of snapshots, corresponding to k integration steps with the high fidelity code, then Φ $(M \times l)$ are the selected snapshots, with $l = k/nsjump$.

The ultimate purpose of this experiment is to assess the possibility of using the reduced system to reconstruct on the fly non-selected snapshots. Since we don't have control on the internal stepsize in the ODE integrator there is a strong possibility that the discrete source wavelet will need to be evaluated at times other than $j * dt$ and therefore we will use linear interpolation as needed.

8. Complexity. The above algorithm can help in two respects:

- Reduced computation when solving many similar problems;
- Reduced storage.

In order to get a feeling for its potential in these two aspects we will make some storage and flop counts to compare the direct (D) and reduced (R) approaches.

We assume that we have M discrete state variables (spatial mesh), T total number of time steps, and that m snapshots are used for reduction ($m << T$).

Work for D, 1 time step: $O(M)$.

Work for R, 1 time step: $O(m^2)$, since reduced system is not sparse.

Work for reduction (pre-process): SVD of $M \times m$ matrix: $O(Mm^2)$.

Reconstruction of 1 snapshot $w(t) = U_m a(t)$: $O(Mm)$.

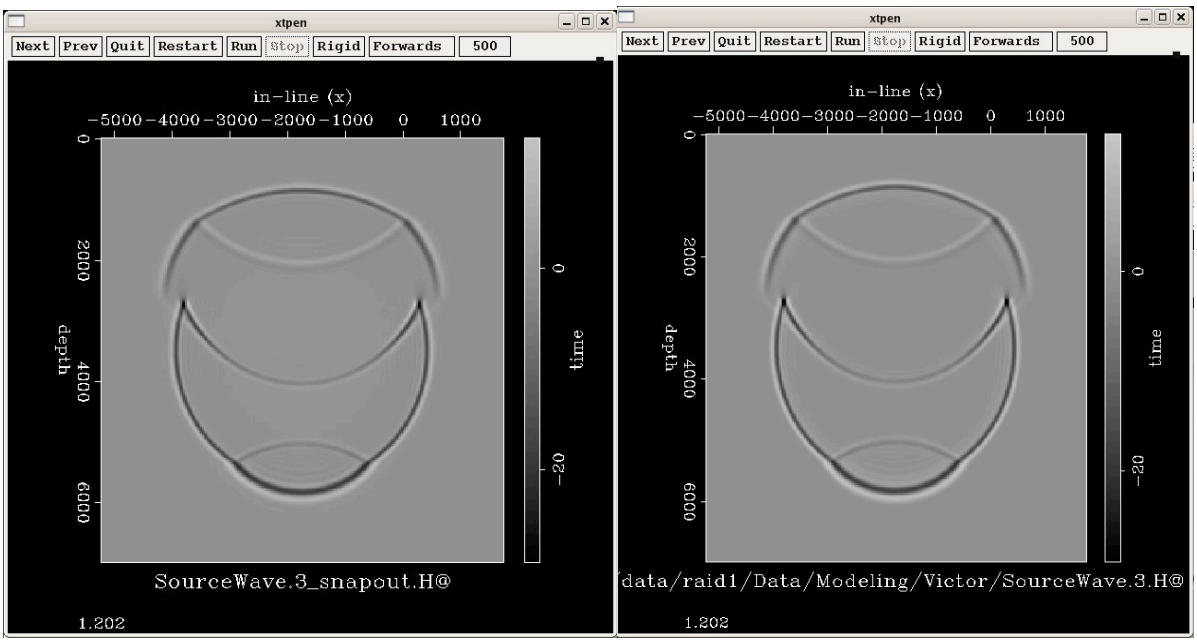
Storage for T full snapshots: $T * M$ words.

Storage for m reduced snapshots: $m * M$ words.

We show in Table 8.1 the cost of computation in flops and storage in words for two possible approaches to the calculation of a number of snapshots for forward modeling with the reduced system to be used for wave equation imaging. The first one assumes that the calculation for all desired snapshots is done a priori and the corresponding coefficients $a(t)$ are saved and then used to reconstruct a particular snapshot. In the second approach, in order to reconstruct a snapshot we integrate the reduced system using as initial conditions the closest saved snapshot (for the computational cost we consider the worst possible case, when me need to integrate essentially to the next saved snapshot (T/m time steps)). By choosing to integrate from the closest snapshot (i.e., backwards if necessary) we can halve that count. In both cases we assume that only the restricted set of m snapshots is available, for an additional storage cost of $M * m$ words.

Modality	Computation	Storage	Reconstruction (one snapshot)
1. Pre-calculate and save	$T * m^2$ (pre-process)	$T * m$	$M * m$
2. Calculate on the fly	$T/2$	m^2	$M * m$

TABLE 8.1
Two possible approaches



(a) Reduced order 3 layer model (b) Exact calculation for 3 layer model

FIGURE 9.1. *Propagation through a 3 layers model*

9. 2D Numerical Results. We consider a simple 2D problem to start: propagation in a square homogeneous domain with a source term (Ricker wavelet) applied in the center. To simplify even further we will stop the propagation before the signal reaches the boundaries, so that we do not have to worry about boundary conditions at this stage. Specifically we have the following uniform mesh:

$$nx = 261,\ ny = 261,\ dx = dy = 26.8,\ dt = 0.0024,$$

and we run for 550 time steps, extracting field snapshots every fourth step. Using a threshold of 0.9999999 the program selects 86 modes, which corresponds to the number of degrees of freedom of the reduced system. This is a reduction in dimensionality of $68121/86 = 791$. Once the dynamics is calculated the reconstructed approximate field is essentially identical to the high fidelity calculation, showing no visible dispersion.

9.1. A three layer example. Then we consider a model with three horizontal constant velocity layers, leaving everything else the same as above. The results are entirely similar, although now, for the same level of energy we require only 82 modes, for a compression ratio of 831. The results are shown in Figures (9.1 a,b). We also explore for this model what happens when one uses different number of snapshots for the analysis (the original set of 550 snapshots is decimated by taking one out of nsjump, for various values of nsjump). The results are shown in Table 9.1.

9.2. A large scale inhomogeneous velocity problem. Now we consider a portion of a real model with variable velocity involving a (high velocity) salt body Figure (9.2). This is a much larger problem with a complex heterogeneous velocity, involving a 2001×2001 points mesh with an spacing of 26.8 m. The source is still applied in the center of the model. The original solver produces 251 snapshots spaced at 0.0024 sec (that requires 4.02 Gbites of memory!). We use for the reduction one out of every two (126 snapshots) and

#snap	rank	nsjump	Compression ratio	result
275	82	2	831	OK
139	71	4	959	OK
69	63	8	1081	OK
55	50	10	1362	OK but starts showing some oscillations ahead of front
35	32	16	2129	It does not work

TABLE 9.1
Running for different number of snapshots

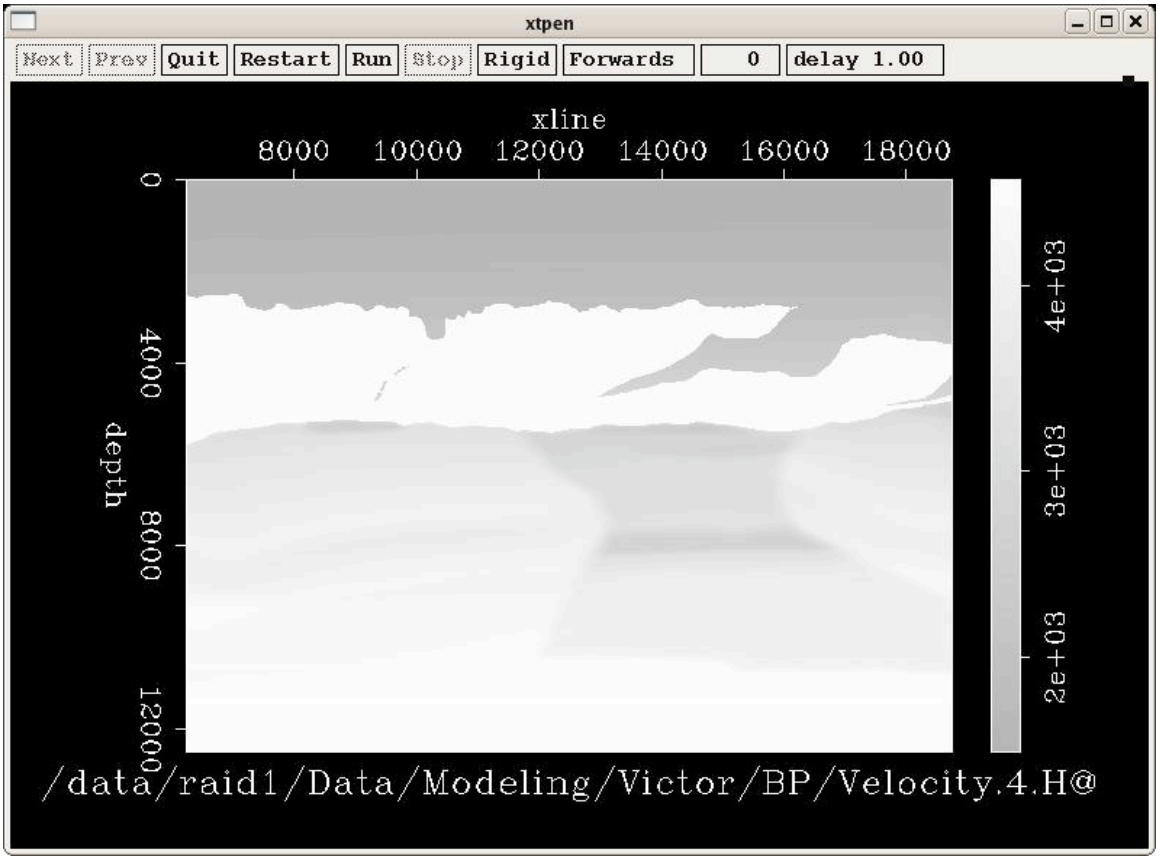
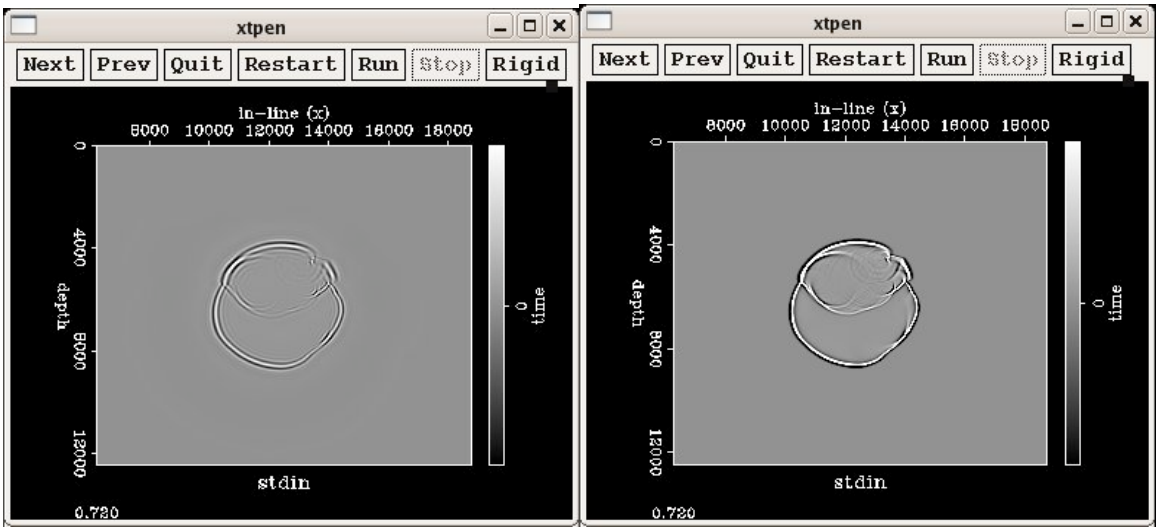


FIGURE 9.2. *Velocity for 2D inhomogeneous model, courtesy of BP.*

the SVD analysis cuts this further to 117 modes. **That is a reduction in the number of degrees of freedom by a factor of 34,222.**



(a) Reduce order solve (b) Full solve

FIGURE 9.3. *Exact and reduced solutions for timestep = 533*

In Figure (9.3) we see the comparison of snapshots for the original and reduced system. In this problem there is no direct comparison of amplitudes since the full fidelity results have been corrected for geometrical spreading while the reduced ones have not. However, all the different phases show in the reduced solution at the correct spatial spots.

10. Solving as a second order system. It would be of interest to solve the problem in its original second order form and also using the same procedure that is used in the high fidelity code, instead of reducing to first order and using a high order external integrator. We do that in this section and show that the results are similar, although the procedure is faster. We use a simple second order approximation to the second derivative centered at the previous point to obtain an explicit integrator:

$$a_i = 2a_{i-1} - a_i + Ra_{i-1} + U_m^T B u(t - dt).$$

The initial conditions are $a_0 = a_1 = 0$.

For the 3 layers problem we see in Fig. a comparison between the exact and the approximate solutions after 500 integration steps. For better comparison we have extracted a 1D section around the middle of the model.

11. Conclusions. We have described a model order reduction method based on projection into a small subspace of time snapshots of the solution of a wave propagation problem. We have shown, in 1 and 2 dimensional problems, from simple to fairly complex media, that the method gives reasonably accurate solutions with a very significant reduction in the number of spatial degrees freedom. The approximations show very low dispersion and some visible dissipation.

For problems in which the kinematics is important, such as in large scale seismic imaging, these approximations should be adequate, although they would require further testing. The impact of the order reduction will be most appreciated in large scale three-dimensional problems that need to be solved repeatedly with small variations in the source position or in material properties.

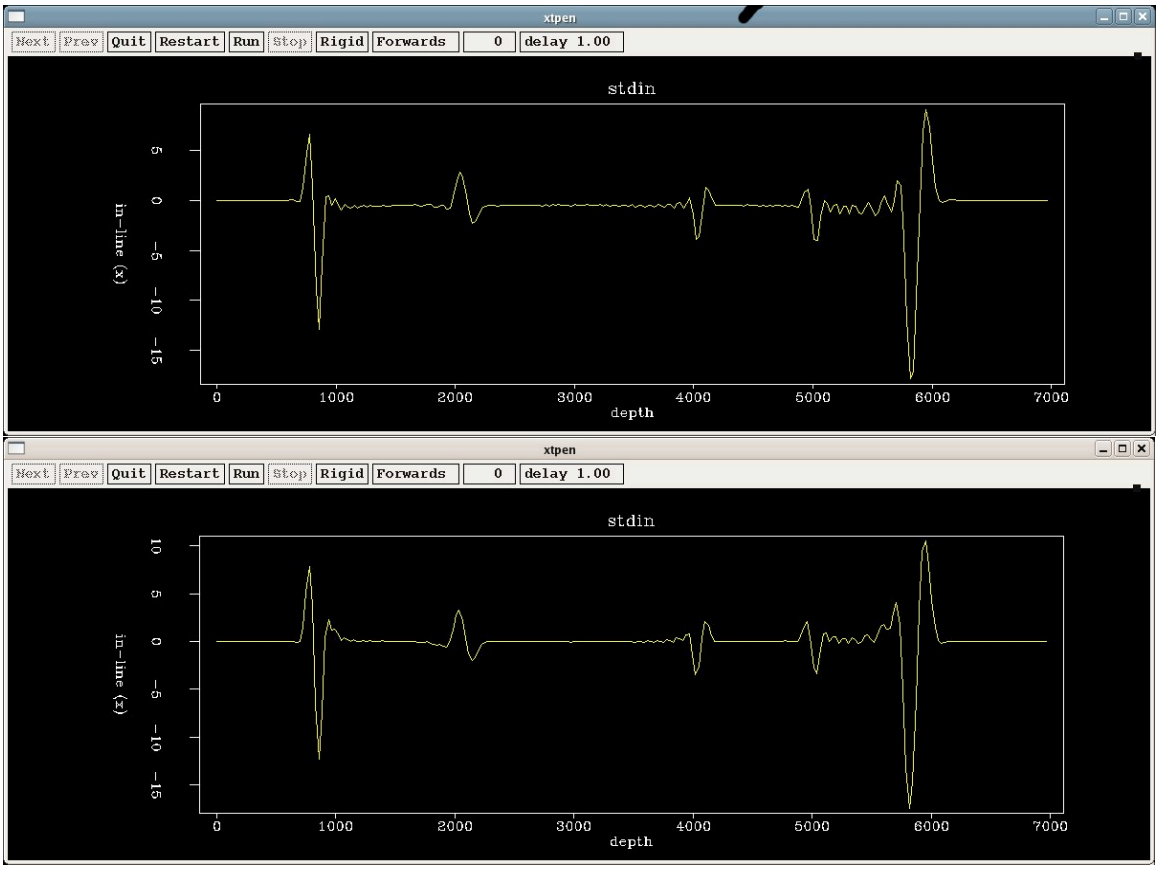


FIGURE 10.1. *MOR and high fidelity solution slice after 500 integration steps.*

REFERENCES

[1] T. J. Klemas, *Full-wave algorithms for model order reduction and electromagnetic analysis of impedance and scattering.* Ph D Thesis, MIT, Electrical Engineering, (2005).

[2] S. Weiland, *Course Model Reduction.* Dep. Electrical Engineering, Eindhoven University of Technology, The Netherlands, (2005).

[3] P. S. Beran and W. A. Silva. *Reduced-order modeling: New approaches for computational physics.* AIAA 2001-0853, NASA Report, (2001).

[4] Peter N. Brown, Alan C. Hindmarsh and George D. Byrne, *VODE: A variable-coefficient ordinary differential equation solver.* SIAM J. Sci. Stat. Comput., 10 (1989) pp. 1038-1051.

[5] R. Freund, *Model reduction methods based on Krilov subspaces.* Acta Numerica, 12 (2003), pp. 267-319.

[6] M. Moonen, P. Van Dooren and J. Vandewalle, *An SVD updating algorithm for subspace tracking.* SIAM J. Matrix Analysis and Appl. 13 (1992**),** pp. 1015-1038.

[7] L. X. Wang, R. V. N. Melnik, *Model reduction applied to square rectangular martensitic transformations using proper orthogonal decomposition.* Applied Numerical Mathematics, 57 (2007) pp. 510-520.

[8] B. Fornberg, *Generation of finite difference formulas on arbitrarily spaced grids.* Math. Comp. 51 (1988**),** pp. 699-706.

[9] H. B. Keller and V. Pereyra, *Symbolic generation of finite difference formulas.* Math. Comp., 32 (1978), pp. 955-971.