Advances in Intelligent Systems Research (AISR), volume 141
International Conference on Applied Mathematics, Modelling and Statistics Application (AMMSA 2017)

Nonlinear Model Reduction for Truss Frame Based on POD-DEIM

Chunchun Cui, Youping Gong*, Jiatian Yu, Jiang Ke, Longbiao Gao and Huipeng Chen
Hangzhou Dianzi University, Hangzhou 310018, Zhejiang, China
*Corresponding author

***Abstract*—Nonlinear model simplification is an important part of many applications, such as flow control, optimization and statistical analysis. In this article, we summarize the POD-DEIM method to solve the nonlinear system of the truss frame .And use an example of a clamped beam to illustrate the DEIM specific simplification process with images which can intuitively observe the image of the specific selection process. The application of the method to the truss frame to verify the correctness of the study. The results show that POD-DEIM is able to approximate fully nonlinear behavior of geometrically nonlinear finite element models with good accuracy.**

Keywords-model order reduction; discrete empirical interpolation method; nonlinear dynamics systems; truss frame

I. INTRODUCTION

The invention of the computer opened a new field of research in physics and engineering. One of the developments is the Finite Element Method .A lot of dofs makes the finite element problem solved expensive. In order to make the finite element analysis (FEA) cheaper, a model order reduction (MOR) technique has been developed. This is done by writing a displacement field with a set of reduced coordinates[2].For linear problems, this can reduce the amount of computations required. However, the number of computations for nonlinear problems can not be reduced in the same way. This paper focuses on a special type of nonlinear problem, geometric nonlinear problem. In general, the internal forces of such systems include linear and non-linear contributions. Although the linear contribution can be reduced by using the reduced coordinates, the evaluation of the nonlinear term needs to know the full order shift field. This creates a bottleneck in the calculation of time. The solution to the bottleneck of the nonlinear term is found in the Discrete Empirical Interpolation (DEIM)[1,3]. In fact, the POD-DEIM reduction is reduced to the approximation of the nonlinear part of the internal force, which requires only a few components (dofs) of the nonlinear internal force vector[5,6].

II. MODEL-ORDER REDUCTION BY THE POD–DEIM METHOD

A. Nonlinear Dynamic Systems

Many scientific modeling and engineering design issues involve partial differential equations (PDEs) that must be solved for time t and spatial variable x. Assume that the problem is written as:

$$\frac{dy(x,t)}{dt} = L(y(x,t)) + F(y(x,t)) \tag{1}$$

Where the linear $L(y(x,t))$ and non-linear $F(y(x,t))$ parts of equation can be written as separate. Ideally, the analytical solution of (1) can be found. Discretising the spatial variable *x* yields:

$$\frac{d}{dt}y(t) = Ay(t) + F(y(t)) \tag{2}$$

The constant matrix A represents the part of the equation that is linear in $\mathbf{y}(t)$.It is assumed that

F(y(t)),a scalar valued nonlinear function in $\mathbf{y}(t)$. These gradients are found by calculating the derivative (Jacobian) of the discretised function with respect to y:

$$\frac{\partial}{\partial y(t)}(Ay(t)) + F(y(t)) = J(y(t)) = A + J_F(Y(t)) \tag{3}$$

where in case of point-to-point functions the matrices A and J*F* are diagonal. Jacobian J*F* then looks like:

$$J_F(y(t)) = diag\left[F_1^{'}(y_1(t)),\ldots,F_n^{'}(y_n(t))\right] \in R^{n\times n} \tag{4}$$

Here, the projection matrix V_k is introduced to project the discrete spatial domain into the subspace $\lambda, k < n$ of lower dimension. Then get a reduced system:

$$\frac{d}{dt}\tilde{y}(t) = \underbrace{V_k^T A V_k}_{\tilde{A}}\tilde{y}(t) + V_k^T F\left(V_k \tilde{y}(t)\right) \tag{5}$$

And corresponding Jacobian:

$$\tilde{J}(y(t)) = \tilde{A} + \underbrace{V_k^T J_F\left(V_k \tilde{y}(t)\right)V_k}_{\tilde{J}_F\left(V_k \tilde{y}(t)\right)} \in R^{k\times k} \tag{6}$$

The use of Galerkin projection does not provide a solution to the problem of having to compute the nonlinear force and its Jacobian. The DEIM algorithm aims to overcome these two bottlenecks.

B. Discrete Empirical Interpolation Method

DEIM is applicable to nonlinear functions as well as to its partial derivatives. The main idea behind this DEIM-approach is to compute the nonlinear term only at m carefully selected locations, with $m << n$, and interpolate elsewhere. The nonlinear equation (2) spans into a subspace $F \in R^{n \times n}$, and the orthogonal basis of the subspace is expressed as:

$$U = [u_1, ..., u_n] \tag{7}$$

The column vector $F\left(V_k \tilde{y}(t)\right)$ can be represented by a sum of these basis vectors:

$$F\left(V_k \tilde{y}(t)\right) = Uc(t) \tag{8}$$

In most algorithms the basis vectors u_i are ordered in such a way that the first basis vector u_i denotes the most meaningful one, and the others follow by order of decreasing importance. So the reduced set of base vectors would be $U_m = [u_1, ..., u_m]$.Suppose that the same assumption holds for the function $F\left(V_k \tilde{y}(t)\right)$:

$$F\left(V_k \tilde{y}(t)\right) = f(t) \approx U_m c(t), \tag{9}$$

Here we use simplified $f(t)$ instead of it .The system (9) needs to be solved for the unknown c_i. This done by introducing a boolean matrix P that selects the m rows that are necessary to make the system invertible:

$$P = [e_{\phi 1}, ...e_{\phi m}] \in R^{n \times m} \tag{10}$$

Using this yet to be determined matrix(9), the basis interpolation of (8) can be made invertible and thus solvable for $c(t)$:

$$P^T f(t) = (P^T U_m) c(t) \Rightarrow c(t) = (P^T U_m)^{-1} P^T f(t) \tag{11}$$

So the $f(t)$ can be described as:

$$f(t) = \underbrace{U_m \left(P^T U_m\right)^{-1} P^T}_{D} f(t), \tag{12}$$

Where D is referred to as the DEIM-matrix. The exact shape of D is still unknown, but due to the selection by P, only m components of the right side f(t) are needed. Our primary interest is how P can be found. This matrix results from the DEIM-algorithm, which will be explained next. DEIM will select for each column in P an index that will be unity.

Algorithm: DEIM

Input: $\{u_i\}_{i=1}^{m} \subset R^n$ linearly independent

Output: $\phi = [\phi_1, ..., \phi_m]^T \in N^m$

1: $[\rho \phi_1] = \max\{|u_1|\}$

2: $U = [u_1], P = [e_{\phi 1}], \phi = [\phi 1]$

3: for $i = 2$ to m do

4: solve $(P^T U)c = P^T u_i$ for c

5: $r = u_i - Uc$

6: $[\rho \phi_i] = \max\{|r|\}$

7: U $\leftarrow [U, u_i], P \leftarrow [P, e_{\phi i}], \phi \leftarrow \begin{bmatrix} \phi \\ \phi_i \end{bmatrix}$

8: end for

The inputs to algorithm are the m basis vectors of the basis U that were found by POD. The output is an array ϕ that contains the index for the unity components in boolean matrix P. Matrix P can thus be constructed from ϕ.

Lines 1-2 are meant to initialize the algorithm. For line 3 on, the iterative procedure is started. On line 4, the interpolation constants in c are solved for. If U_C is subtracted from u_i (line 5), the residual r will always be nonzero. Lines 6 prevents the algorithm from stalling. To conclude the explanation, the remaining steps of line 7 are pointed out. The first operation is the updating of U with the new basis vector u_i, which will be needed for the next iteration. The last operation on line 7 appends the newly found index ϕ_i to the array of index. When the for-loop ends, this array will be returned as an output.

III. USE THE ICON DISPLAY DEIM ALGORITHM HOW TO WORK

The DEIM-algorithm described in the previous section can be visualized with a series of plotted basis vectors u_i, the approximation Uc of u_i, and the residual r. Suppose that a beam is clamped between two vertical walls, and a vertical load is exerted at the middle. The beam is discretized in 10 elements of equal length(figure 1).All segments have the same mass, for each segment equally divided among two nodes. A second simplification is that only shear (vertical) forces are considered. Since both ends are clamped in the walls, a total of 9 Degrees Of freedom remain. It is assumed that these shear forces $f(q) \in R^{9 \times 1}$ are a nonlinear function of the vertical deflections q.

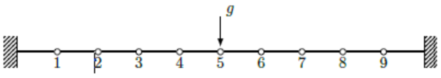


FIGURE I. THE CLAMPED-CLAMPED BEAM

The external force g(t) is applied to the beam at t=0 (step load).The beam will start oscillating, generating time-varying shear forces at each of the nodes. Suppose that for each time increment a snapshot is taken of the shear forces. Hence, if there are N time steps, a $9 \times N$ snapshot matrix Fs can be made. The columns of this snapshot matrix span a subspace $F \subset R^9$ .A basis can be calculated for F by means of a Singular Value Decomposition.

$$F_S = U \Sigma V \tag{13}$$

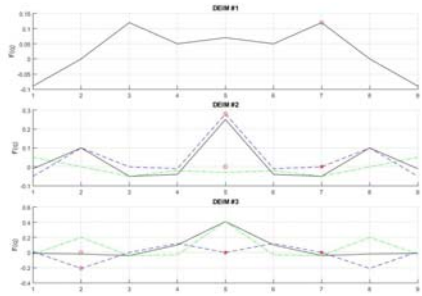
The matrix U contains the basis vectors $u_1,\ldots,u_9$ that belong to the subspace F . Matrix Σ contains a diagonal with singular values, indicating the order of decreasing dominance of basis vectors u_i. Using the force eigenmodes derived from POD, any possible shear force within subspace F can be written as a superposition of these eigenmodes:

$$f(t) = Uc(t) \tag{14}$$

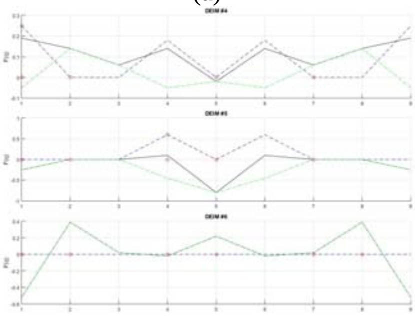
This superposition of force modes can be truncated, considering a smaller number of modes m than the total of 9.However, if $m \le 9$, the matrix $U_m \in R^{9\times m}$ becomes rectangular. This brings back the problem of selecting m rows in u_m described in the previous section. The process of selecting the m rows is visualised by Figure 2 (a) and (b) ,which shows pilots of basis vectors u_i ,interpolations U_c and their $r = u_i - Uc$ for 6 iterations of algorithm DEIM. Step 1 of DEIM initialises algorithm DEIM (lines 1-2).In step 2, $P^T u_2 = P^T Uc$ is solved for c, such that Uc equals u_2 in component 7.In step 3,the interpolation process uses the newly obtained point to make an approximation of u_3 ,now using two interpolation points 5 and 7.Hence a maximum in $|r|$ can always be found, which ensures that the process will never stop before all required m iteration steps are carried out. The remaining steps will obtain more and more interpolation points. After step 5, all linearly independent indices have been selected .This explains why the interpolation exactly fits u_6. The calculations done in step 6 are obsolete, because 5 iterations already represent the full order system.

IV. APPLY THE ALGORITHM TO TRUSS FRAME

One of the more simple types of finite elements is the bar element (sometimes called truss element), of which several descriptions (both linear and nonlinear) exist. The truss frame is shown in Figure 3. The frame consists of 14 nodes, connected by a total of 20 bar elements. The structure is loaded at the top end on the right side (node 7).



(a)



(b)

Legend: ——— $= u_i$ - - - - - - $= Uc$ - - - - - - r

∘ : newly selected DEIM index

* : previously selected DEIM index

FIGURE II. (A)DEIM STEPS 1-3. (B)DEIM STEPS 4-6

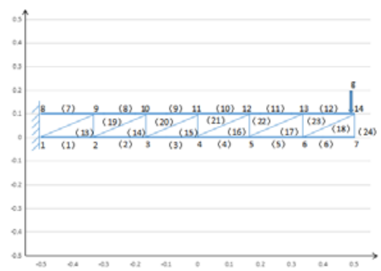
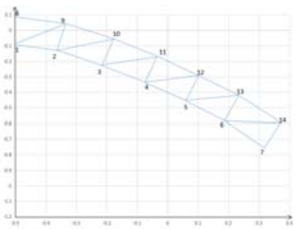
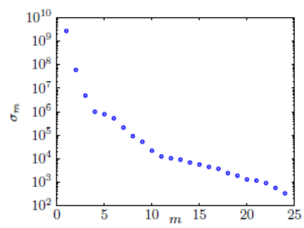


FIGURE III. THE TRUSS FRAME MADE FROM GEOMETRICALLY NONLINEAR BAR ELEMENTS



(a)



(b)

FIGURE IV. (A) DEFORMED SHAPE FOR THE TRUSS FRAME UNDER STATIC LOADING (B) SINGULAR VALUES CORRESPONDING TO THE BASIS

The first step is to do a static loading to verify that the structure is indeed showing nonlinear behavior (Figure 4a).The dynamic response is solved for a ramp load at the tip of the truss at node 14. After 1/3rd of the time steps, the load reaches its maximum amplitude, at which the load level is maintained throughout the remaining time steps. Again, the difference between the linearized and the full order nonlinear system is apparent, (Figure 5).

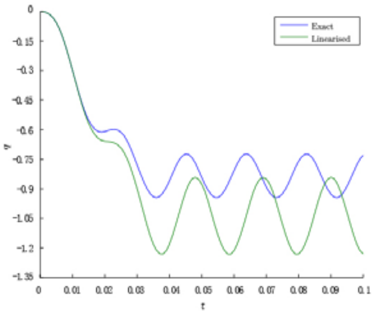


FIGURE V. THE DISPLACEMENT RESPONSE FOR NODE 7 OF THE TRUSS FRAME

Note that the full system of equations is of the size $q, f \in R^{20}$ .This is since there are 14 nodes that each have 2 dofs. Nodes 1 and 8 are constrained. So the remaining $14\times 2 - 2\times 2 = 20$ dofs are free. The snapshots obtained for the nonlinear part of the internal forces yield a matrix of basis vectors $U \in R^{20\times 20}$ .The corresponding singular values show a distribution (Figure 4b) that indicates that much of the nonlinear internal force dynamics is contained in the subspace spanned by the first four basis vectors. The first four singular values have a total sum of 2.8011e^{9}.The sum of all singular values 2.8098e^{9}.So the proportion reached 99.8 % contained in the subspace spanned by the basis vectors $U_4 = [u_1 \ldots u_4]$.This determined the number of dofs to be selected for testing the POD-DEIM reduction. The POD-DEIM reduction has been tested with 1, 2, 3 and 4 dofs. The results for the displacements are given in Figure 6.As can be seen, the responses for 2, 3 and 4 dofs are able to approximate the full 20-dof system with reasonable accuracy during the first 1/3rd of the total time. The 4-dof system is able to track the original response. The results obtained for the geometrically nonlinear truss frame show that the application of DEIM to a POD-reduced system can provide good results. Through the calculation can be obtained when in the steady state the error shown in Table 1.

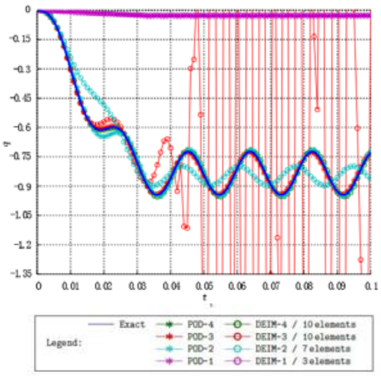


FIGURE VI. RESULTS OBTAINED BY DIRECT APPLICATION OF DEIM ON THE EQUATIONS OF MOTION OF THE TRUSS FRAME

TABLE I. ERROR ANALYSIS

m	1	2	3	4
data	0.023	0.795	0.734	0.740
error	96.8%	0.729%	1.0%	0.05%

For m = 4 dofs, the POD-DEIM response could accurately track both POD-reduced and the unreduced system. This POD-DEIM approximation required only information from 4 out of 20 components from the nonlinear internal force vector $f \in R^{20\times 1}$, together with $4\times 4 = 16$ corresponding components of the Jacobian matrix. Without the additional reduction of DEIM, the POD-reduced system would theoretically have required 20×20 = 400 components2 to be known from the Jacobian. The fact that the responses are accurate and stable while using a much smaller number of components implies that the computational cost can be reduced. By calculating the time of the calculation (Table 2).Through the chart can be seen after the reduction of the model calculation time has been greatly improved

TABLE II. TIME ANALYSIS

	Unreduced model	Reduced model
time	151s	43s

As a validation case, we chose the truss frame which systems include linear and non-linear contributions. The reduced-order model contains only a few degrees of freedom, but still expresses the characteristics of the original system in a very accurate way .And the calculation time has been saved.

V. CONCLUSIONS

This paper introduces the POD-DEIM algorithm and graphically shows the operation of the DEIM algorithm, and through the reduction of the truss model, which reduce the nonlinear contributions in a truss frame from 20 elements back to just 4 elements. It has been shown that DEIM is able to approximate fully nonlinear behavior of geometrically nonlinear finite element models with good accuracy.

ACKNOWLEDGMENT

The project was supported by Zhejiang province higher Visiting Scholar project “Model order reduction technology and used in MEMS, FX2014080”. And the project was supported by “Research on multi-disciplinary Integrated design Optimization Method Based on MBSE, LY15E050023” .Also the project was supported by “Research and Development of Automatic Membrane Filtration Intelligent Agglomeration Operation Module, 2016YFF0103203”.

REFERENCES

[1] S. Chaturantabut, D.C. Sorensen, Nonlinear model reduction via discrete empirical interpolation, SIAM J. Sci. Comput. 32(5) (2010) 2737–2764.

[2] de Pando M F, Schmid P J, Sipp D. Nonlinear model-order reduction for compressible flow solvers using the Discrete Empirical Interpolation Method[J]. Journal of Computational Physics, 2016, 324:194-209

[3] Stanko Z P, Boyce S E, Yeh W. Nonlinear model reduction of unconfined groundwater flow using POD and DEIM[J]. Advances in Water Resources, 2016, 97:130-143.

[4] S. Chaturantabut, D.C. Sorensen, Application of POD and DEIM on dimension reduction of non-linear miscible viscous fingering in porous media, Math. Comput. Model. Dyn. Syst. 17(4) (2011) 337–353

[5] D.A. Knoll, D.E. Keyes, Jacobian-free Newton–Krylov methods: a survey of approaches and applications, J. Comput. Phys. 193(2) (2004) 357–397.

[6] Antil H, Heinkenschloss M, Sorensen D C. Application of the Discrete Empirical Interpolation Method to Reduced Order Modeling of Nonlinear and Parametric Systems[Book Section]. Book Section]. Ed.Eds., 2014,Vol. 9(pp 101-136.

[7] Stefanescu R, Sandu A, Navon I M. Comparison of POD reduced order strategies for the nonlinear 2D shallow water equations[J]. International Journal for Numerical Methods in Fluids, 2014, 76(8):497-521.

[8] Sargsyan S, Brunton S L, Kutz J N. Nonlinear model reduction for dynamical systems using sparse sensor locations from learned libraries[J]. Physical Review E, 2015, 92(0333043).

[9] Xiao D, Fang F, Buchan A G,et al. Non-linear model reduction for the Navier-Stokes equations using residual DEIM method[J]. Journal of Computational Physics, 2014, 263:1-18.