Homework Exercises for Chapter 9
Solutions

EXERCISE 9.1 (5 pts) If the tetrahedron is materially homogeneous, given the four corner locations specifies the fabrication properties uniquely. Nothing more is needed.

If the material is a layered composite and the element domain cuts through two or more layers, fabrication properties have to be specified. Typically these involve layer thicknesses and equations of layer interface planes. The implementation then becomes far more complex since the integral of $\mathbf{B}^T\,\mathbf{E}\,\mathbf{B}$ over the domain cannot be taken out, and subvolumes become tapered prisms. Such elements are rarely implemented in such geometric detail; pre-homogenization of material properties is more common.

EXERCISE 9.2 (20 pts) Given the corner Cartesian coordinate, compute the volume V from (9.3) and the entries of $\mathbf{J}^{-1}$, including a_i, b_i, c_i and V_{0i}, from (9.11) and (9.12). To get the altitude h_i replace the coordinates of the i^{th} corner in the LHS of the normalized face equation (9.16):

$$h_i = \frac{1}{S_i}(6V_{0i} + a_i\,x_i + b_i\,y_i + c_i\,z_i), \quad \text{in which} \quad S_i = \sqrt{a_i^2 + b_i^2 + c_i^2}. \tag{E9.2}$$

Finally, solve $V = \frac{1}{3} A_i\,h_i$ for A_i to get

$$A_i = \frac{V}{h_i} = \frac{V\,S_i}{6V_{0i} + a_i\,x_i + b_i\,y_i + c_i\,z_i}, \quad i = 1, 2, 3, 4. \tag{E9.3}$$

This scheme is implemented in module `Tet4FaceAreasViaAltitudes`, listed in Figure E9.1. It is referenced as `{ A1,A2,A3,A4 }=Tet4FaceAreasViaAltitudes[xyztet]`. The only argument is `xyztet`, which stores the $\{x, y, z\}$ coordinates of the corners as `{ { x1,y1,z1 },{ x2,y2,z2 },{ x3,y3,z3 },{ x4,y4,z4 } }`. The module returns the 4 face areas as list `{ A1,A2,A3,A4 }`. Note that the volume V is computed via (9.13) instead of taking the Jacobian determinant.

An alternative procedure uses a well known cross product property. Select two face sides as directed vectors emanating from any of its corners. For instance to do face 1 with corners $\{2, 3, 4\}$, pick $\mathbf{x}_{32}$ and $\mathbf{x}_{42}$; whence

$$A_1 = \tfrac{1}{2}||\mathbf{x}_{32} \times \mathbf{x}_{42}||_2. \tag{E9.4}$$

This scheme is implemented in module `Tet4FaceAreasViaXProduct`, listed in Figure E9.2. The argument and function return is exactly the same as for `Tet4FaceAreasViaAltitudes`. The module uses the built-in *Mathematica* function `Cross` to do cross products.

The script listed in Figure E9.3 symbolically tests both modules using a tetrahedron defined by the node coordinates

$$\texttt{\{\{0,0,0\},\{1,c,d\},\{d,1,c\},\{c,d,1\}\}}. \tag{E9.5}$$

in which c and d are arbitrary geometric dimensions. The output results are shown at the bottom of Figure E9.3. The results are verified to be identical. If $c = d = 0$ are zero, the areas of the unit reference tetrahedron are $A_1 = \frac{1}{2}, \sqrt{3}$, $A_2 = A_3 = A_4 = \frac{1}{2}$, which are readily verified by inspection.

Which scheme is better? Comparing the code in the two modules, evidently the cross product one is simpler and quicker. But there is an important difference. The altitudes scheme returns *signed* areas, since both V and h_i are signed quantities, and so is $3V/h_i$. On the other hand, (E9.4) returns the area magnitude, which is always positive. This can be important if one needs to check for negative areas as signal of input errors.

A third way of computing face areas is described in [808, p. 250]. The three area projections on the $\{x, y, z\}$ planes, given by 3×3 determinants, are combined by a well known formula of analytical geometry. For example, consider area A_4, and let its projections on the yz, zx and xy planes be denoted by A_{4x}, A_{4y} and A_{4z}, respectively. Expanding the determinants of [808] gives

$$A_{4x} = -a_4/2, \quad A_{4y} = -b_4/2, \quad A_{4z} = -c_4/2. \tag{E9.6}$$

The sum of the squares of the projected areas is the square of the face area: $A_4^2 = A_{4x}^2 + A_{4y}^2 + A_{4z}^2 = (a_4^2 + b_4^2 + c_4^2)/4 = S_4^2/4$, whence $A_4 = S_4/2$ and, in general

$$A_i = \tfrac{1}{2}\, S_i. \tag{E9.7}$$

This is the most compact and elegant approach, but it also loses the area signs along the way.

EXERCISE 9.3 (15 pts)

(a) According to the first of (9.19) the partial derivative of F_b with respect to x is

$$\frac{\partial F_b}{\partial x} = \frac{a_i}{6V}\frac{\partial F_b}{\partial \zeta_i} = \frac{1}{6V}\big(a_1\zeta_2\zeta_3\zeta_4 + a_2\zeta_1\zeta_3\zeta_4 + a_3\zeta_1\zeta_2\zeta_4 + a_4\zeta_1\zeta_2\zeta_3\big). \tag{E9.8}$$

But along any edge two tetrahedral coordinates vanish. For example, over the edge connecting 1 and 2, $\zeta_3 = \zeta_4 = 0$. Thus $\partial F_b/\partial x = 0$ identically over any edge. Likewise $\partial F_b/\partial y$ and $\partial F_b/\partial z$ vanish. (Since corners lie at the intersection of 3 edges, those derivatives also vanish there.)

(b) Using the integral formula (9.22) gives $\int_{\Omega^e} F_b\, d\Omega = (6V) \times 1! \times 1! \times 1! \times 1!/(1+1+1+1+3)! = (6V)/7! = V\,(6/5040)$, whence the volume integral is $V/840$.

EXERCISE 9.4 Not assigned.

EXERCISE 9.5 Not assigned.

EXERCISE 9.6 (20 pts) The total pressure force is $p\,A_i$, which is directed along the unit interior normal of the i^{th} face. These direction cosines are $\{\bar{a}_i = a_i/S_i, \bar{b}_i = b_i/S_i, \bar{c}_i/S_i\}$, which are computed as indicated in §9.1.7. The face area A_i can be computed by any of the methods outlined in the solution of Exercise 9.2. Using NbN lumping each corner node on that face gets 1/3 of the load, or $p\,A_i/3$. Specializing the results for face 1 gives

$$\mathbf{f}^e = \frac{1}{3} p\, A_1 [\, 0 \quad 0 \quad 0 \quad \bar{a}_1 \quad \bar{b}_1 \quad \bar{c}_1 \quad \bar{a}_1 \quad \bar{b}_1 \quad \bar{c}_1 \quad \bar{a}_1 \quad \bar{b}_1 \quad \bar{c}_1\,]^T. \tag{E9.9}$$

```
Tet4FaceAreasViaAltitudes[xyztet_]:=Module[{
   x1,y1,z1,x2,y2,z2,x3,y3,z3,x4,y4,z4,
   x12,x13,x14,x21,x23,x24,x31,x32,x34,x41,x42,x43,
   y12,y13,y14,y21,y23,y24,y31,y32,y34,y41,y42,y43,
   z12,z13,z14,z21,z23,z24,z31,z32,z34,z41,z42,z43,
   a1,b1,c1,a2,b2,c2,a3,b3,c3,a4,b4,c4,S1,S2,S3,S4,
   V,V1,V2,V3,V4,h1,h2,h3,h4,A1,A2,A3,A4},
  {{x1,y1,z1},{x2,y2,z2},{x3,y3,z3},{x4,y4,z4}}=xyztet;
   x12=x1-x2;x13=x1-x3;x14=x1-x4;x23=x2-x3;x24=x2-x4;x34=x3-x4;
   x21=-x12; x31=-x13; x32=-x23; x41=-x14; x42=-x24; x43=-x34;
   y12=y1-y2;y13=y1-y3;y14=y1-y4;y23=y2-y3;y24=y2-y4;y34=y3-y4;
   y21=-y12; y31=-y13; y32=-y23; y41=-y14; y42=-y24; y43=-y34;
   z12=z1-z2;z13=z1-z3;z14=z1-z4;z23=z2-z3;z24=z2-z4;z34=z3-z4;
   z21=-z12; z31=-z13; z32=-z23; z41=-z14; z42=-z24; z43=-z34;
   {a1,b1,c1}={y42*z32-y32*z42,x32*z42-x42*z32,x42*y32-x32*y42};
   {a2,b2,c2}={y31*z43-y34*z13,x43*z31-x13*z34,x31*y43-x34*y13};
   {a3,b3,c3}={y24*z14-y14*z24,x14*z24-x24*z14,x24*y14-x14*y24};
   {a4,b4,c4}={y13*z21-y12*z31,x21*z13-x31*z12,x13*y21-x12*y31};
   {S1,S2,S3,S4}=Sqrt[{a1^2+b1^2+c1^2,a2^2+b2^2+c2^2,
                       a3^2+b3^2+c3^2,a4^2+b4^2+c4^2}];
   V1=x2*(y3*z4-y4*z3)+x3*(y4*z2-y2*z4)+x4*(y2*z3-y3*z2);
   V2=x1*(y4*z3-y3*z4)+x3*(y1*z4-y4*z1)+x4*(y3*z1-y1*z3);
   V3=x1*(y2*z4-y4*z2)+x2*(y4*z1-y1*z4)+x4*(y1*z2-y2*z1);
   V4=x1*(y3*z2-y2*z3)+x2*(y1*z3-y3*z1)+x3*(y2*z1-y1*z2);
   V=(V1+V2+V3+V4)/6;
   JJinv={{V1,y42*z32-y32*z42,x32*z42-x42*z32,x42*y32-x32*y42},
          {V2,y31*z43-y34*z13,x43*z31-x13*z34,x31*y43-x34*y13},
          {V3,y24*z14-y14*z24,x14*z24-x24*z14,x24*y14-x14*y24},
          {V4,y13*z21-y12*z31,x21*z13-x31*z12,x13*y21-x12*y31}};
   h1=(a1*x1+b1*y1+c1*z1+V1)/S1; h2=(a2*x2+b2*y2+c2*z2+V2)/S2;
   h3=(a3*x3+b3*y3+c3*z3+V3)/S3; h4=(a4*x4+b4*y4+c4*z4+V4)/S4;
   {A1,A2,A3,A4}=3*{V/h1,V/h2,V/h3,V/h4};
   Return[{A1,A2,A3,A4}]];
```

FIGURE E9.1. *Mathematica* module to compute face areas via altitudes.

```
Tet4FaceAreasViaXProduct[xyztet_]:=Module[{
   x1,y1,z1,x2,y2,z2,x3,y3,z3,x4,y4,z4,
   x12,x13,x14,x21,x23,x24,x31,x32,x34,x41,x42,x43,
   y12,y13,y14,y21,y23,y24,y31,y32,y34,y41,y42,y43,
   z12,z13,z14,z21,z23,z24,z31,z32,z34,z41,z42,z43,
   v1,v2,v3,v4,A1,A2,A3,A4},
  {{x1,y1,z1},{x2,y2,z2},{x3,y3,z3},{x4,y4,z4}}=xyztet;
   x12=x1-x2;x13=x1-x3;x14=x1-x4;x23=x2-x3;x24=x2-x4;x34=x3-x4;
   x21=-x12; x31=-x13; x32=-x23; x41=-x14; x42=-x24; x43=-x34;
   y12=y1-y2;y13=y1-y3;y14=y1-y4;y23=y2-y3;y24=y2-y4;y34=y3-y4;
   y21=-y12; y31=-y13; y32=-y23; y41=-y14; y42=-y24; y43=-y34;
   z12=z1-z2;z13=z1-z3;z14=z1-z4;z23=z2-z3;z24=z2-z4;z34=z3-z4;
   z21=-z12; z31=-z13; z32=-z23; z41=-z14; z42=-z24; z43=-z34;
   v1=Simplify[Cross[{x32,y32,z32},{x42,y42,z42}]];
   A1=Sqrt[v1[[1]]^2+v1[[2]]^2+v1[[3]]^2]/2;
   v2=Simplify[Cross[{x43,y43,z43},{x13,y13,z13}]];
   A2=Sqrt[v2[[1]]^2+v2[[2]]^2+v2[[3]]^2]/2;
   v3=Simplify[Cross[{x14,y14,z14},{x24,y24,z24}]];
   A3=Sqrt[v3[[1]]^2+v3[[2]]^2+v3[[3]]^2]/2;
   v4=Simplify[Cross[{x21,y21,z21},{x31,y31,z31}]];
   A4=Sqrt[v4[[1]]^2+v4[[2]]^2+v4[[3]]^2]/2;
   Return[{A1,A2,A3,A4}]];
```

FIGURE E9.2. *Mathematica* script to compute face areas via cross products.

```
ClearAll[c,d];  xyztet={{0,0,0},{1,c,d},{d,1,c},{c,d,1}};
ar=Tet4FaceAreasViaAltitudes[xyztet];  ar=Simplify[ar,c∈Reals&&d∈Reals];
Print["f-areas via altitudes=",ar];
arX=Tet4FaceAreasViaXProduct[xyztet];  arX=Simplify[arX,c∈Reals&&d∈Reals];
Print["f-areas via Xproduct=",arX];  Print["check=",Simplify[arX-ar]];
```

f-areas via altitudes= $\{\frac{1}{2}\sqrt{3}(1+c^2-d+d^2-c(1+d)),\ \frac{1}{2}\sqrt{1+c^4+c^2(-1+d)^2+d^2+d^4-2\,c\,d\,(1+d)},$
$\frac{1}{2}\sqrt{1+c^4+c^2(-1+d)^2+d^2+d^4-2\,c\,d\,(1+d)},\ \frac{1}{2}\sqrt{1+c^4+c^2(-1+d)^2+d^2+d^4-2\,c\,d\,(1+d)}\}$

f-areas via x-product= $\{\frac{1}{2}\sqrt{3}(1+c^2-d+d^2-c(1+d)),\ \frac{1}{2}\sqrt{1+c^4+c^2(-1+d)^2+d^2+d^4-2\,c\,d\,(1+d)},$
$\frac{1}{2}\sqrt{1+c^4+c^2(-1+d)^2+d^2+d^4-2\,c\,d\,(1+d)},\ \frac{1}{2}\sqrt{1+c^4+c^2(-1+d)^2+d^2+d^4-2\,c\,d\,(1+d)}\}$

check={0,0,0,0}

FIGURE E9.3. *Mathematica* script to test the modules of Figures E9.1 and E9.2