Derivation of element stiffness matrices using computer algebra

D. V. Griffiths
Department of Engineering
University of Manchester
Manchester M13 9PL, U.K.

Abstract

A computer algebra package called Maple is used to generate exact expressions for the stiffness matrix of a plane 4-node quadrilateral element. The expressions are conveniently stored in a subroutine library and run significantly faster than 'traditional' approaches using Gaussian quadrature.

1 Introduction

The 'raw materials' for generation of an elastic element stiffness matrix in two-dimensions are:

- The element type (4-node, 6-node, etc.).
- The strain conditions (plane strain, plane stress, etc.).
- The elastic properties (Young's Modulus, Poisson's Ratio).
- The nodal coordinates.

This paper uses the 4-node quadrilateral element in plane strain to illustrate the use of a computer algebra system to generate exact expressions for the stiffness matrix. Analytical expressions for the stiffness matrix of a *rectangular* four-node element have been published by Hacker and Schreyer [1]. It is well known that this element's stiffness matrix is exactly integrated using four Gauss-points per element. This paper describes how the stiffness matrix of a general quadrilateral element can be expressed in closed form by expanding and simplifying the four terms in the numerical integration summation. The computer algebra system Maple [2] was used to help generate the expressions.

Computer algebra systems (CAS) have considerable potential in the area of finite element software generation. In particular, Bettess and Bettess [3] and Barbier *et al* [4] showed how the computer algebra system REDUCE [5] could be used to automatically generate shape functions for any finite element. The systems usually have the additional facility of being able to generate output in the FORTRAN programming language; thus complex algebraic expressions can be coded without the usual risk of typographical errors.

The ability of CAS to simplify and factorise complex algebraic terms has limitations however, so some of the expressions produced by the CAS had to be further simplified by hand in order to arrive at a form suitable for publication.

2 Formulation

The element stiffness matrix **k** can be written as an integral (see e.g. Zienkiewicz [6]) of the form:

$$\mathbf{k} = \int_{V^e} \mathbf{B}^{\mathrm{T}}\mathbf{D}\mathbf{B}\,\mathrm{d(vol)} \tag{1}$$

where the stress/strain **D** matrix is:

$$\mathbf{D} = \begin{bmatrix} E_1 & E_2 & 0 \\ E_2 & E_1 & 0 \\ 0 & 0 & G \end{bmatrix} \tag{2}$$

where the shear modulus is given by $G = E/(2(1+\nu))$, with E and ν denoting Young's modulus and Poisson's ratio respectively.

For plane strain:

$$E_1 = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)} \quad , \quad E_2 = \frac{\nu E_1}{(1-\nu)} \tag{3}$$

B is the strain/displacement matrix.

The terms in the matrix **B** are derivatives of the shape functions with respect to *global* coordinates, so the usual tranformations must be performed at each Gauss point. These transformations and the notation used throughout this paper are consistent with those used in the text by Smith and Griffiths [7].

2.1 A typical Maple program

This section gives a listing of a typical Maple program for computing the contribution of a Gauss point to a particular term in the element stiffness matrix **k**. The term in **k** is defined by the integers `IROW` and `ICOL`, and the Gauss point by the integers `II` and `JJ`. The listing below would compute the contribution of the Gauss point given by (`II = 1, JJ = 1`) to the term k_{11}.

```
DEE:= array(1..3,1..3):SAMP:=array(1..2,1..2):
BTDB:= array(1..8,1..8):
DERIV:=array(1..2,1..4):DER:= array(1..2,1..4):
COORD:=array(1..4,1..2):JAC:= array(1..2,1..2):
JAC1:= array(1..2,1..2):BEE:= array(sparse,1..3,1..8):
with(linalg):
readlib(evalm):
#
# Plane Strain
E1:=YM*(1-V)/(1+V)/(1-2*V):
E2:= V*E1/(1-V):
G:=YM/2/(1+V):
#
DEE[1,1]:=E1:  DEE[1,2]:=E2:  DEE[1,3]:=0:
DEE[2,1]:=E2:  DEE[2,2]:=E1:  DEE[2,3]:=0:
DEE[3,1]:=0:  DEE[3,2]:=0:  DEE[3,3]:=G:
#
COORD[1,1]:=X1:  COORD[1,2]:=Y1:
COORD[2,1]:=X2:  COORD[2,2]:=Y2:
COORD[3,1]:=X3:  COORD[3,2]:=Y3:
COORD[4,1]:=X4:  COORD[4,2]:=Y4:
#
# Assuming two Gauss points (NGP=2)
SAMP[1,1]:=1/3**(1/2):  SAMP[1,2]:=1:
SAMP[2,1]:=-SAMP[1,1]:  SAMP[2,2]:=1:

# Enter Row and Column of stiffness matrix term
IROW:=1:ICOL:=1:
#
# Enter Gauss point term
II:=2:JJ:=2:
#
ETA:=SAMP[II,1]:  XI:=SAMP[JJ,1]:
ETAM:=1/4*(1-ETA): ETAP:=1/4*(1+ETA):
XIM :=1/4*(1-XI): XIP :=1/4*(1+XI):
DER[1,1]:=-ETAM: DER[1,2]:=-ETAP:
DER[1,3]:=ETAP: DER[1,4]:=ETAM:
DER[2,1]:=-XIM : DER[2,2]:= XIM:
DER[2,3]:=XIP : DER[2,4]:=-XIP:
#
JAC:=multiply(DER,COORD):
JAC1:=inverse(JAC):
DET:=det(JAC):
DERIV:=multiply(JAC1,DER):
#
```

```
# Form the B matrix
for M from 1 to 4 do
K:=2*M:
L:=K-1:
X:=DERIV[1,M]:
BEE[1,L]:=X:
BEE[3,K]:=X:
Y:=DERIV[2,M]:
BEE[2,K]:=Y:
BEE[3,L]:=Y:
od:
#
GP:= dotprod(col(BEE,IROW),col(DEE,1))*BEE[1,ICOL]:
GP:=GP+dotprod(col(BEE,IROW),col(DEE,2))*BEE[2,ICOL]:
GP:=GP+dotprod(col(BEE,IROW),col(DEE,3))*BEE[3,ICOL]:
TERM:=GP*DET*SAMP[II,2]*SAMP[JJ,2];
#
subs(YM=100,V=1/4,
X1=0,X2=1/4,X3=4/10,X4=7/10,
Y1=0,Y2=75/100,Y3=85/100,Y4=5/100,"):
evalf(",7);
#
quit;
```

Substitution of the following nodal coordinates:

Node	(x,y)
1	(0.00, 0.00)
2	(0.25, 0.75)
3	(0.40, 0.85)
4	(0.70, 0.05)

together with elastic properties $E = 100, \nu = 0.25$ (plane strain), the following contribution to k_{11} from each Gauss-point is obtained:

Gauss point coordinate	Contribution
(0.4071, 0.6620)	3.296
(0.2534, 0.6104)	3.014
(0.5132, 0.2063)	25.162
(0.1762, 0.1713)	23.866
Total	55.338

2.2 Analytical Expressions

So as not to take up too much space, the expressions for k_{11} only will be given in this paper. Very similar expressions however are obtained for all terms of the stiffness matrix. Taking into account both symmetry of the matrix and the relationship between terms that can be obtained by simple rotation of nodal coordinates, it can be shown that the six independent stiffness terms given by k_{11}, k_{21}, k_{31}, k_{41}, k_{51} and k_{61} are sufficient to generate the other 58 terms of the element matrix.

All stiffness terms are of the form:

$$k_{ij} = \frac{1}{2}\left\{\frac{A_2(E^*s_1 + Gs_2) + f_1(E^*s_3 + Gs_4)}{3A_2^2 - f_1^2} + \frac{A_2(E^*t_1 + Gt_2) + f_2(E^*t_3 + Gt_4)}{3A_2^2 - f_2^2}\right\} \tag{4}$$

where E^* equals either E_1 or E_2 as indicated.

$$\begin{aligned} A_2 &= (x_4 - x_2)(y_3 - y_1) - (x_3 - x_1)(y_4 - y_2) \\ &= \text{twice the area of the element} \end{aligned} \tag{5}$$

and

$$f_1 = (x_1 + x_3)(y_4 - y_2) - (y_1 + y_3)(x_4 - x_2) - 2(x_2y_4 - x_4y_2) \tag{6}$$
$$f_2 = (y_2 + y_4)(x_3 - x_1) - (x_2 + x_4)(y_3 - y_1) - 2(x_3y_1 - x_1y_3) \tag{7}$$

The functions $s_1, s_2, s_3, s_4, t_1, t_2, t_3$ and t_4 depend on the nodal coordinates.

2.3 Term - k_{11}

$$E^* = E_1 \tag{8}$$

$$s_1 = 2(y_4 - y_2)^2 \tag{9}$$
$$s_2 = 2(x_4 - x_2)^2 \tag{10}$$
$$s_3 = -s_1/2 \tag{11}$$
$$s_4 = -s_2/2 \tag{12}$$
$$t_1 = (y_2 - y_3)^2 + (y_3 - y_4)^2 + (y_4 - y_2)^2 \tag{13}$$
$$t_2 = (x_2 - x_3)^2 + (x_3 - x_4)^2 + (x_4 - x_2)^2 \tag{14}$$
$$t_3 = (y_4 - y_3)^2 - (y_3 - y_2)^2 \tag{15}$$
$$t_4 = (x_4 - x_3)^2 - (x_3 - x_2)^2 \tag{16}$$

3 Concluding Remarks

The stiffness matrix of a plane four-node quadrilateral finite element can be expressed in closed form with the help of a computer algebra package such as Maple. A typical term has been presented in this paper.

Although the expressions are quite long, they are comprised entirely of assignment statements which when coded in a FORTRAN program will run faster than the conventional numerical formulation. Initial indications are that the 'analytical' approach gives a four-fold reduction in CPU-time on a scalar machine.

Although the exactly integrated four-node element is not widely used, the techniques used to generate the stiffness terms could easily be extrapolated to other types of element matrix (e.g. mass), and higher order elements. Work is presently under way to obtain expressions for matrices of more 'popular' elements, such as the four-node element with selective reduced integration and the eight-node element with uniform reduced integration.

References

[1] W.L Hacker and H.L. Schreyer. Eigenvalue analysis of compatible and incompatible rectangular four-node quadrilateral elements. *Int J Numer Methods Eng*, 28(3):687–703, 1989.

[2] D. Harper. *An introduction to Maple.* Computer Algebra Support Project, University of Liverpool, 1989. Maple is available from WATCOM Publications Ltd., Waterloo, Ontario, Canada N2L 3X2.

[3] P. Bettess and J.A. Bettess. Automatic generation of shape function routines. In G.N. Pande and J. Middleton, editors, *Proc. Int. Conf. Num. Meths. in Eng.:Theory and Applications*, volume 1. Martinus Nuhoff Publishers, 1987.

[4] C. Barbier, P.J. Clark, P. Bettess, and J.A. Bettess. Automatic generation of shape functions for finite element analysis using REDUCE. *Eng Comput (Swansea, Wales)*, 7(4):349–358, 1990.

[5] G. Rayna. *REDUCE - Software for Algebraic Computation.* Springer-Verlag, 1987.

[6] O.C. Zienkiewicz and R.L. Taylor. *The finite element method*, volume 1. McGraw Hill, London, New York, 4th edition, 1989.

[7] I.M. Smith and D.V. Griffiths. *Programming the Finite Element Method.* John Wiley and Sons, Chichester, New York, 2nd edition, 1988.