

Use of computer algebra systems in finite element software development

D.V. Griffiths

Professor, Geomechanics Research Center, Colorado School of Mines, Golden, CO 80401, USA

Abstract

This paper explains how computer algebra systems can be a valuable aid in the development of finite element software. Several examples used recently in the subroutine library of an extensive suite of finite element programs and an associated textbook are described.

Introduction

“The major purpose of a Computer Algebra System (CAS) is to manipulate a formula symbolically using the computer. For example, expanding, factorizing, root finding, or simplifying an algebraic polynomial are some of the common uses of CAS. However, many systems listed here have gone far beyond that and also offer other functionalities like numerical calculation, graphics, and simulations.”

The above definition was taken from a “Scientific Applications of Linux” web site (e.g. www.sai.msu.su/sa1/A/1/) which provides a useful summary of the numerous CAS packages available. These packages range from commercial to Shareware to GPL (General Public License) software.

This paper discusses how CAS have been used recently to generate some Fortran subroutines that are part of an extensive suite of finite element programs described in detail by Smith and Griffiths (2004)¹.

The CAS used in the present work is Maple (www.maplesoft.com) which reflects the experience of the author, however the developments described in this paper could certainly have been achieved on other packages. A vital attribute of any CAS to be used for finite element software development however, is that it has the option of outputting complicated algebraic expressions directly into the Fortran language. This feature avoids the possibility of coding errors involving lengthy algebraic expressions, and largely removes the need for simplification routines in the CAS.

The strain-displacement matrix

Nearly all finite element programs for solid mechanics applications involve the generation of the element stiffness matrix. In 2-d solid elasticity, the element matrix can be written in the familiar form:

¹The 4th edition of this textbook is due for publication in the latter half of 2004. Many of the subroutines described are also to be found in the 3rd edition published in 1998

$$[\mathbf{k}_m] = \int \int [\mathbf{B}]^T [\mathbf{D}] [\mathbf{B}] dx dy \quad (1)$$

where $[\mathbf{B}]$ is the strain-displacement matrix containing derivatives of shape functions with respect to x and y , and $[\mathbf{D}]$ is the stress-strain matrix containing Young's modulus E and Poisson's ratio ν .

```

with(codegen,fortran); with(linalg);
deriv:=array(1..2,1..8):der:= array(1..2,1..8):
coord:=array(1..8,1..2):jac:= array(1..2,1..2):
jac1:=array(1..2,1..2):
##### coord are the nodal (x,y) coordinates
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:
coord[5,1]:=x5: coord[5,2]:=y5: coord[6,1]:=x6: coord[6,2]:=y6:
coord[7,1]:=x7: coord[7,2]:=y7: coord[8,1]:=x8: coord[8,2]:=y8:
##### terms closely related to local coordinates (xi,eta)
etam:=(1-eta): etap:=(1+eta): xim :=(1-xi): xip :=(1+xi):
##### shape function derivatives with respect to (xi,eta)
der[1,1]:=etam*(2*xi+eta)/4: der[1,2]:=-etam*etap/2:
der[1,3]:=etap*(2*xi-eta)/4: der[1,4]:=-etap*xip/4:
der[1,5]:=etap*(2*xi+eta)/4: der[1,6]:= etap*etam/2:
der[1,7]:=etam*(2*xi-eta)/4: der[1,8]:=-etam*xip/4:
der[2,1]:=xim*(xi+2*eta)/4: der[2,2]:=-xim*eta/4:
der[2,3]:=xim*(2*eta-xi)/4: der[2,4]:= xim*xip/2:
der[2,5]:=xip*(xi+2*eta)/4: der[2,6]:=-xip*eta/4:
der[2,7]:=xip*(2*eta-xi)/4: der[2,8]:=-xim*xip/2:
##### conversion to derivatives with respect to (x,y)
jac:=multiply(der,coord):jac1:=inverse(jac):
deriv:=multiply(jac1,der):
##### Fortran output
fortran(deriv,optimized);

```

Figure 1: Maple code to generate shape function derivatives in Fortran for a plane 8-node quadrilateral

Typically, finite element codes generate the $[\mathbf{B}]$ matrix at the element integrating points, firstly for the purposes of generating the stiffness matrix from (1) and secondly for obtaining strains once the displacements have been computed.

Figure 1 shows a typical Maple code for generating the derivative terms needed for $[\mathbf{B}]$ matrix at any local coordinate (ξ, η) given as input for a general 8-node quadrilateral element.

The program essentially mimics the coding that would be followed in a

conventional finite element program. The CAS performs everything symbolically however, so the resulting exact expressions can be very long indeed. For example, a typical term in the $[\mathbf{B}]$ matrix such as the derivative $\partial N_1/\partial x$ as a function of the 8 nodal (x, y) coordinates and the local coordinate (ξ, η) occupies well over 100 continuous lines of output. Simplification and factorization routines are available in most CAS packages, however in the author's experience these are often not as "intelligent" as one might wish.

Problems relating to the length of the expressions, particularly the possibility of typographical errors in conversion, can be largely overcome by requesting the output to be formatted directly into a high level programming language such as Fortran. This option has been requested in the example shown in Figure 1 by the command `with(codegen,fortran)`, and the output itself is generated by the command `fortran(deriv,optimized)`. In this example, all 16 derivative terms can be described in about 300 lines of Fortran. With some limited simplification of terms, this subroutine² is given as `bee8`.

The stiffness matrix of general quadrilaterals

Analytical expressions for the stiffness matrix of a *general* quadrilateral based on (1) have proved more problematic, especially for high order and 3-d elements. The main difficulty appears to be the complexity of the determinant of the Jacobian matrix in the denominator of the integrands. Several authors have tackled this problem, especially for low order elements. Babu and Pinder (1984) and Rathod (1988) developed exact expressions for 4-node quadrilaterals which led to logarithmic terms. Alternative formulations were proposed that led to series expansions. Yew *et al.*(1995) developed closed form integration formulas for "mixed" finite elements noting that these elements avoided entirely the need for the Jacobian determinant denominator. Videla *et al.*(1996) used the Derive software package to present explicit formulas for the stiffness matrix of a plane 4-node elastic element. Very favourable accuracy and timing comparisons were made with numerically integration formulations, especially for distorted elements. In spite of these developments, the advantages of numerical integration are hard to deny (e.g. Okabe 1981).

Griffiths (1994) presented semi-analytical formulations for the stiffness matrix of a general 4-node element, by symbolically performing conventional Gauss-Legendre quadrature using 4 integrating points per element. This approach is used by Smith and Griffiths(2004) in the subroutine `stiff4`. Griffiths (1993) and Griffiths and Mustoe(1995) continued this theme by considering symbolic expressions based on selective reduced integration (4

²See the Summary section for the web location of this and other subroutines mentioned in this paper.

integrating points for deviatoric and 1 integrating point for the volumetric) of a general 4-node element. This approach is recommended in order to avoiding “locking” as $\nu \rightarrow 0.5$ in low-order elements. These semi-analytical approaches gave identical results to those obtained using conventional numerical integration, but ran significantly faster. Using the same approach, Cardoso(1995) developed the general straight-sided 8-node plane-strain element stiffness matrix symbolically based on “reduced” integration (4 Gauss-points per element), available as subroutine `stiff8`.

It is generally accepted that 4-point Gauss-Legendre integration is “accurate enough” for a plane 4-node quadrilateral element providing it is not too distorted. Consider the general quadrilateral test element shown in Figure 2.

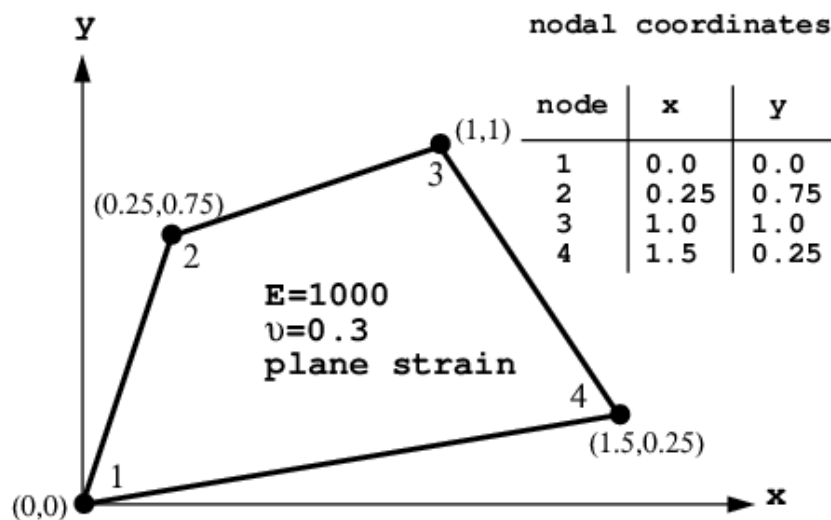


Figure 2: Four node planar test element

Using the conventional approach, if we integrate the term k_{11} of the element stiffness matrix using a gradually increasing number of integrating points (nip), convergence is rapidly obtained and nip=4 gives a good result as shown in Table 1.

Table 1. Influence of Gauss-Legendre integration order on stiffness term k_{11}

k_{11}	nip
267.86	1
350.04	4
351.65	9
351.70	16
351.70	25

The shape functions for the 4-node element shown in Figure 2 in terms of local coordinates (ξ, η) are well known as:

$$\begin{aligned} N_1 &= (1 - \xi)(1 - \eta)/4 & N_2 &= (1 - \xi)(1 + \eta)/4 \\ N_3 &= (1 + \xi)(1 + \eta)/4 & N_4 &= (1 + \xi)(1 - \eta)/4 \end{aligned} \quad (2)$$

however with symbolic software we can easily develop the shape functions in Cartesian coordinates. While it is possible to express these quite generally in terms of nodal coordinates, to avoid length expressions here, we present the shape functions only for the test element as:

$$\begin{aligned} N_1 &= +\frac{11}{54}\sqrt{144y^2 - 456y + 289 + 144x} - \frac{4}{3}x + \frac{14}{9}y - \frac{133}{54} \\ N_2 &= -\frac{10}{27}\sqrt{144y^2 - 456y + 289 + 144x} + \frac{4}{3}x - \frac{32}{9}y + \frac{170}{27} \\ N_3 &= +\frac{17}{54}\sqrt{144y^2 - 456y + 289 + 144x} - \frac{4}{3}x + \frac{38}{9}y - \frac{289}{54} \\ N_4 &= -\frac{4}{27}\sqrt{144y^2 - 456y + 289 + 144x} + \frac{4}{3}x - \frac{20}{9}y + \frac{68}{27} \end{aligned} \quad (3)$$

As a check the shape functions all satisfy the requirement that:

$$N_i(x_j, y_j) \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \quad \text{and} \quad \sum_{i=1}^4 N_i = 1 \quad (4)$$

Although these expressions appear highly nonlinear, they all reduce to linear functions, and the square root term disappears when the equation of any of the four sides is substituted. The benefits of local coordinates and numerical integration of stiffness become obvious if we look at the actual function to be integrated over the area of the quadrilateral region. Considering again the k_{11} term, we get for the plane-strain element shown in Figure 2:

$$k_{11} = \int \int^{Area} f(x, y) dx dy \quad (5)$$

where, making the substitution $d = 144y^2 - 456y + 144x + 289$, we get,

$$f(x, y) = 40000(54751 + (924y - 2849)\sqrt{d} + 21312y^2 - 67488y + 12600x)/(1053d) \quad (6)$$

Lichti(2003) discussed evaluation of (5) by a “naive” superposition strategy involving rectangles and right-angled triangles as shown in Figure 3. This has been completed by the author using the Maple program given in Figure 4.

The analytical expression for the stiffness term is given by:

$$\begin{aligned} k_{11} &= \frac{100}{85293} \{120467600 \ln 2 + 46766500 \ln 5 - 10747825 \ln 11 - 44996875 \ln 17\} \\ &\quad - \frac{19302500}{3159} = 351.69944 \end{aligned} \quad (7)$$

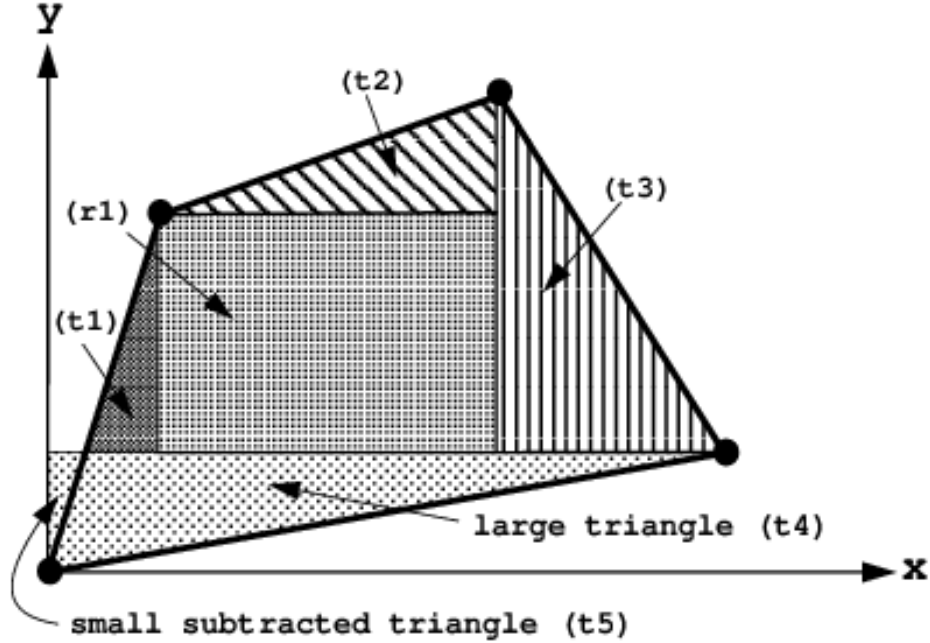


Figure 3: Superposition of regions for integration of test

The stiffness matrix of rectangular quadrilaterals

Analytical evaluation of the stiffness matrix of rectangular quadrilateral elements is considerably easier than for general quadrilaterals. When evaluating the integrals of (1), the shape functions can be formulated in Cartesian coordinates and the terms of the $[\mathbf{B}]$ matrix can be obtained by direct differentiation. Furthermore, integration can proceed by separation of the variables with fixed limits in the x - and y -directions as shown in Figure 5 and equation (8).

$$[\mathbf{k}_m] = \int_0^b \int_0^a [\mathbf{B}]^T [\mathbf{D}] [\mathbf{B}] dx dy \quad (8)$$

CAS are a useful tool for generating shape functions of finite elements. For example, the shape functions for a “new” elements such as the 14-node brick described by Smith and Griffiths(2004) were isolated using CAS. Here we demonstrate the derivation of a typical shape function of a rectangular 8-node elements in Cartesian coordinates. The terms of a typical shape function can be obtained from Pascal’s triangle as follows:

```

> restart; d:=144*y^2-456*y+144*x+289:
> fxy :=40000*(54751+(924*y-2849)*sqrt(d)+
> 21312*y^2-67488*y+12600*x)/(1053*d):
> r1:=int(int(fxy,x=1/4..1),y=1/4..3/4): # r1 main rectangle
> t1:=int(int(fxy,x=y/3..1/4),y=1/4..3/4): # t1 left triangle
> t2:=int(int(fxy,x=3*y-2..1),y=3/4..1): # t2 top triangle
> t3:=int(int(fxy,x=1..-2*y/3+5/3),y=1/4..1):# t3 right triangle
> t4:=int(int(fxy,x=0..6*y),y=0..1/4): # t4 lower (big) triangle
> t5:=int(int(fxy,x=0..y/3),y=0..1/4): # t5 lower(small) triangle
> k11:=r1+t1+t2+t3+t4-t5: k11:=evalf(%);
k11 := 351.69944

```

Figure 4: Maple code to generate analytical k_{11} term for the test element

$$N_1 = c_1 + c_2x + c_3y + c_4x^2 + c_5xy + c_6y^2 + c_7x^2y + c_8xy^2 \quad (9)$$

followed by the setting up and solution of 8 simultaneous equations from (4) for the c_i coefficients. A typical Maple program is shown in Figure 6.

Leading to the shape function,

$$N_1 = 1 - \frac{3x}{a} - \frac{3y}{b} + \frac{2x^2}{a^2} + \frac{5xy}{ab} + \frac{2y^2}{b^2} - \frac{2x^2y}{a^2b} - \frac{2xy^2}{ab^2} \quad (10)$$

A similar approach can be used for the other 7 shape functions of this element.

Although the exact stiffness matrices of both 4- and 8-node rectangular elements are easily evaluated analytically from (8), it is well known that reduced numerical integration in the case of the 8-node element can lead to better performance (e.g. Zienkiewicz and Taylor 1989). In the Smith and Griffiths(2004) text, CAS have been used to evaluate the exact stiffness matrix of a 4-node rectangular element by analytical integration and the “reduced” stiffness matrix of a rectangular 8-node element using 4 Gauss points. The coding for both these cases can be found in subroutine `rect_km`. For example, the exact 4-node rectangle gives,

$$k_{11} = \frac{1}{6} \frac{E}{(1+v)(1-2v)} \left\{ \frac{a}{b} (1-2v) + \frac{2b}{a} (1-v) \right\} \quad (11)$$

and the “reduced” integration 8-node rectangle gives,

$$k_{11} = \frac{5}{18} \frac{E}{(1+v)(1-2v)} \left\{ \frac{a}{b} (1-2v) + \frac{2b}{a} (1-v) \right\} \quad (12)$$

The similarity of the equations is striking even though they involve quite different elements and integration strategies. The stiffness term k_{11} in both cases is dependent on the aspect ratio a/b of the element, however some

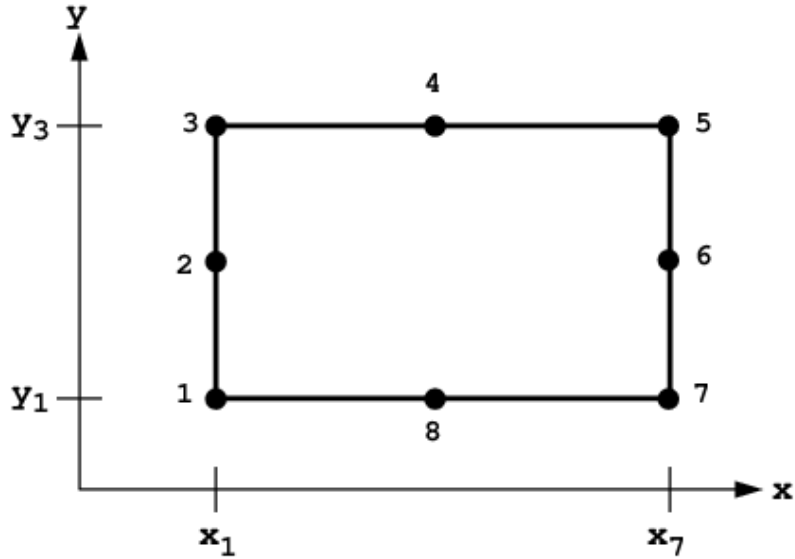


Figure 5: A rectangular 8-node element

```

> restart;
> N1:=c1+c2*x+c3*y+c4*x^2+c5*x*y+c6*y^2+c7*x^2*y+c8*x*y^2:
> eq1:=subs(x=0, y=0, N1)=1: eq2:=subs(x=0, y=b/2, N1)=0:
> eq3:=subs(x=0, y=b, N1)=0: eq4:=subs(x=a/2, y=b, N1)=0:
> eq5:=subs(x=a, y=b, N1)=0: eq6:=subs(x=a, y=b/2, N1)=0:
> eq7:=subs(x=a, y=0, N1)=0: eq8:=subs(x=a/2, y=0, N1)=0:
> eqs:={eq1,eq2,eq3,eq4,eq5,eq6,eq7,eq8}: vars:={c1,c2,c3,c4,c5,c6,c7,c8}:
> solve(eqs, vars);

```

Figure 6: Maple code to generate shape function N_1 for an 8-node rectangular element

off-diagonal terms of $[\mathbf{k}]$, when generated using a CAS, can be shown to be entirely independent of the dimensions of the element. For example, the exact 4-node rectangle gives,

$$k_{12} = \frac{1}{8} \frac{E}{(1+v)(1-2v)} \quad (13)$$

and the “reduced” integration 8-node rectangle gives,

$$k_{12} = \frac{17}{72} \frac{E}{(1+v)(1-2v)} \quad (14)$$

The conductivity matrix

This element matrix is used to model steady state and transient analyses in fluid or heat flow. The matrix is analogous to the stiffness matrix in solid analysis and is defined:

$$[\mathbf{k}_c] = \int \int [\mathbf{T}]^T [\mathbf{K}] [\mathbf{T}] dx dy \quad (15)$$

The similarity with (1) is obvious. Matrix $[\mathbf{T}]$ is analogous to $[\mathbf{B}]$ in solid analysis and holds derivatives of shape functions with respect to x and y . The matrix $[\mathbf{K}]$ is analogous to $[\mathbf{D}]$ in solid analysis and holds material conductivity properties. The development of “analytical” formulations of $[\mathbf{k}_c]$ follows essentially the same logic as for the stiffness matrix $[\mathbf{k}]$ discussed previously. If we consider a 4-node rectangular element, exact integration of $[\mathbf{k}_c]$ from (15) is easily achieved by direct integration and separation of variables. For general 4-node quadrilateral elements, analytical solutions for $[\mathbf{k}_c]$ are also possible using a similar strategy to that described earlier for stiffness leading to (7). Numerical integration using 4 Gauss points is undoubtedly simpler however, and generally considered to be “accurate enough”.

Table 2 shows the influence of increasing the order of numerical integration (`nip`) on term k_{c11} for the test element shown in Figure 2 assuming isotropic conductivity properties equal to unity.

Table 2. Influence of Gauss-Legendre integration order on conductivity term k_{c11}

k_{c11}	<code>nip</code>
0.51786	1
0.63460	4
0.63606	9
0.63609	16
0.63609	25

Subroutine `seep4` has been developed using CAS to compute $[\mathbf{k}_c]$ based on 4 Gauss points. The methodology used to generate this subroutine is essentially the same as that used to develop `stiff4` in solid elasticity.

The plastic matrix

In nonlinear analysis of elasto-plastic materials involving a yield function F , a potential function Q and an elastic stress-strain matrix $[\mathbf{D}^e]$, stresses are computed using a plastic constitutive matrix given by:

$$[\mathbf{D}^p] = \frac{[\mathbf{D}^e] \left\{ \frac{\partial Q}{\partial \boldsymbol{\sigma}} \right\} \left\{ \frac{\partial F}{\partial \boldsymbol{\sigma}} \right\}^T [\mathbf{D}^e]}{\left\{ \frac{\partial F}{\partial \boldsymbol{\sigma}} \right\}^T [\mathbf{D}^e] \left\{ \frac{\partial Q}{\partial \boldsymbol{\sigma}} \right\}} \quad (16)$$

Subroutines based on the analytical forms of the plastic matrix have been developed for both plane-strain and 3-d conditions using CAS for the case of materials with failure criteria described by von Mises (subroutine `vmdpl`):

$$F = \bar{\sigma} - \sqrt{3}c_u \quad (17)$$

where c_u is the undrained shear strength, and Mohr-Coulomb (subroutine `mcdpl`):

$$F = \sigma_m \sin \phi + \bar{\sigma} \left(\frac{\cos \theta}{\sqrt{3}} - \frac{\sin \theta \sin \phi}{3} \right) - c \cos \phi \quad (18)$$

$$Q = \sigma_m \sin \psi + \bar{\sigma} \left(\frac{\cos \theta}{\sqrt{3}} - \frac{\sin \theta \sin \psi}{3} \right) - c \cos \psi \quad (19)$$

where ϕ and c are the friction angle and cohesion of the soil, and ψ is the dilation angle.

The terms σ_m , $\bar{\sigma}$ and θ are stress invariants defined:

$$\sigma_m = \frac{\sigma_x + \sigma_y + \sigma_z}{3} \quad (20)$$

$$\bar{\sigma} = \frac{1}{\sqrt{2}} \left[(\sigma_x - \sigma_y)^2 + (\sigma_y - \sigma_z)^2 + (\sigma_z - \sigma_x)^2 + 6\tau_{xy}^2 + 6\tau_{yz}^2 + 6\tau_{zx}^2 \right]^{1/2} \quad (21)$$

$$\theta = \frac{1}{3} \arcsin \left(\frac{-27J_3}{\bar{\sigma}^3} \right) \quad (22)$$

where

$$J_3 = s_x s_y s_z - s_x \tau_{yx}^2 - s_z \tau_{zx}^2 - s_z \tau_{xy}^2 + 2\tau_{xy} \tau_{yz} \tau_{zx} \quad (23)$$

and

$$s_x = \frac{1}{3}(2\sigma_x - \sigma_y - \sigma_z) \quad \text{etc.} \quad (24)$$

Summary of CAS generated subroutines for finite element analysis

Table 3 shows a list of Fortran 95 subroutines described in the text by Smith and Griffiths (2004) and available for download from web site: www.mines.edu/fs_home/vgriffit/4th_ed/source/library/main/

Table 3. CAS generated subroutines

Name	Function
bee8	Returns the “analytical” form of the $[\mathbf{B}]$ matrix for general plane 8-node elements
mcdpl	Returns the plastic stress-strain matrix $[\mathbf{D}^p]$ for a Mohr-Coulomb material
rect_km	Returns the stiffness matrix $[\mathbf{k}]$ of a rectangular plane-strain 4-(exact) or 8-node (reduced) quadrilateral element
seep4	Returns the “analytical” conductivity matrix $[\mathbf{k}_c]$ for a general plane 4-node element based on 4 Gauss points.
stiff4	Returns the “analytical” stiffness matrix $[\mathbf{k}]$ of a general plane-strain 4-node quadrilateral element based on 4 Gauss points.
stiff8	Returns the “analytical” stiffness matrix $[\mathbf{k}]$ of a general plane-strain 8-node quadrilateral element based on 4 Gauss points.
vmdpl	Returns the plastic stress-strain matrix $[\mathbf{D}^p]$ for a von Mises material.

Concluding Remarks

Computer Algebra Systems (CAS) offer great potential for generation of engineering analysis software. Furthermore, the analytical expressions produced by CAS, can give insight into the nature of finite element terms that are usually manipulated numerically, and rarely seen in closed form. This paper has described some applications of CAS that were used in the development of a suite of finite element programs developed by Smith and Griffiths (2004). In most cases the Fortran code was generated directly by the CAS from the algebraic expressions, which were occasionally lengthy and complex. Examples covered include strain-displacement matrices, quadrilateral element stiffness matrices (rectangular and general), conductivity matrices and plastic matrices for nonlinear analysis. The source code for all the Fortran 95 subroutines described in this paper can be downloaded from the author’s web page. The work described in this paper is a step towards the goal of creating a comprehensive library of CAS-generated software, available in the public domain covering a wide range of FE element types and applications.

Acknowledgement

The work described in this paper is part of a collaborative project between

the Colorado School of Mines and the Universidad Central de Venezuela entitled, “Finite element software development using computer algebra systems”. The support of the NSF Americas Program Proposal Number 0106665 is acknowledged.

References

- [1] D. Babu and G. F. Pinder. Analytical integration formulae for linear isoparametric finite elements. *Int J Numer Methods Eng*, 20(6):1153–1166, 1984.
- [2] J-P. Cardoso. Generation of finite element matrices using computer algebra. Master’s thesis, School of Engineering, University of Manchester, 1994.
- [3] D. V. Griffiths. Use of computer algebra systems to generate element stiffness matrices. In J. R. Whitehead, editor, *The Mathematics of Finite Elements and Applications-Highlights 1993*, pages 373–374. John Wiley and Sons, Chichester, New York, 1993.
- [4] D. V. Griffiths. Stiffness matrix of the four-node quadrilateral element in closed-form. *Int J Numer Methods Eng*, 37(6):1027–1038, 1994.
- [5] D. V. Griffiths and G. G. W. Mustoe. Selective reduced integration of the four node plane element in closed-form. *J Eng Mech, ASCE*, 121(6):725–729, 1995.
- [6] J. Lichti. Generation of an analytical stiffness matrix for a four-node plane strain finite element. Master’s thesis, Division of Engineering, Colorado School of Mines, 2003.
- [7] M. Okabe. Analytical integration formulae related to convex quadrilateral finite elements. *Comput Methods Appl Mech Eng*, 29:201–208, 1978.
- [8] H.T. Rathod. Some analytical integration formulae for a four node isoparametric element. *Comput Struc*, 30(5):1101–1109, 1988.
- [9] I. M. Smith and D. V. Griffiths. *Programming the Finite Element Method*. John Wiley and Sons, Chichester, New York, 4th edition, 2004.
- [10] L. Videla, M. Cerrolaza, and N. Aparicio. Explicit integration of the stiffness matrix of a four-node plane elasticity finite element. *Comm Appl Numer Methods*, 12(11):731–743, 1996.

- [11] C. K. Yew, J. T. Boyle, and D. MacKenzie. Closed form integration of element stiffness matrices using a computer algebra system. *Comput Struc*, 56(4):529–539, 1995.