Rocfrac Development Guide

by Scot Breitenfeld[1]

11th December 2003

[1]This file is currently maintained by SCOT BREITENFELD, brtnfld@uiuc.edu. Please send comments or error corrections to that address.

Contents

Chapter 1

Theory and Concepts

For an introduction to the fundamental theories and concepts behind the Cohesive Volumetric Finite Element (CVFE) scheme the reader is encouraged to review Jeff Baylor's 1995 UIUC Thesis "A numerical Simulation of impact-induced damage of composite materials".

1.1 Code Structure

This section describes the basic algorithm and structure of the code.

The code uses for the direct integration the explicit central-difference method which approximates the nodal displacements, accelerations and velocities as

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \Delta t \dot{\mathbf{u}}_n + \frac{1}{2}\Delta t^2 \ddot{\mathbf{u}}_n, \tag{1.1}$$

$$\dot{\mathbf{u}}_{n+1} = \dot{\mathbf{u}}_n + \frac{\Delta t}{2}\left(\ddot{\mathbf{u}}_n + \ddot{\mathbf{u}}_{n+1}\right), \tag{1.2}$$

$$\ddot{\mathbf{u}}_{n+1} = -\mathbf{M}^{-1}\left(\mathbf{R}_{in_{n+1}} - \mathbf{R}_{co_{n+1}} + \mathbf{R}_{ex}\right). \tag{1.3}$$

Where, Δt is the time step, $\mathbf{M}$ is the lumped mass matrix, and $\mathbf{R}_{in}, \mathbf{R}_{co}, \mathbf{R}_{ex}$ are the global internal force, cohesive force and external force vectors, respectively. In the numerical implementation the sum within the brackets of equation 1.3 is replaced with one vector called $\mathbf{R}_{net}$, yet the argument name used in the subroutines that calculate the respective force vectors still retains the original force variable naming.

The numerical algorithm is as follows:

- Data input.
- Set initial conditions ($\mathbf{u}(0)$, $\dot{\mathbf{u}}(0)$, $\ddot{\mathbf{u}}(0)$).
- Solution time increment loop:
  - Update the nodal displacements.

$$u_{n+1} = u_n + \Delta t \dot{u}_n + \frac{1}{2}\Delta t^2 \ddot{u}_n$$

- Update mesh position.
- Calculate the mesh normals at the nodes.
- Update mass matrix due to change in undeformed configuration.
- Calculate R_bar.
- Calculate mesh velocity vector.
- Compute the internal force vector R_{co}.
- Calculate the mesh acceleration.
- Compute the internal force vector R_{in}.
- Update the nodal velocity and accelerations.

$$\ddot{u}_{n+1} = M^{-1} R_{net_{n+1}},$$

$$\dot{u}_{n+1} = \dot{u}_n + \frac{\Delta t}{2}\left(\ddot{u}_n + \ddot{u}_{n+1}\right).$$

- Apply displacement boundary conditions.

1.2 Call Graph

This section describes the call graph of the subroutines. The main driver code is called rocfrac.f. The call graph is pictured in Figure 1.1. Not that the calls to the MPI libraries are not included.

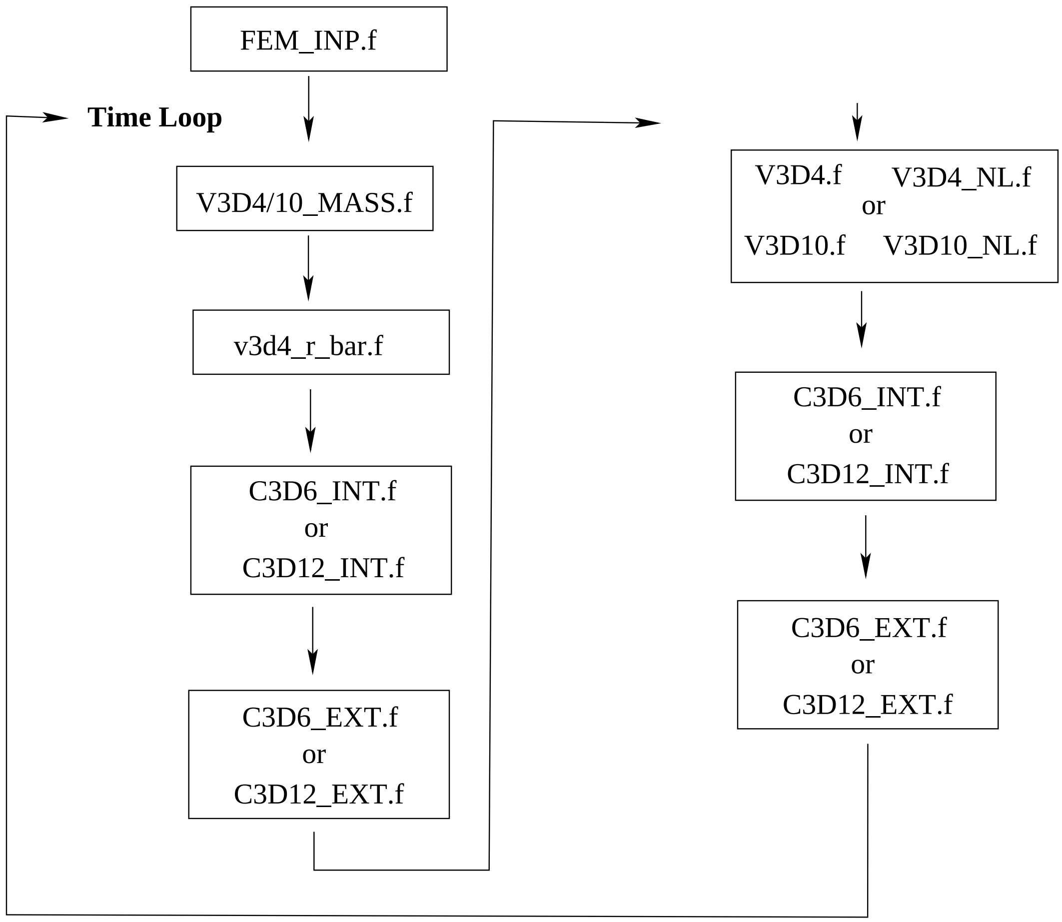

Figure 1.1: Call Tree for rocfrac

Chapter 2

Subroutines and Functions

This section contains a detail listing of the subroutines and functions and their purpose, notes, inputs and output.

2.1 Rocfrac/Rocfrac/Source/angle_rad_3d.f90

NAME

```
    angle_rad_3d
```

FUNCTION

```
    returns the angle in radians between two rays in 3D.
```

NOTES

```
   The routine always computes the SMALLER of the two angles between
   two rays.  Thus, if the rays make an (exterior) angle of
   1.5 radians, the (interior) angle of 0.5 radians will be reported.
 Formula:
   X dot Y = Norm(X) * Norm(Y) * Cos ( Angle(X,Y) )
```

INPUTS

```
real X1, Y1, Z1, X2, Y2, Z2, X3, Y3, Z3, are three points
which define the rays.  The rays are:
( X1-X2, Y1-Y2, Z1-Z2 ) and ( X3-X2, Y3-Y2, Z3-Z2 ).
```

OUTPUT

```
real ANGLE_RAD_3D, the angle between the two rays, in radians.
This value will always be between 0 and PI.  If either ray has
zero length, then the angle is returned as zero.
```

2.2 Rocfrac/Rocfrac/Source/CauchyStressPrinc.f90

NAME

```
CauchyStressPrinc
```

FUNCTION

```
Determines the Cauchy stress tensor for materials defined in
principal directions
```

INPUTS

```
ndime  -->  number of dimensions
xmu    -->  mu coefficient
xlamb  -->  lambda coefficient
detf   -->  determinant of F, i.e. J
```

```
stret  -->  vector containing the stretches

btens  -->  matrix containing the the principal directions
```

OUTPUT

```
sigma  -->  Cauchy stress tensor

sprin  -->  principal stresses
```

2.3 Rocfrac/Rocfrac/Source/arruda_boyce.f90

NAME

```
ARRUDA_BOYCE
```

FUNCTION

```
Arruda-Boyce constitutive model, returns

2nd Piola-Kircheoff Stresses
```

INPUTS

```
F11,F12,F13,F21,F22,F23,F31,F32,F33 -- componets

     of the deformation gradient [F]

mu, kappa -- material parameters

ielem -- element id number
```

OUTPUT

```
S11,S22,S33,S12,S23,S13  -- componets of

      the 2nd Piola-Kircheoff Stresses
```

USES

```
rs, Solve_x
```

2.4 Rocfrac/Rocfrac/Source/arruda_boyce.f90/Solve_x

NAME

```
Solve_x
```

FUNCTION

```
Solve for the x
```

INPUTS

```
    xmu -- mu
      i -- element id
stretch -- principle stretch
```

OUTPUT

```
      x  -- x
```

2.5 Rocfrac/Rocfrac/Source/arruda_boyce.f90/rs

NAME

```
rs
```

FUNCTION

```
calls the recommended SEQUENCE of
subroutines from the eigensystem SUBROUTINE package (eispack)
to find the eigenvalues and eigenvectors (IF desired)
of a REAL symmetric matrix.
```

INPUTS

```
nm  must be set to the row DIMENSION of the two-dimensional
 array parameters as declared in the calling PROGRAM
 DIMENSION statement.
n  is the order of the matrix  a.
a  CONTAINS the REAL symmetric matrix.
matz  is an INTEGER variable set equal to zero IF
   ONLY eigenvalues are desired.  otherwise it is set to
   any non-zero INTEGER for both eigenvalues and eigenvectors.
```

OUTPUT

```
w  CONTAINS the eigenvalues in ascending order.
z  CONTAINS the eigenvectors IF matz is not zero.
ierr  is an INTEGER output variable set equal to an error
      completion code described in the documentation for tqlrat
      and tql2.  the normal completion code is zero.
fv1  and  fv2  are temporary storage arrays.
```

NOTES

```
questions and comments should be directed to burton s. garbow,
mathematics and computer science div, argonne national laboratory
this version dated august 1983.
```

2.6 Rocfrac/Rocfrac/Source/arruda_boyce.f90/pythag

NAME

```
pythag
```

FUNCTION

```
finds SQRT(a**2+b**2) without overflow or destructive underflow
```

2.7 Rocfrac/Rocfrac/Source/arruda_boyce.f90/tql1

NAME

```
tql1
```

FUNCTION

```
finds the eigenvalues of a symmetric
tridiagonal matrix by the ql method.
```

INPUTS

```
n is the order of the matrix.
d CONTAINS the diagonal elements of the input matrix.
```

```
    e CONTAINS the subdiagonal elements of the input matrix
      in its last n-1 positions.  e(1) is arbitrary.
```

OUTPUT

```
    d CONTAINS the eigenvalues in ascending order.  IF an
      error EXIT is made, the eigenvalues are correct and
      ordered for indices 1,2,...ierr-1, but may not be
      the smallest eigenvalues.
    e has been destroyed.
    ierr is set to
      zero       for normal RETURN,
      j          IF the j-th eigenvalue has not been
                 determined after 30 iterations.
```

USES

```
  pythag for SQRT(a*a + b*b) .
```

NOTES

```
  this SUBROUTINE is a translation of the algol PROCEDURE tql1,
  num. math. 11, 293-306(1968) by bowdler, martin, reinsch, and
  wilkinson.
  handbook for auto. comp., vol.ii-linear algebra, 227-240(1971).
  questions and comments should be directed to burton s. garbow,
```

```
mathematics and computer science div, argonne national laboratory
this version dated august 1983.
```

2.8 Rocfrac/Rocfrac/Source/arruda_boyce.f90/tql2

NAME

```
tql2
```

FUNCTION

```
Finds the eigenvalues and eigenvectors
of a symmetric tridiagonal matrix by the ql method.
the eigenvectors of a full symmetric matrix can also
be found IF  tred2  has been used to reduce this
full matrix to tridiagonal form.
```

NOTES

```
this SUBROUTINE is a translation of the algol PROCEDURE tql2,
num. math. 11, 293-306(1968) by bowdler, martin, reinsch, and
wilkinson.
handbook for auto. comp., vol.ii-linear algebra, 227-240(1971).
questions and comments should be directed to burton s. garbow,
mathematics and computer science div, argonne national laboratory
this version dated august 1983.
```

INPUTS

```
nm must be set to the row DIMENSION of two-dimensional
  array parameters as declared in the calling PROGRAM
  DIMENSION statement.
n is the order of the matrix.
d CONTAINS the diagonal elements of the input matrix.
e CONTAINS the subdiagonal elements of the input matrix
  in its last n-1 positions.  e(1) is arbitrary.
z CONTAINS the transformation matrix produced in the
  reduction by  tred2, IF performed.  IF the eigenvectors
  of the tridiagonal matrix are desired, z must contain
  the identity matrix.
```

OUTPUT

```
d CONTAINS the eigenvalues in ascending order.  IF an
  error EXIT is made, the eigenvalues are correct but
  unordered for indices 1,2,...,ierr-1.
e has been destroyed.
z CONTAINS orthonormal eigenvectors of the symmetric
  tridiagonal (or full) matrix.  IF an error EXIT is made,
  z CONTAINS the eigenvectors associated WITH the stored
  eigenvalues.
ierr is set to
  zero       for normal RETURN,
```

```
    j          IF the j-th eigenvalue has not been
               determined after 30 iterations.
```

USES

```
pythag for  SQRT(a*a + b*b) .
```

2.9 Rocfrac/Rocfrac/Source/arruda_boyce.f90/tred1

NAME

```
tred1
```

FUNCTION

```
Reduces a REAL symmetric matrix
to a symmetric tridiagonal matrix using
orthogonal similarity transformations.
```

INPUTS

```
   nm must be set to the row DIMENSION of two-dimensional
     array parameters as declared in the calling PROGRAM
     DIMENSION statement.
   n is the order of the matrix.
   a CONTAINS the REAL symmetric input matrix.  ONLY the
     lower triangle of the matrix need be supplied.
```

OUTPUT

```
a CONTAINS information about the orthogonal trans-
  formations used in the reduction in its strict lower
  triangle.  the full upper triangle of a is unaltered.
d CONTAINS the diagonal elements of the tridiagonal matrix.
e CONTAINS the subdiagonal elements of the tridiagonal
  matrix in its last n-1 positions.  e(1) is set to zero.
e2 CONTAINS the squares of the corresponding elements of e.
  e2 may coincide WITH e IF the squares are not needed.
```

NOTES

```
this SUBROUTINE is a translation of the algol PROCEDURE tred1,
num. math. 11, 181-195(1968) by martin, reinsch, and wilkinson.
handbook for auto. comp., vol.ii-linear algebra, 212-226(1971).
questions and comments should be directed to burton s. garbow,
mathematics and computer science div, argonne national laboratory
this version dated august 1983.
```

2.10 Rocfrac/Rocfrac/Source/arruda_boyce.f90/tred2

NAME

```
tred2
```

FUNCTION

```
reduces a REAL symmetric matrix to a
```

```
symmetric tridiagonal matrix using and accumulating
orthogonal similarity transformations.
```

INPUTS

```
nm must be set to the row DIMENSION of two-dimensional
  array parameters as declared in the calling PROGRAM
  DIMENSION statement.
n is the order of the matrix.
a CONTAINS the REAL symmetric input matrix.  ONLY the
  lower triangle of the matrix need be supplied.
```

OUTPUT

```
d CONTAINS the diagonal elements of the tridiagonal matrix.
e CONTAINS the subdiagonal elements of the tridiagonal
  matrix in its last n-1 positions.  e(1) is set to zero.
z CONTAINS the orthogonal transformation matrix
  produced in the reduction.
a and z may coincide.  IF distinct, a is unaltered.
```

NOTES

```
this SUBROUTINE is a translation of the algol PROCEDURE tred2,
num. math. 11, 181-195(1968) by martin, reinsch, and wilkinson.
handbook for auto. comp., vol.ii-linear algebra, 212-226(1971).
questions and comments should be directed to burton s. garbow,
```

```
mathematics and computer science div, argonne national laboratory

this version dated august 1983.
```

2.11 Rocfrac/Rocfrac/Source/cal_shdx.f90

NAME

```
cal_shdx
```

FUNCTION

```
computes shape fn. values and parametric derivatives at Gauss pts.
```

INPUTS

```
  meshpos -- mesh coordinates

  ndim    -- dimension of problem

 nnode    -- number of nodes

 nintk    -- number of integration points

surf_eleme --  connectivity of surf_elem

  iface  -- face number of tetrahedral
```

OUTPUT

```
shdx -- shape fn. derivatives
```

2.12 Rocfrac/Rocfrac/Source/shcalc_3d10.f90

NAME

```
shcalc_3d10
```

FUNCTION

```
EVALUALTE SHAPE FUNCTION AND ITS DERIVATIVES FOR 10-NODE TET.
```

AUTHOR

```
written by Changyu Hwang  Nov. 20, 2001
```

2.13 Rocfrac/Rocfrac/Source/ainv.f90

NAME

```
ainv
```

FUNCTION

```
Computes the det and inverse of a (3x3) matrix
```

USED BY

```
shcalc, shcalc_3d10
```

INPUTS

```
ndim -- size of input array (must be 3)
ajac -- Input array (ndim x ndim)
```

OUTPUT

```
ajacin -- inverse of ajac
det -- determinate of ajac
```

2.14 Rocfrac/Rocfrac/Source/shcalc.f90

NAME

```
shcalc
```

FUNCTION

```
EVALUALTE SHAPE FUNCTION AND ITS DERIVATIVES FOR 4-NODE TET.
```

2.15 Rocfrac/Rocfrac/Source/v3d4_r_bar.f90

NAME

```
v3d4_r_bar
```

FUNCTION

```
           _
Caluculates R for the ALE forumulation for
the 4-node tetrahedral
```

INPUTS

```
d_bar -- Displacement from mesh motion and
       surface regression
numnp -- Number of Nodes
NumElv -- Number of volumetric elements
lmcstet -- Element connectivity
meshcoor -- mesh coordinates
```

```
nstart -- starting element number

nend -- ending element number
```

OUTPUT

```
          _
 Rbar -- R additional internal force from ALE
```

2.16 Rocfrac/Rocfrac/Source/vol_elem_mat.f90

NAME

```
VOL_ELEM_MAT
```

FUNCTION

```
Forms the material compliance matix, [C] for an isotropic material
               -1
Note: [E] = [C]    and {epsilon} = [C]{sigma}
```

INPUTS

```
  e -- Young's modulus

 xnu -- Possion's ratio

numat_vol -- number of volumetric elements

Integration -- = 1 pseudo-Reduced integration (split Cijkl)

               = 0 Cijkl
```

OUTPUT

```
     ci -- elastic stiffness constants

     cj -- split stiffness constants
```

2.17 Rocfrac/Rocfrac/Source/v3d4n_nl.f90

NAME

```
    v3d4n_nl
```

FUNCTION

```
    Computes the internal force vector for a 4-node tetrahedral

    NODE BASED ELEMENT  for finite deformations.
```

INPUTS

```
    NumNP -- Number of nodes

   NumEL  -- Number of elements

    coor  -- number of coordinates

   disp   -- Nodal Displacement

   nodes --  Nodal connectivity

   NumElNeigh -- Number of elements in contact with node

   ElConn -- Element connectivity

   alpha -- volume ratio

   Ahat -- undeformed volume of node

   NumMatVol -- number of materials
```

```
xmu -- material parameter

xlambda -- material parameter
```

OUTPUT

```
Rnet -- internal force vector
```

2.18 Rocfrac/Rocfrac/Source/vol_elem_mat.f90

NAME

```
VOL_ELEM_MAT
```

FUNCTION

```
Caluculates the volume and surface area for tetraderal
elements
```

2.19 Rocfrac/Rocfrac/Source/principal_stress.f90

NAME

```
principal_stress
```

FUNCTION

```
Computes Principal Values of Symmetric Second Rank Tensor
```

INPUTS

```
S = Symmetric Second-Rank Tensor Stored as a Vector
```

```
        P = Principal Values

     .. The Components of S Must be Stored in the Following Orders

        2-D Problems,  S11,S12,S22

        3-D Problems,  S11,S12,S13,S22,S23,S33

   OUTPUTS

       SVonMises -- VonMises Stress
```

2.20 Rocfrac/Rocfrac/Source/max_dt_solid.f90

NAME

```
     max_dt_solid
```

FUNCTION

```
     Determines the maximum time step
```

INPUTS

```
     glb -- global array
```

OUTPUT

```
     dt_courant -- maximum time step
```

2.21 Rocfrac/Rocfrac/Source/jacobi.f90

NAME

```
    jacobi
```

FUNCTION

```
Evaluates the stretches and principal directions given the b matrix
using the Jacobi iteration. Adapted from numerical recpies
```

INPUTS

```
btens  -->  left Cauchy-Green tensor
```

OUTPUT

```
stret  -->  vector containing the stretches
princ  -->  matrix containing the three principal column vectors
```

2.22 Rocfrac/Rocfrac/Source/feminp.f90

NAME

```
feminp
```

FUNCTION

```
READ INPUT INFORMATION (i.e. Analysis Deck File)
```

INPUTS

```
glb -- global array
myid -- processor id (starting at 0)
```

2.23 Rocfrac/Rocfrac/Source/feminp/PREFIX_SUB

NAME

```
PREFIX_SUB
```

FUNCTION

```
Reads prefix keyword (i.e. Analysis Deck File)
```

INPUTS

```
glb -- global array
```

2.24 Rocfrac/Rocfrac/Source/feminp/ALE_SUB

NAME

```
ALE_SUB
```

FUNCTION

```
ALE keyword turns on ALE routines
```

INPUTS

```
glb -- global array
```

2.25 Rocfrac/Rocfrac/Source/feminp/ALE_SUB

NAME

```
ALE_SUB
```

FUNCTION

```
Courant limit multiplier
```

INPUTS

```
glb -- global array

keywd -- keywd for control deck
```

2.26 Rocfrac/Rocfrac/Source/feminp/MATMODEL_HYPERELA

NAME

```
MATMODEL_HYPERELASTIC
```

FUNCTION

```
reads material model type, and the

hyperelastic material parameters
```

INPUTS

```
glb -- global array

keywd -- keywd for control deck
```

OUTPUT

```
tmp_E -- Young's Modulus

tmp_xnu -- Possion's ratio

tmp_rho -- Density

tmp_alpha -- thermal coefficient of expansion

tmp_iSolnType -- material type model
```

2.27 Rocfrac/Rocfrac/Source/feminp/MATMODEL_ELASTIC

NAME

```
MATMODEL_ELASTIC
```

FUNCTION

```
reads material model type, and the
elastic material parameters
```

INPUTS

```
glb -- global array
keywd -- keywd for control deck
```

OUTPUT

```
tmp_E -- Young's Modulus
tmp_xnu -- Possion's ratio
tmp_rho -- Density
tmp_alpha -- thermal coefficient of expansion
tmp_iSolnType -- material type model
```

2.28 Rocfrac/Rocfrac/Source/feminp/BOUNDARY_SUB

NAME

```
BOUNDARY_SUB
```

FUNCTION

```
This option is used to prescibe boundary conditions
at nodes.
```

INPUTS

```
glb -- global array
```

2.29 Rocfrac/Rocfrac/Source/feminp/BOUNDARY_SUB

NAME

```
BOUNDARY_SUB
```

FUNCTION

```
This option is used to prescibe mesh motion boundary
conditions at nodes.
```

INPUTS

```
glb -- global array
```

2.30 Rocfrac/Rocfrac/Source/feminp/ELEMENT_SUB

NAME

```
ELEMENT_SUB
```

FUNCTION

```
Specifies the element type
```

INPUTS

```
glb -- global array
```

2.31 Rocfrac/Rocfrac/Source/bc_enforce.f90

NAME

```
bc_enforce
```

FUNCTION

```
Enforces the structural boundary conditions
```

INPUTS

```
numbound -- number of nodes with enforced boundary
       conditions
   numnp -- number of nodes
      id -- element id
       r -- imposed type of boundary condition
   slope -- loading amplitude slope
   prop  -- proportion of amplitude for loading
     vb  -- imposed velocity
     ab  -- impoesed acceleration
  delta  -- time increment
   Rnet  -- sum of forces
     xm  -- lumped nodal mass matrix
```

```
 DampEnabled  -- flag for damping

     CurrTime -- current time
```

OUTPUT

```
        v  -- nodal velocity with bc

        a  -- nodal acceleration with bc

        d  -- nodal displacement with bc
```

2.32 Rocfrac/Rocfrac/Source/arruda_boyce.cauchy.f90

NAME

```
ARRUDA_BOYCE_CAUCHY
```

FUNCTION

```
Arruda-Boyce constitutive model, returns Cauchy Stress
```

INPUTS

```
F11,F12,F13,F21,F22,F23,F31,F32,F33 -- componets

     of the deformation gradient [F]

mu, kappa -- material parameters

ielem -- element id number
```

OUTPUT

```
Cchy11, Cchy22, Cchy33,Cchy12,Cchy13,Cchy23  -- componets

of the Cauchy stress tensor
```

USES

```
rs, Solve_x
```

2.33 Rocfrac/Rocfrac/Source/VolRatio.f90

NAME

```
VolRatio
```

FUNCTION

```
Calculates the Volume Ratio for the node based elements
using either the circumcenter of centroid.
```

INPUTS

```
n1,n2,n3,n4 -- node number
coor -- nodeal coordinates
numnp -- number of nodes
NdMassLump -- = 1 then circumcenter method
              = 0 then centroid method
```

OUTPUT

```
AlphaR -- Ratio of the Volume for a node
```

2.34 Rocfrac/Rocfrac/Source/UpdateMassMatrix.f90

NAME

```
UpdateMassMatrix
```

FUNCTION

```
Updates the mass matrix due to change in undeformed
configuration. Calls the appriopate mass matrix subroutine
and handles the parallel communication.
```

INPUTS

```
glb -- global variables
```

USES

```
GENX_RocFrac, V3D4_MASS, V3D4N_MASS, V3D10_MASS
```

2.35 Rocfrac/Rocfrac/Source/RocFracMain

NAME

```
RocFracMain.f90
```

FUNCTION

```
3D Dynamic  Explicit Code  with  ALE  formulation for regressing
boundaries Finite Element Analysis Code with additional fracture
simulation using cohesive elements.
```

USAGE

```
Finite  Element code to solve the 3-Dimensional TRANSIENT structural problem
```

USES

```
  RocFracSubInterface, UpdateStructuralSoln, feminp, VolRatio, vol_elem_mat,
  RocFracInterfaceInitial, RocFracInterfaceBuff,  UpdateMassMatrix, V3D4_volum
  max_dt_solid, UpdateRbar,v3d4_ale,V3D10_ALE,FluidPressLoad, TractPressLoad,
  UpdateStructural,principal_stress,bc_enforce


  Global variables stored in modules :  GENX_RocFrac,GENX_RocFracComm,GENX_Roc
```

COPYRIGHT

```
  University of Illinois, Urbana-Champaign, (C) 2003
```

AUTHOR

```
principal : M.S. Breitenfeld, P.H. Geubelle
  - email : brtnfld@uiuc.edu, geubelle@uiuc.edu
contributing : Changyu Huang, Amit Acharya
```

CREATION DATE

```
     2001
```

2.36 Rocfrac/Rocfrac/Source/RocFracMain

NAME

```
  RocFracInterfaceBuff
```

FUNCTION

```
Passes the variables to the fluid code. It transfers the new mesh
```

```
velocity and displacement to the interface mesh  arrays that are
registered with RocCom
```

2.37 Rocfrac/Rocfrac/Source/IntegralCheck.f90

NAME

```
CheckIntegral
```

FUNCTION

```
Sums the conservation quantities over all the
processors. MPI_REDUCE
```

INPUTS

```
glb -- global array
```

OUTPUT

```
IntegralArray -- Conservation term:
            1. Volume
            2. Mass
            3. x-momentum
            4. y-momentum
            5. z-momentum
            6. energy
            7. burning area
            8. non-burning area
```

2.38 Rocfrac/Rocfrac/Source/GENX_RocFracInterp.f90

NAME

```
GENX_RocFracInterp
```

FUNCTION

```
Global module for interface variables
```

2.39 Rocfrac/Rocfrac/Source/GENX_RocFracComm.f90

NAME

```
GENX_RocFracComm
```

FUNCTION

```
Global MPI array modules
```

2.40 Rocfrac/Rocfrac/Source/GENX_RocFrac.f90

NAME

```
GENX_RocFrac
```

FUNCTION

```
Global variable module
```

2.41 Rocfrac/Rocfrac/Source/FluidPressLoad.f90

NAME

```
FluidPressLoad
```

FUNCTION

```
Transforms the pressure given in the deformed state to
the undeformed configuration. This subroutine is for
the formulation where all quantities are with respect to
the undeformed configuration.  Assumes the
pressure (tractions) are constant over the surface of
the triangle.
1-3 for 4 node tet (i.e. 3 node triangles)
4-6 for 10 node tet (i.e. 6 node triangles, mid-side nodes get traction)
```

2.42 Rocfrac/Rocfrac/Source/locchr.f90/locchr

NAME

```
locchr
```

FUNCTION

```
Locates the keyword value after a keyword
```

INPUTS

```
text -- character string
varna -- variable name to search for
lvari -- length of the variable name
kpos0 -- initial position in 'text' so start looking for varna
```

OUTPUT

```
kpos1 -- Start of keyword value in string

kpos2 -- End of keywork value in string
```

2.43 Rocfrac/Rocfrac/Source/locchr.f90/conchr

NAME

```
conchr
```

FUNCTION

```
To determine if a control deck keyword is specified
```

INPUTS

```
text -- character string

varna -- variable name to search for

lvari -- length of the variable name

kpos0 -- initial position in 'text' so start looking for varna
```

OUTPUT

```
key --  0 = no, 1 = yes
```

2.44 Rocfrac/Rocfrac/Source/locchr.f90/dtext

NAME

```
dtext
```

FUNCTION

```
To determine the string length
```

INPUTS

```
text -- character string
```

OUTPUT

```
lll -- length of string
```

2.45 Rocfrac/Rocfrac/Source/locchr.f90/dtext

NAME

```
dtext
```

FUNCTION

```
Converts a character string to an integer
```

INPUTS

```
char -- character string
```

OUTPUT

```
key -- integer
```

2.46 Rocfrac/Rocfrac/Source/locchr.f90/rchar

NAME

```
rchar
```

FUNCTION

```
Converts a character string to a real
```

INPUTS

```
char -- character string
```

OUTPUT

```
key -- real
```

2.47 Rocfrac/Rocfrac/Source/ALEUpdateMassMatrix.f90

NAME

```
ALEUpdateMassMatrix
```

FUNCTION

```
Updates the nodal coordinates as a result

of the regressing boundaries. Calculates
```

```
the new lumped inverse mass matrix from the new
nodal coordinates, MPI calls handle
communication between partition boundaries
```

USED BY

```
RocfracMain
```

USES

```
GENX_RocFrac -- Global variables
V3D4_MASS, V3D4N_MASS, V3D10_MASS
```

INPUTS

```
glb -- global array
```

```
OUTPUTS
  xmass -- Inverse lumped mass matrix
```