

StressRefine Library Programmers guide

Conventional finite elements only require the local node numbers associated with an element. p-adaptive elements also require definitions of edges and faces to assure basis function continuity.

StressRefine creates these automatically, but it does require 3 setup calls.

1. allocate space for elements. this is the number of conventional elements in your model that you want to convert to p-adaptive. If your model has 1,000,000 but you only need to convert 50,000, allocate space for 50,000
2. Loop over all the elements that you want to convert.

```
int id = 0
```

```
Loop over all elements
```

```
//if want to convert this element:
```

```
model.CreateElem(int id, int userid, nnodes, int nodes[], mat)
```

```
id++
```

```
End loop
```

in CreateElem, nnodes is the number of corner and midnodes in the element, e.g, 10 for a quadratic tet, nodes is the vector of node numbers for the element (corners first followed by midedges), and "mat" contains the element material properties, described below. Also the element numbering convention for stressrefine is shown below.

3. call model.FillGlobalFaces() faces are automatically allocated and created, with global node order assigned for continuity.

Henceforth you can calculate element stiffnesses the same as for an isoparametric element, except the isoparametric shape functions are only used for mapping (e.g. calculating Jacobian). For displacement basis functions, the functions in the library are used:

```
model.basis.ElementBasisFuncs(double r, double s, double t, SRelement* elem, double* basisvec,  
double* dbasisdr, double* dbasisds, double* dbasisdt, SRBasisCallType calltype = both)
```

The return value is total number of functions for the element, and this outputs the basis functions in basisvec and their derivatives in dbasisdr, dbasisds, dbasisdt. implementation is in basis.cpp.

Material Properties

These are stored in class SRMaterial, see material.h: Isotropic materials have E, nu, rho, tref, alphax, alphay, alphaz, allowableStress. AllowableStress is optional.

Othotropic materials have struct cij, see material.h, while general anisotropic materials have class SrgenAnisoCij which contains a 6x6 stiffness matrix, which must be symmetric.

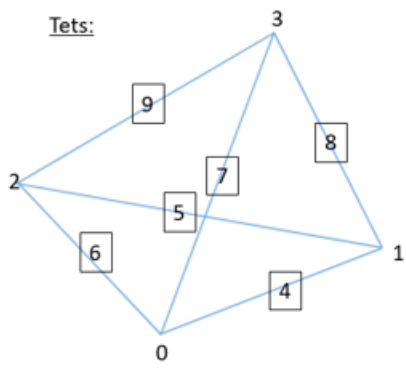
For all materials, call assignTempRho to set reference temperature, coefficients of thermal expansion, and density. If the material is isotropic, only alphax is used, alphay and alphaz can be set to the same as alphax or 0. assignOrtho(SRcij& cij);

For isotropic materials call assignIso(E,nu). For orthotropic, call assignOrtho(cij) . and for general orthotropic call assignGenAniso(SRgenAnisoCij& gcij). only the elements either above or below the diagonal need be specified, it will be made symmetric.

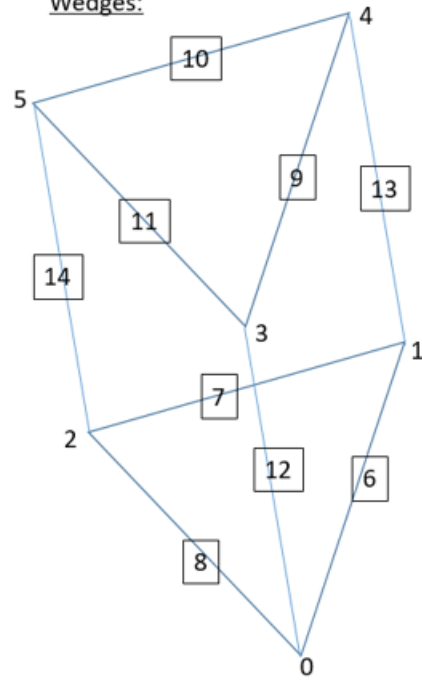
Element Node Numbering

The stressRefine element local node numbering convention is shown below for tets, wedges, and bricks. The quadratic midnodes are shown with a box around them, next to the edge they are in the middle of.

Tets:



Wedges:



Bricks:

