EVALUATION OF PARTIAL FACTORIZATION FOR CONDENSATION OF SHELL AND SOLID-SHELL ELEMENTAL MATRICES

***P. Jarzebski*[1] *and K. Wisniewski*[1]**
[1]*Institute of Fundamental Technological Research, Warsaw, Poland*

1. Introduction

Mixed/enhanced shell finite elements involve additional local parameters which must be condensed out to reduce the size of a tangent elemental matrix to the standard one.

The number of elemental parameters can be quite large, sometimes exceeding the number of nodal variables associated with the element, and, typically, this condensation is performed by elimination of particular types of parameters one after the other by inversion of respective submatrices. This approach must be adjusted to each new formulation of a finite element, which is a clear drawback.

An alternative approach is to condense out all parameters together, which involves computation of the Schur complement. To speed up this process, a *Partial Factorization* (PF) technique is tested as a replacement for the standard Schur complement computations. In this paper, we test and compare its performance for a range of shell, solid-shell and 3D elements.

2. Finite elements with additional parameters

For the considered class of finite elements, the governing functional F depends on the nodal displacements and rotational parameters $\mathbf{u}_I$ and the elemental multipliers $\mathbf{q}$. For kinematically non-linear problems, the stationarity condition of $F(\mathbf{u}_I, \mathbf{q})$ yields a system of equilibrium equations for an element, $\mathbf{r}_u(\mathbf{u}_I, \mathbf{q}) = \mathbf{0}$ and $\mathbf{r}_q(\mathbf{u}_I, \mathbf{q}) = \mathbf{0}$. The linearized (Newton) form of these equations is as follows:

$$(1) \qquad \begin{bmatrix} \mathbf{K} & \mathbf{L} \\ \mathbf{L}^T & \mathbf{K}_{qq} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{u}_I \\ \Delta\mathbf{q} \end{bmatrix} = - \begin{bmatrix} \mathbf{r}_u \\ \mathbf{r}_q \end{bmatrix},$$

where $\mathbf{K} \doteq \partial\mathbf{r}_u/\partial\mathbf{u}_I$, $\mathbf{L} \doteq \partial\mathbf{r}_u/\partial\mathbf{q}$ and $\mathbf{K}_{qq} \doteq \partial\mathbf{r}_q/\partial\mathbf{q}$. Note that $\mathbf{K}$ and $\mathbf{K}_{qq}$ are symmetric and, in general, indefinite.

To eliminate the multipliers at the element level and to reduce the size of a tangent matrix to the standard one, defined by the number of nodes on the element and dofs/node, we calculate $\Delta\mathbf{q}$ from the second equation, and we use it in the first equation, which yields

$$(2) \qquad \mathbf{K}^*\Delta\mathbf{u}_I = -\mathbf{r}^*, \qquad \text{where} \qquad \mathbf{K}^* = \mathbf{K} - \mathbf{L}\mathbf{K}_{qq}^{-1}\mathbf{L}^T \quad \text{and} \quad \mathbf{r}^* = \mathbf{r}_u - \mathbf{L}\mathbf{K}_{qq}^{-1}\mathbf{r}_q.$$

We see that the reduced (or condensed) matrix $\mathbf{K}^*$ is defined as the Schur complement of $\mathbf{K}$, and typically is calculated in several steps: (1) triangularization of $\mathbf{K}_{qq}$, (2) one back-substitution for each column of $\mathbf{L}^T$ to obtain $\mathbf{K}_{qq}^{-1}\mathbf{L}^T$, and (3) multiplication of $\mathbf{L}$ by $\mathbf{K}_{qq}^{-1}\mathbf{L}^T$ and substraction of the product from $\mathbf{K}$. To speed up the above condensation, we implemented and tested an alternative method described below.

3. Partial Factorization (PF)

To define the PF method, let us first interchange the order of $\mathbf{u}_I$ and $\mathbf{q}$ so instead of the matrix of eq. (1), we have the matrix $\mathbf{A}$ defined below. The LU decomposition of $\mathbf{A}$ is

$$(3) \qquad \mathbf{A} \doteq \begin{bmatrix} \mathbf{K}_{qq} & \mathbf{L}^T \\ \mathbf{L} & \mathbf{K} \end{bmatrix} = \begin{bmatrix} \mathbf{L}_{11} & 0 \\ \mathbf{L}_{21} & \mathbf{L}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{U}_{11} & \mathbf{U}_{12} \\ 0 & \mathbf{U}_{22} \end{bmatrix} = \begin{bmatrix} \mathbf{L}_{11}\mathbf{U}_{11} & \mathbf{L}_{11}\mathbf{U}_{12} \\ \mathbf{L}_{21}\mathbf{U}_{11} & \mathbf{L}_{21}\mathbf{U}_{12} + \mathbf{L}_{22}\mathbf{U}_{22} \end{bmatrix}.$$

Noting that $\mathbf{L}_{21}\mathbf{U}_{11} = \mathbf{L}$ and $\mathbf{L}_{11}\mathbf{U}_{12} = \mathbf{L}^T$, we obtain $\mathbf{L}_{21}\mathbf{U}_{12} = \mathbf{L}\mathbf{U}_{11}^{-1}\mathbf{L}_{11}^{-1}\mathbf{L}^T = \mathbf{L}\mathbf{K}_{qq}^{-1}\mathbf{L}^T$. Hence,

$$(4) \quad \mathbf{K} - \mathbf{L}_{21}\mathbf{U}_{12} = \mathbf{K} - \mathbf{L}\mathbf{K}_{qq}^{-1}\mathbf{L}^T = \mathbf{K}^*,$$

where $\mathbf{K}^*$ is the reduced matrix of eq. (2). We do not compute the bottom-right block factors $\mathbf{L}_{22}$ and $\mathbf{U}_{22}$ to obtain this matrix, which explains the term ”partial” factorization.

4. Numerical results

Tests were performed using only 1 core of a multi-core machine (2 processors Xeon X5650 2.66GHz with 6 cores each, running under Linux). This is in accord with the parallelization of a loop over elements which we implemented in [1], by which each core processes a different finite element.

The stiffness matrices used in computations were obtained for the central elements of patch tests of [MacNeal, Harder, 1985]. The computations were repeated 1 million times for each matrix. The speedups are presented in Table 1, where the best results for each matrix are boldfaced.

		Shell elements			Solid-shell elements			3D elements		
Solver	Method	EAS10	HW35	HW43	EAS10	HW29	HW47	EADG12	EAS30	HW60
DSYTRF	1RHS	0.26	0.48	0.68	0.30	0.56	0.60	0.29	0.45	0.75
MA64	1RHS	0.57	1.02	1.36	0.72	1.37	1.51	0.84	1.23	1.74
DSYTRF	mRHS	0.97	1.72	2.06	1.04	1.93	1.82	1.12	1.49	2.20
MA64	mRHS	0.62	1.04	1.37	0.77	1.39	1.82	0.90	1.26	1.75
DSYTRF	ownPF	**1.27**	**2.47**	**3.37**	**1.37**	**2.64**	**2.52**	**1.57**	**1.86**	**2.41**
MA64	PF	0.58	1.41	1.91	0.72	1.90	2.10	0.85	1.49	2.31
Matrix density [%]		91.20	100.00	58.70	98.40	48.20	29.20	40.60	22.20	13.70
Reference time [secs]		6.77	49.96	129.08	10.46	75.19	108.19	13.34	53.88	192.54

Table 1. Speedup for particular methods and solvers related to the time for scheme of eq. (3) and solver of [3].

Solver and method. For reference, we used the scheme of eq. (2) and the LUDCMP routine of [3]. Besides, we tested two routines based on the Gauss elimination: DSYTRF of [LAPACK ver. 3.2] and MA64 of [HSL, 2013]. To obtain $\mathbf{K}_{qq}^{-1}\mathbf{L}^T$, the back-substitution routine is called in two ways: either for each column of $\mathbf{L}^T$ separately (“1RHS”) or for all columns of $\mathbf{L}^T$ together (“mRHS”). Besides, “ownPF” indicates our modification of the code to perform the PF.

Finite elements. Three types of elements were tested: 4-node shells, 8-node solid-shells and 8-node 3D elements. Their formulations are designated by: HW - based on the Hu-Washizu functional and enhanced for shell and solid-shell elements [2], EAS - based on the potential energy with the Enhanced Assumed Strain, and EADG - based on the potential energy with Enhanced Assumed Displacement Gradient. The number of additional parameters follows these letters.

Concluding, the tests show that the implementation of the *Partial Factorization* (PF) is beneficial for almost all elements and the method using “DSYTRF” and “ownPF” provides the best speedup.

5. References

[1] P. Jarzebski, K. Wisniewski, R.L. Taylor (2015). On paralelization of the loop over elements in FEAP, *Computational Mechanics*, **56(1)**, 77-86.

[2] K. Wisniewski, E. Turska (2012). Four-node mixed Hu-Washizu shell element with drilling rotation. *Int. J. Num. Meth. Engng*, **90**, 506-536.

[3] W. H. Press et al. (1999). *Numerical Recipes in Fortran 77*, Cambridge Univeristy Press.