Karlsruhe Institute of Technology

Institute of Mechanics

Highly Efficient Implementation of ANS/ EAS ‘Solid-Shell’ Finite Elements for Explicit Time Integration

December 2010

Steffen Mattern, Karl Schweizerhof

Institute of Mechanics
Kaiserstr. 12
D-76131 Karlsruhe
Tel.: +49 (0) 721/ 608-2071
Fax: +49 (0) 721/608-7990
E-Mail: info@ifm.kit.edu
www.ifm.uni-karlsruhe.de

Highly Efficient Implementation of ANS/ EAS 'Solid-Shell' Finite Elements for Explicit Time Integration

Steffen Mattern Karl Schweizerhof

Abstract

In an explicit time integration algorithm, the most time-consuming part of an analysis is the computation of the internal nodal force vector. An implementation concept for element routines for volumetric shell – so-called Solid-Shell – elements based on the application of the symbolic programming tool AceGen, a plug-in for the computer algebra software Mathematica is presented. Symbolic implementation means that vector and matrix operations and differentiations can be performed symbolically in order to generate program code, which supports the implementation, reduces programming effort and helps to avoid programming errors almost completely. The program code is generated in Fortran and simultaneously optimized automatically, which leads to very efficient routines compared to manually implemented code.

1 Explicit Time Integration

Explicit time integration is commonly used in finite element analysis and perfectly suited to highly dynamic applications, e.g. crash or impact. The efficiency of the time integration scheme on global level is based on the application of diagonalized mass matrices; as a consequence the computation of the accelerations at the current time step

$$\mathbf{a}^n = \mathbf{M}^{-1} \left(\mathbf{f}^{ext,n} - \mathbf{f}^{int,n} \right) \tag{1}$$

involves only vector operations. However, the time step size is limited to a critical value, consequently transient analyses require a high number of time steps. This is where efficiency in the handling of the operations for force calculations in each step plays the dominant role. In every time step, the internal forces $\mathbf{f}^{int,n}$ have to be integrated over all elements, thus a dominant part of the required CPU-time is spent on element level.

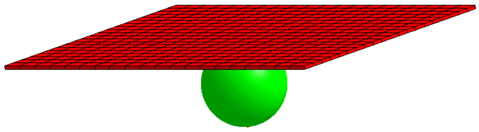
2 Solid-Shell Elements

The concept of Solid-Shell Finite Elements as presented e. g. in [1] provides a shell formulation with full 3D capabilities and displacement degrees of freedom only. In this contribution, isoparametric curved Solid-Shell elements are used with bi-linear/ bi-quadratic interpolation in membrane and linear interpolation in thickness direction. For the discretization of the initial geometry, this leads to

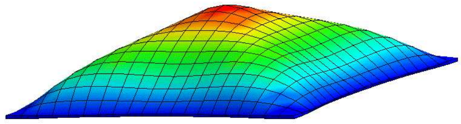
$$\mathbf{X}^{el}(\xi, \eta, \zeta) = \sum_{i=1}^{nip} \left(\frac{1}{2} N_i(\xi, \eta) \, \boldsymbol{\Theta}(\zeta) \, \mathbf{X}_i \right), \tag{2}$$

where nip is the number of in-plane nodes. The upper and lower nodal locations are described by the vector $\mathbf{X}_i = \begin{bmatrix} \mathbf{X}_{iu} & \mathbf{X}_{il} \end{bmatrix}^T$, the interpolation is performed linearly in thickness direction with the interpolation matrix $\boldsymbol{\Theta}(\zeta)$. The in-plane interpolation is achieved in the present case with linear ($nip = 4$) or quadratic ($nip = 9$) Lagrangian shape functions. According to the isoparametric concept, the displacements are interpolated with the same shape functions.

As is commonly known, pure displacement element formulations lead to artificial stiffness effects – the so-called locking phenomena. In order to reduce locking, recent contributions (see e. g. [2, 3]) suggest reduced integration rules in combination with stabilization techniques for Solid-Shell elements. The formulations presented in this contribution are programmed with standard (full) integration rules and locking phenomena are treated using the methods of '*Assumed Natural Strains*' (ANS) [4, 5] and '*Enhanced Assumed Strains*' (EAS) [6, 7]. The ANS method is based on the evaluation of strains at special assembling points together with an interpolation in order to treat geometrical locking effects such



time = 0.00 s



time = 0.15 s

Solid-Shell subroutine	**contact subroutine**	**CPU-time**	**relative**
manually	manually	3202 s	100.00 %
manually	AceGen	3103 s	96.91 %
AceGen	manually	309 s	9.65 %
AceGen	AceGen	207 s	6.46 %

Figure 1: impact of a thin elastic plate on a rigid sphere – CPU-times for different element routines

as shear or membrane locking. The compatible strain field, which also shows a material locking effect, controlled by the Poisson-ratio is then enhanced by introducing additional degrees of freedom – the so-called EAS parameters – which are condensed out by local equation solving.

3 Efficient Implementation

An efficient implementation of the element routines especially the mentioned computation of the residual force vector is achieved in the current project, using the specific programming tool AceGen [8], a plug-in for the computer algebra software Mathematica. The advantage of such tools is – after a first glance – the straight forward and extremely fast generation of element program variations due to the use of e.g. symbolic differentiation of equations and the error-free code development at the same time. As the generated code is automatically optimized, a very efficient implementation can be achieved. Comparison of manually programmed and automatically optimized element code within the in-house FE program lead to a reduction of the necessary CPU-time of up to 90 % for several numerical examples.

4 Numerical Example

The impact of a thin elastic plate on a rigid sphere is simulated using fully integrated linear Solid-Shells in order to show the speed-up, achieved by the application of symbolic programming. The element routines as well as a 'Mortar'-type penalty contact formulation have been implemented both manually and with AceGen. Figure 1 shows one state of the dynamic simulation and a table containing the numerical effort for the different simulations of a $20 \times 20 \times 1$ element mesh which requires approximately 50 000 time steps. The high efficiency of the implemented routines can be reached without any manual improvement of the program code, as AceGen simultaneously optimizes the code regarding the operations. It has to be noted that – as expected – the incorporation of the EAS-method slows down the performance considerably, thus only very few parameters should be taken. Further investigations on this example, which cannot be discussed here in detail, show the advantage of the quadratic geometry interpolations, especially at large deformations when the geometry has to be correctly captured e. g. for stability problems.

5 Conclusions and Outlook

The implementation concept using symbolic programming with AceGen is perfectly suited for element development in the context of explicit time integration as it allows simplified and error-free programming and provides highly efficient program routines. Further element formulations – especially standard shell elements with linear and quadratic geometry and displacement interpolations – are currently implemented. Also a very promising application is the implementation of complex material models, as here the advantage of automatic differentiation is particularly interesting.

References

[1] R. Hauptmann and K. Schweizerhof, A systematic development of 'solid-shell' element formulations for linear and non-linear analyses employing only displacement degrees of freedom, Int. J. Numer. Methods Eng., **42(1)**, 49–69 (1998).

[2] R. P. R. Cardoso, J. W. Yoon, M. Mahardika, S. Choudhry, R. J. Alves de Sousa and R. A. Fontes Valente, Enhanced assumed strain (EAS) and assumed natural strain (ANS) methods for one-point quadrature solid-shell elements, Int. J. Numer. Methods Eng., **75(2)**, 156–187 (2008).

[3] M. Schwarze and S. Reese, A reduced integration solid-shell finite element based on the EAS and the ANS concept – Geometrically linear problems, Int. J. Numer. Methods Eng., **80(10)**, 1322–1355 (2009)

[4] K.-J. Bathe and E. N. Dvorkin, A formulation of general shell elements - the use of mixed interpolation of tensorial components, Int. J. Numer. Methods Eng., **22(3)**, 697–722 (1986).

[5] P. Betsch and E. Stein, An assumed strain approach avoiding artificial thickness straining for a non-linear 4-node shell element, Commun. Numer. Methods Eng., **11**, 899–909 (1995).

[6] J. C. Simo and M. S. Rifai, A class of mixed assumed strain methods and the method of incompatible modes, Int. J. Numer. Methods Eng., **29(8)**, 1595–1638 (1990).

[7] J. C. Simo and F. Armero, Geometrically non-linear enhanced strain mixed methods and the method of incompatible modes, Int. J. Numer. Methods Eng., **33**, 1413–1449 (1992).

[8] J. Korelc, `http://www.fgg.uni-lj.si/Symech/` (2010)