Introduction

This notebook performs the math to volume-integrate basic monomials up to second order over intervals in 1d; quadrilaterals and triangles in 2d; and hexagons, wedges and tetrahedrons in 3d. (I left out pyramids. Sorry. This should be done later.) The results appear the the Portage file src/support/operator.h .

Basic definitions for bases and shape functions - only quadratic, which includes linear and constant

```
basis1[x_List] := {1, x[[1]], x[[1]]^2/2};
basis1Lin[x_List] := {1, x[[1]]};
basis2[x_List] := {1, x[[1]], x[[2]], x[[1]]^2/2, x[[1]] * x[[2]], x[[2]]^2/2};
basis2Lin[x_List] := {1, x[[1]], x[[2]]};
basis3[x_List] := {1, x[[1]], x[[2]], x[[3]]};
basis3Quad[x_List] := {1, x[[1]], x[[2]], x[[3]] , x[[1]]^2/2,
   x[[1]] * x[[2]], x[[1]] * x[[3]], x[[2]]^2/2, x[[2]] * x[[3]], x[[3]]^2/2};
```

```
transFactor[b_, q_List] :=
  Block[{bq, dim, bsub, c, cl, bqc, bbqc, bb, bblist, cof, bbsub},
   bq = b[q];
   If[bq[[1]] =!= 1, Print["first element of basis must be 1"]; Return[]];
   dim = Length[bq];
   bsub = Reverse[Table[ bq[[i]] → bb[i], {i, 2, dim}]];
   cl = c /@ Range[0, Length[q] - 1];
   bqc = Expand[ b[q + cl]];
   bbqc = bqc /. bsub;
   bblist = bb /@ Range[dim];
   cof = Table[ Coefficient[bbqc[[i]], bblist[[j]]], {i, dim}, {j, dim}];
   Do[ cof[[i, 1]] = Flatten[CoefficientList[bbqc[[i]], bblist]][[1]], {i, dim}];
   bbsub = Reverse[Table[ bb[i] -> bq[[i]], {i, 1, dim}]];
   check = Table[ Simplify[(cof[[i]].bblist /. bbsub ) - bqc[[i]]], {i, dim}];
   If[Union[check] === {0},
    cof,
    Print["resulting basis is not the same as the original"];
    Null
   ]
  ];
```

```
shapequad[x_List] :=
  Block[{},
   Assert[ Length[x] = 2 ];
   Return[ {(1 - x[[1]]) (1 - x[[2]]),
     x[[1]] (1 - x[[2]]),
     x[[1]] x[[2]],
     (1 - x[[1]]) x[[2]]}];
  ];
shapetri[x_List] :=
  Block[{},
   Assert[ Length[x] = 2 ];
   Return[ {1 - x[[1]] - x[[2]], x[[1]] , x[[2]]}];
  ];
shapehex[x_List] :=
  Block[{},
   Assert[ Length[x] = 3 ];
   Return[ {(1 - x[[1]]) (1 - x[[2]]) (1 - x[[3]]),
     x[[1]] (1 - x[[2]]) (1 - x[[3]]),
     x[[1]] x[[2]] (1 - x[[3]]),
     (1 - x[[1]]) x[[2]] (1 - x[[3]]),
     (1 - x[[1]]) (1 - x[[2]]) x[[3]],
     x[[1]] (1 - x[[2]]) x[[3]],
     x[[1]] x[[2]] x[[3]],
     (1 - x[[1]]) x[[2]] x[[3]]}];
  ];
shapewedge[x_List] :=
  Block[{},
   Assert[ Length[x] = 3 ];
   Return[ {
     (1 - x[[1]] - x[[2]]) (1 - x[[3]]), x[[1]] (1 - x[[3]]) , x[[2]] (1 - x[[3]]),
     (1 - x[[1]] - x[[2]]) x[[3]], x[[1]] x[[3]], x[[2]] x[[3]]}];
  ];
shapetet[x_List] :=
  Block[{},
   Assert[ Length[x] = 3 ];
   Return[ {1 - x[[1]] - x[[2]] - x[[3]], x[[1]] , x[[2]], x[[3]]}];
  ];
```

Check

```
shapetri /@ {{0, 0}, {1, 0}, {0, 1}} - IdentityMatrix[3] // Flatten // Union
```

```
{0}
```

```
shapequad /@ {{0, 0}, {1, 0}, {1, 1}, {0, 1}} - IdentityMatrix[4] // Flatten // Union
```

```
{0}
```

```
shapetet /@ {{0, 0, 0}, {1, 0, 0}, {0, 1, 0}, {0, 0, 1}} - IdentityMatrix[4] // Flatten //
 Union
```

```
{0}
```

```
shapehex /@ {{0, 0, 0}, {1, 0, 0}, {1, 1, 0}, {0, 1, 0}, {0, 0, 1},
      {1, 0, 1}, {1, 1, 1}, {0, 1, 1}} - IdentityMatrix[8] // Flatten // Union
```

```
{0}
```

```
shapewedge /@ {{0, 0, 0}, {1, 0, 0}, {0, 1, 0}, {0, 0, 1}, {1, 0, 1}, {0, 1, 1}} -
   IdentityMatrix[6] // Flatten // Union
```

```
{0}
```

Definition of integration limits and symbolic point sets for each element

```
qvec1 =  {q0}; qvec2 = {q0, q1}; qvec3 = {q0, q1, q2};
qv1int = {#, 0, 1} & /@ qvec1; qv2int = {#, 0, 1} & /@ qvec2;
qv3int = {#, 0, 1} & /@ qvec3;
qv2intTri = {{q0, 0, 1 - q1}, {q1, 0, 1}};
qv3intWedge = {{q0, 0, 1 - q1}, {q1, 0, 1}, {q2, 0, 1}};
qv3intTet = {{q0, 0, 1 - q1 - q2}, {q1, 0, 1 - q2}, {q2, 0, 1}};
ptsInterval = Table[p[i][j], {i, 0, 1}, {j, 0, 0}];
ptsTri = Table[p[i][j], {i, 0, 2}, {j, 0, 1}];
ptsQuad = Table[p[i][j], {i, 0, 3}, {j, 0, 1}];
ptsTet = Table[p[i][j], {i, 0, 3}, {j, 0, 2}];
ptsWedg = Table[p[i][j], {i, 0, 5}, {j, 0, 2}];
ptsHex = Table[p[i][j], {i, 0, 7}, {j, 0, 2}];
prules[pts_] := p[i_][j_] → pts[[i + 1]][[j + 1]];
```

Definition of coordinate transformation from reference element to arbitrary element and Jacobian

```
coordmap[pts_List, shape_, q_List] := Sum[pts[[i]] shape[q][[i]], {i, Length[pts]}];
jacobian[cmap_, q_List] :=
  Table[ D[cmap[[i]], q[[j]]], {i, Length[cmap]}, {j, Length[q]}];
```

Check

Integrate the bases on arbitrary elements

```
basis2IntTri = Block[{xvec, jac, integrand,
    coeff, orders, integrandSimple, coffsubs, coffsubs2, integral},
   xvec = coordmap[ptsTri, shapetri, qvec2];
   jac = Det@jacobian[xvec, qvec2];
   integrand = Collect[basis2[xvec] jac, qvec2];
   coeff = CoefficientList[#, qvec2] & /@ integrand;
   orders = Dimensions /@ coeff;
   integrandSimple = Table[Sum[c[k, i - 1, j - 1] qvec2[[1]] ^ (i - 1) qvec2[[2]] ^ (j - 1),
      {i, orders[[k, 1]]}, {j, orders[[k, 2]]}], {k, Length[integrand]}];
  coffsubs = Flatten /@ Table[c[k, i - 1, j - 1] → coeff[[k, i, j]],
      {k, Length[integrand]}, {i, orders[[k, 1]]}, {j, orders[[k, 2]]}];
   coffsubs2 = Map[Simplify, coffsubs, {3}];
   Print[integrand - Table[
       integrandSimple[[i]] /. coffsubs2[[i]], {i, Length[integrand]}] // Simplify];
   integralSimple = Integrate[Integrate[integrandSimple, qv2intTri[[1]]],
     qv2intTri[[2]]];
   integral = Table[integralSimple[[i]] /. coffsubs2[[i]],
      {i, Length[integralSimple]}] // Simplify
  ] //
  Simplify
```

{0, 0, 0, 0, 0, 0}

$$\Big\{\frac{1}{2}\left(-p[1][1]\,p[2][0]+p[0][1]\,(-p[1][0]+p[2][0])+p[0][0]\,(p[1][1]-p[2][1])+p[1][0]\,p[2][1]\right),$$

$$\frac{1}{6}\,(p[0][0]+p[1][0]+p[2][0])\,\left(-p[1][1]\,p[2][0]+p[0][1]\,(-p[1][0]+p[2][0])+p[0][0]\,(p[1][1]-p[2][1])+p[1][0]\,p[2][1]\right),$$

$$-\frac{1}{6}\,(p[0][1]+p[1][1]+p[2][1])\,\left(p[0][1]\,(p[1][0]-p[2][0])+p[1][1]\,p[2][0]-p[1][0]\,p[2][1]+p[0][0]\,(-p[1][1]+p[2][1])\right),$$

$$\frac{1}{24}\left(p[0][0]^2+p[1][0]^2+p[1][0]\,p[2][0]+p[2][0]^2+p[0][0]\,(p[1][0]+p[2][0])\right)\left(-p[1][1]\,p[2][0]+p[0][1]\,(-p[1][0]+p[2][0])+p[0][0]\,(p[1][1]-p[2][1])+p[1][0]\,p[2][1]\right),$$

$$\frac{1}{24}\Big(-2\,p[1][0]\,p[1][1]^2\,p[2][0]-p[1][1]^2\,p[2][0]^2+p[0][1]^2\left(-p[1][0]^2+p[2][0]^2\right)+2\,p[1][0]^2\,p[1][1]\,p[2][1]-2\,p[1][1]\,p[2][0]^2\,p[2][1]+p[1][0]^2\,p[2][1]^2+2\,p[1][0]\,p[2][0]\,p[2][1]^2+p[0][0]^2\,(p[1][1]-p[2][1])\,(2\,p[0][1]+p[1][1]+p[2][1])+p[0][1]\left(-2\,p[1][0]^2\,p[1][1]+2\,p[2][0]^2\,p[2][1]\right)-2\,p[0][0]\left(-p[1][0]\,p[1][1]^2+p[0][1]^2\,(p[1][0]-p[2][0])+p[2][0]\,p[2][1]^2\right)\Big),$$

$$-\frac{1}{24}\left(p[0][1]\,(p[1][0]-p[2][0])+p[1][1]\,p[2][0]-p[1][0]\,p[2][1]+p[0][0]\,(-p[1][1]+p[2][1])\right)\left(p[0][1]^2+p[1][1]^2+p[1][1]\,p[2][1]+p[2][1]^2+p[0][1]\,(p[1][1]+p[2][1])\right)\Big\}$$

```
basis2IntTri =
  {1/2 (-p[1][1] p[2][0] + p[0][1] (-p[1][0] + p[2][0]) + p[0][0] (p[1][1] - p[2][1]) +
      p[1][0] p[2][1]), 1/6 (p[0][0] + p[1][0] + p[2][0]) (-p[1][1] p[2][0] +
      p[0][1] (-p[1][0] + p[2][0]) + p[0][0] (p[1][1] - p[2][1]) + p[1][0] p[2][1]),
   -1/6 (p[0][1] + p[1][1] + p[2][1]) (p[0][1] (p[1][0] - p[2][0]) +
      p[1][1] p[2][0] - p[1][0] p[2][1] + p[0][0] (-p[1][1] + p[2][1])),
   1/24 (p[0][0]^2 + p[1][0]^2 + p[1][0] p[2][0] + p[2][0]^2 + p[0][0] (p[1][0] + p[2][0]))
    (-p[1][1] p[2][0] + p[0][1] (-p[1][0] + p[2][0]) +
      p[0][0] (p[1][1] - p[2][1]) + p[1][0] p[2][1]),
   1/24 (-2 p[1][0] p[1][1]^2 p[2][0] - p[1][1]^2 p[2][0]^2 + p[0][1]^2 (-p[1][0]^2 + p[2][0]^2) +
      2 p[1][0]^2 p[1][1] p[2][1] - 2 p[1][1] p[2][0]^2 p[2][1] + p[1][0]^2 p[2][1]^2 +
      2 p[1][0] p[2][0] p[2][1]^2 + p[0][0]^2 (p[1][1] - p[2][1])
        (2 p[0][1] + p[1][1] + p[2][1]) + p[0][1] (-2 p[1][0]^2 p[1][1] + 2 p[2][0]^2 p[2][1]) -
      2 p[0][0] (-p[1][0] p[1][1]^2 + p[0][1]^2 (p[1][0] - p[2][0]) + p[2][0] p[2][1]^2)),
   -1/24 (p[0][1] (p[1][0] - p[2][0]) + p[1][1] p[2][0] - p[1][0] p[2][1] +
      p[0][0] (-p[1][1] + p[2][1]))
    (p[0][1]^2 + p[1][1]^2 + p[1][1] p[2][1] + p[2][1]^2 + p[0][1] (p[1][1] + p[2][1]))};

basis2IntQuad = Block[{xvec, jac, integrand,
   coeff, orders, integrandSimple, coffsubs, coffsubs2, integral},
  xvec = coordmap[ptsQuad, shapequad, qvec2];
  jac = Det@jacobian[ xvec, qvec2];
  integrand = Collect[basis2[xvec] jac, qvec2];
  coeff = CoefficientList[#, qvec2] & /@ integrand;
  orders = Dimensions /@ coeff;
  integrandSimple = Table[Sum[ c[k, i - 1, j - 1] qvec2[[1]] ^ (i - 1) qvec2[[2]] ^ (j - 1),
      {i, orders[[k, 1]]}, {j, orders[[k, 2]]}], {k, Length[integrand]}];
  coffsubs = Flatten /@ Table[ c[k, i - 1, j - 1] → coeff[[k, i, j]],
      {k, Length[integrand]}, {i, orders[[k, 1]]}, {j, orders[[k, 2]]}];
  coffsubs2 = Map[Simplify, coffsubs, {3}];
  Print[
   integrand - Table[integrandSimple[[i]] /. coffsubs2[[i]], {i, Length[integrand]}] //
    Simplify];
  integralSimple = Integrate[integrandSimple, qv2int[[1]], qv2int[[2]]];
  integral = Table[integralSimple[[i]] /. coffsubs2[[i]],
      {i, Length[integralSimple]}] // Simplify
 ]
{0, 0, 0, 0, 0, 0}
```

```
basis2IntQuad =
  {1/2 (-p[1][1] p[2][0] + p[1][0] p[2][1] - p[2][1] p[3][0] + p[0][1] (-p[1][0] + p[3][0]) +
      p[0][0] (p[1][1] - p[3][1]) + p[2][0] p[3][1]),
   1/6 (-p[1][0] p[1][1] p[2][0] - p[1][1] p[2][0]^2 + p[1][0]^2 p[2][1] +
      p[1][0] p[2][0] p[2][1] - p[2][0] p[2][1] p[3][0] -
      p[2][1] p[3][0]^2 + p[0][1] (-p[1][0]^2 + p[3][0]^2) +
      p[0][0]^2 (p[1][1] - p[3][1]) + p[2][0]^2 p[3][1] + p[2][0] p[3][0] p[3][1] +
      p[0][0] (p[1][0] p[1][1] + p[0][1] (-p[1][0] + p[3][0]) - p[3][0] p[3][1])),
   1/6 (-p[1][1]^2 p[2][0] + p[1][0] p[1][1] p[2][1] - p[1][1] p[2][0] p[2][1] +
      p[1][0] p[2][1]^2 - p[2][1]^2 p[3][0] + p[0][1]^2 (-p[1][0] + p[3][0]) +
      p[2][0] p[2][1] p[3][1] - p[2][1] p[3][0] p[3][1] + p[2][0] p[3][1]^2 +
      p[0][1] (-p[1][0] p[1][1] + p[0][0] (p[1][1] - p[3][1]) + p[3][0] p[3][1]) +
      p[0][0] (p[1][1]^2 - p[3][1]^2)),
   1/24 (-p[1][0]^2 p[1][1] p[2][0] - p[1][0] p[1][1] p[2][0]^2 - p[1][1] p[2][0]^3 +
      p[1][0]^3 p[2][1] + p[1][0]^2 p[2][0] p[2][1] + p[1][0] p[2][0]^2 p[2][1] -
      p[2][0]^2 p[2][1] p[3][0] - p[2][0] p[2][1] p[3][0]^2 - p[2][1] p[3][0]^3 +
      p[0][1] (-p[1][0]^3 + p[3][0]^3) + p[0][0]^3 (p[1][1] - p[3][1]) +
      p[2][0]^3 p[3][1] + p[2][0]^2 p[3][0] p[3][1] + p[2][0] p[3][0]^2 p[3][1] +
      p[0][0]^2 (p[1][0] p[1][1] + p[0][1] (-p[1][0] + p[3][0]) - p[3][0] p[3][1]) +
      p[0][0] (p[1][0]^2 p[1][1] + p[0][1] (-p[1][0]^2 + p[3][0]^2) - p[3][0]^2 p[3][1])),
   1/24 (-2 p[1][0] p[1][1]^2 p[2][0] - p[1][1]^2 p[2][0]^2 + 2 p[1][0]^2 p[1][1] p[2][1] -
      2 p[1][1] p[2][0]^2 p[2][1] + p[1][0]^2 p[2][1]^2 + 2 p[1][0] p[2][0] p[2][1]^2 -
      2 p[2][0] p[2][1]^2 p[3][0] - p[2][1]^2 p[3][0]^2 + p[0][1]^2 (-p[1][0]^2 + p[3][0]^2) +
      2 p[2][0]^2 p[2][1] p[3][1] - 2 p[2][1] p[3][0]^2 p[3][1] + p[2][0]^2 p[3][1]^2 +
      2 p[2][0] p[3][0] p[3][1]^2 + p[0][0]^2 (p[1][1] - p[3][1])
        (2 p[0][1] + p[1][1] + p[3][1]) + p[0][1] (-2 p[1][0]^2 p[1][1] + 2 p[3][0]^2 p[3][1]) -
      2 p[0][0] (-p[1][0] p[1][1]^2 + p[0][1]^2 (p[1][0] - p[3][0]) + p[3][0] p[3][1]^2)),
   1/24 (-p[1][1]^3 p[2][0] + p[1][0] p[1][1]^2 p[2][1] - p[1][1]^2 p[2][0] p[2][1] + p[1][0]
        p[1][1] p[2][1]^2 - p[1][1] p[2][0] p[2][1]^2 + p[1][0] p[2][1]^3 - p[2][1]^3 p[3][0] +
      p[0][1]^3 (-p[1][0] + p[3][0]) + p[2][0] p[2][1]^2 p[3][1] - p[2][1]^2 p[3][0] p[3][1] +
      p[2][0] p[2][1] p[3][1]^2 - p[2][1] p[3][0] p[3][1]^2 + p[2][0] p[3][1]^3 +
      p[0][1]^2 (-p[1][0] p[1][1] + p[0][0] (p[1][1] - p[3][1]) + p[3][0] p[3][1]) +
      p[0][0] (p[1][1]^3 - p[3][1]^3) +
      p[0][1] (-p[1][0] p[1][1]^2 + p[3][0] p[3][1]^2 + p[0][0] (p[1][1]^2 - p[3][1]^2)))};
```

```
basis3IntHex = Block[{xvec, jac, integrand, coeff, orders,
    integrandSimple, coffsubs, coffsubs2, basIntegral, integral},
  xvec = coordmap[ptsHex, shapehex, qvec3];
  jac = Det@jacobian[ xvec, qvec3];
  integrand = Collect[basis3[xvec] jac, qvec3];
  coeff = CoefficientList[#, qvec3] & /@ integrand;
  orders = Dimensions /@ coeff;
  integrandSimple = Table[Sum[ c[m, i - 1, j - 1, k - 1]
       qvec3[[1]] ^ (i - 1) qvec3[[2]] ^ (j - 1) qvec3[[3]] ^ (k - 1), {i, orders[[m, 1]]},
     {j, orders[[m, 2]]}, {k, orders[[m, 3]]}], {m, Length[integrand]}];
  coffsubs = Flatten /@ Table[ c[m, i - 1, j - 1, k - 1] → coeff[[m, i, j, k]],
     {m, Length[integrand]}, {i, orders[[m, 1]]},
     {j, orders[[m, 2]]}, {k, orders[[m, 3]]}];
  Print[integrand - Table[integrandSimple[[i]] /. coffsubs[[i]],
       {i, Length[integrand]}] // Simplify];
  baseIntegral[i_, j_, k_] = Integrate[qvec3[[1]] ^ i qvec3[[2]] ^ j qvec3[[3]] ^ k,
    qv3int[[1]], qv3int[[2]], qv3int[[3]]];
  integral = Table[Sum[ c[m, i - 1, j - 1, k - 1] baseIntegral[i - 1, j - 1, k - 1],
       {i, orders[[m, 1]]}, {j, orders[[m, 2]]}, {k, orders[[m, 3]]}] /.
      coffsubs[[m]], {m, Length[integrand]}] // Simplify
 ]

{0, 0, 0, 0}
```

```
{1/12 (p[0][0] p[1][2] p[2][1] - p[0][0] p[1][1] p[2][2] +
    p[1][2] p[2][1] p[3][0] - p[1][1] p[2][2] p[3][0] + p[0][0] p[1][2] p[3][1] -
    p[1][2] p[2][0] p[3][1] + p[0][0] p[2][2] p[3][1] + p[1][0] p[2][2] p[3][1] -
    p[0][0] p[1][1] p[3][2] + OutputSizeLimit`Skeleton[137] +
    p[4][0] p[5][1] p[7][2] + p[2][1] p[6][0] p[7][2] + p[3][1] p[6][0] p[7][2] -
    p[4][1] p[6][0] p[7][2] - p[5][1] p[6][0] p[7][2] - p[2][0] p[6][1] p[7][2] -
    p[3][0] p[6][1] p[7][2] + p[4][0] p[6][1] p[7][2] + p[5][0] p[6][1] p[7][2] +
    p[0][1] (-p[2][2] p[3][0] + p[2][0] p[3][2] - p[3][2] p[4][0] + p[3][0] p[4][2] -
       p[4][2] p[5][0] + p[1][2] (-p[2][0] - p[3][0] + p[4][0] + p[5][0]) +
       p[1][0] (p[2][2] + p[3][2] - p[4][2] - p[5][2]) + p[4][0] p[5][2] -
       p[3][2] p[7][0] + p[4][2] p[7][0] + p[3][0] p[7][2] - p[4][0] p[7][2])),
 1/72 (OutputSizeLimit`Skeleton[1]), 1/72 OutputSizeLimit`Skeleton[1],
 1/72
  (OutputSizeLimit`Skeleton[506] + OutputSizeLimit`Skeleton[1])}
```

large output | show less | show more | show all | set size limit...

```
basis3IntWedge = Block[{xvec, jac, integrand, coeff, orders,
    integrandSimple, coffsubs, coffsubs2, basIntegral, integral},
   xvec = coordmap[ptsWedg, shapewedge, qvec3];
   jac = Det@jacobian[ xvec, qvec3];
   integrand = Collect[basis3[xvec] jac, qvec3];
   coeff = CoefficientList[#, qvec3] & /@ integrand;
   orders = Dimensions /@ coeff;
   integrandSimple = Table[Sum[ c[m, i - 1, j - 1, k - 1]
       qvec3[[1]] ^ (i - 1) qvec3[[2]] ^ (j - 1) qvec3[[3]] ^ (k - 1), {i, orders[[m, 1]]},
      {j, orders[[m, 2]]}, {k, orders[[m, 3]]}], {m, Length[integrand]}];
   coffsubs = Flatten /@ Table[ c[m, i - 1, j - 1, k - 1] → coeff[[m, i, j, k]],
      {m, Length[integrand]}, {i, orders[[m, 1]]},
      {j, orders[[m, 2]]}, {k, orders[[m, 3]]}];
   Print[integrand - Table[integrandSimple[[i]] /. coffsubs[[i]],
       {i, Length[integrand]}] // Simplify];
   baseIntegral[i_, j_, k_] = Integrate[qvec3[[1]] ^ i qvec3[[2]] ^ j qvec3[[3]] ^ k,
     qv3intWedge[[3]], qv3intWedge[[2]], qv3intWedge[[1]]];
   integral = Table[Sum[ c[m, i - 1, j - 1, k - 1] baseIntegral[i - 1, j - 1, k - 1],
        {i, orders[[m, 1]]}, {j, orders[[m, 2]]}, {k, orders[[m, 3]]}] /.
       coffsubs[[m]], {m, Length[integrand]}] // Simplify
  ]
```

```
{0, 0, 0, 0}
```

```
{1/12 (2 p[0][0] p[1][2] p[2][1] - 2 p[0][0] p[1][1] p[2][2] -
    p[0][0] p[1][2] p[3][1] + p[0][0] p[2][2] p[3][1] + p[0][0] p[1][1] p[3][2] -
    p[0][0] p[2][1] p[3][2] - p[1][2] p[2][1] p[4][0] + p[1][1] p[2][2] p[4][0] +
    p[1][2] p[3][1] p[4][0] - p[1][1] p[3][2] p[4][0] - p[0][0] p[1][2] p[4][1] +
    ··· 54 ··· + p[1][0] p[2][1] p[5][2] - p[2][1] p[3][0] p[5][2] +
    p[0][0] p[3][1] p[5][2] + p[2][0] p[3][1] p[5][2] + p[1][1] p[4][0] p[5][2] +
    p[2][1] p[4][0] p[5][2] - 2 p[3][1] p[4][0] p[5][2] -
    p[1][0] p[4][1] p[5][2] - p[2][0] p[4][1] p[5][2] + 2 p[3][0] p[4][1] p[5][2] +
    p[0][1] (-p[2][2] p[3][0] + p[2][0] p[3][2] - p[3][2] p[4][0] + p[1][2]
         (-2 p[2][0] + p[3][0] + p[4][0]) + p[1][0] (2 p[2][2] - p[3][2] - p[4][2]) +
       p[3][0] p[4][2] - p[2][2] p[5][0] + p[3][2] p[5][0] + p[2][0] p[5][2] -
       p[3][0] p[5][2])), ··· 1 ···, ··· 1 ···, 1/72 (··· 1 ···)}
```

large output | show less | show more | show all | set size limit...

```
basis3IntTet = Block[{xvec, jac, integrand, coeff, orders,
    integrandSimple, coffsubs, coffsubs2, basIntegral, integral},
  xvec = coordmap[ptsTet, shapetet, qvec3];
  jac = Det@jacobian[ xvec, qvec3];
  integrand = Collect[basis3Quad[xvec] jac, qvec3];
  coeff = CoefficientList[#, qvec3] & /@ integrand;
  orders = Dimensions /@ coeff;
  integrandSimple = Table[Sum[ c[m, i - 1, j - 1, k - 1]
       qvec3[[1]] ^ (i - 1) qvec3[[2]] ^ (j - 1) qvec3[[3]] ^ (k - 1), {i, orders[[m, 1]]},
      {j, orders[[m, 2]]}, {k, orders[[m, 3]]}], {m, Length[integrand]}];
  coffsubs = Flatten /@ Table[ c[m, i - 1, j - 1, k - 1] → coeff[[m, i, j, k]],
      {m, Length[integrand]}, {i, orders[[m, 1]]},
      {j, orders[[m, 2]]}, {k, orders[[m, 3]]}];
  Print[integrand - Table[integrandSimple[[i]] /. coffsubs[[i]],
       {i, Length[integrand]}] // Simplify];
  baseIntegral[i_, j_, k_] = Integrate[qvec3[[1]] ^ i qvec3[[2]] ^ j qvec3[[3]] ^ k,
    qv3intTet[[3]], qv3intTet[[2]], qv3intTet[[1]]];
  integral = Table[Sum[ c[m, i - 1, j - 1, k - 1] baseIntegral[i - 1, j - 1, k - 1],
        {i, orders[[m, 1]]}, {j, orders[[m, 2]]}, {k, orders[[m, 3]]}] /.
       coffsubs[[m]], {m, Length[integrand]}] // Simplify
 ]
```

```
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0}
```

$$\Big\{\frac{1}{6}\,(p[0][0]\,p[1][2]\,p[2][1] - p[0][0]\,p[1][1]\,p[2][2] - p[1][2]\,p[2][1]\,p[3][0] + p[1][1]\,p[2][2]\,p[3][0] - p[0][0]\,p[1][2]\,p[3][1] + p[1][2]\,p[2][0]\,p[3][1] + p[0][0]\,p[2][2]\,p[3][1] - p[1][0]\,p[2][2]\,p[3][1] + p[0][2]\,(p[1][1]\,(p[2][0] - p[3][0]) + p[2][1]\,p[3][0] - p[2][0]\,p[3][1] + p[1][0]\,(-p[2][1] + p[3][1])) + p[0][0]\,p[1][1]\,p[3][2] - p[1][1]\,p[2][0]\,p[3][2] - p[0][0]\,p[2][1]\,p[3][2] + p[1][0]\,p[2][1]\,p[3][2] + p[0][1]\,(-p[2][2]\,p[3][0] + p[1][2]\,(-p[2][0] + p[3][0]) + p[1][0]\,(p[2][2] - p[3][2]) + p[2][0]\,p[3][2])),$$

$$\frac{1}{24}\,(p[0][0] + p[1][0] + p[2][0] + p[3][0])\,(p[0][0]\,p[1][2]\,p[2][1] - p[0][0]\,p[1][1]\,p[2][2] - p[1][2]\,p[2][1]\,p[3][0] + p[1][1]\,p[2][2]\,p[3][0] - p[0][0]\,p[1][2]\,p[3][1] + p[1][2]\,p[2][0]\,p[3][1] + p[0][0]\,p[2][2]\,p[3][1] - p[1][0]\,p[2][2]\,p[3][1] + p[0][2]\,(p[1][1]\,(p[2][0] - p[3][0]) + p[2][1]\,p[3][0] - p[2][0]\,p[3][1] + p[1][0]\,(-p[2][1] + p[3][1])) + p[0][0]\,p[1][1]\,p[3][2] - p[1][1]\,p[2][0]\,p[3][2] - p[0][0]\,p[2][1]\,p[3][2] + p[1][0]\,p[2][1]\,p[3][2] + p[0][1]\,(-p[2][2]\,p[3][0] + p[1][2]\,(-p[2][0] + p[3][0]) + p[1][0]\,(p[2][2] - p[3][2]) + p[2][0]\,p[3][2])),$$

$$\frac{1}{24}\,(p[0][1] + p[1][1] + p[2][1] + p[3][1])\,(p[0][0]\,p[1][2]\,p[2][1] - p[0][0]\,p[1][1]\,p[2][2] - p[1][2]\,p[2][1]\,p[3][0] + p[1][1]\,p[2][2]\,p[3][0] - p[0][0]\,p[1][2]\,p[3][1] + p[1][2]\,p[2][0]\,p[3][1] + p[0][0]\,p[2][2]\,p[3][1] - p[1][0]\,p[2][2]\,p[3][1] + p[0][2]\,(p[1][1]\,(p[2][0] - p[3][0]) + p[2][1]\,p[3][0] - p[2][0]\,p[3][1] + p[1][0]\,(-p[2][1] + p[3][1])) +$$

```
    p[0][0] p[1][1] p[3][2] - p[1][1] p[2][0] p[3][2] - p[0][0] p[2][1] p[3][2] +
    p[1][0] p[2][1] p[3][2] + p[0][1] (-p[2][2] p[3][0] +
       p[1][2] (-p[2][0] + p[3][0]) + p[1][0] (p[2][2] - p[3][2]) + p[2][0] p[3][2])),
 1/24 (p[0][2] + p[1][2] + p[2][2] + p[3][2]) (p[0][0] p[1][2] p[2][1] -
    p[0][0] p[1][1] p[2][2] - p[1][2] p[2][1] p[3][0] + p[1][1] p[2][2] p[3][0] -
    p[0][0] p[1][2] p[3][1] + p[1][2] p[2][0] p[3][1] + p[0][0] p[2][2] p[3][1] -
    p[1][0] p[2][2] p[3][1] + p[0][2] (p[1][1] (p[2][0] - p[3][0]) +
       p[2][1] p[3][0] - p[2][0] p[3][1] + p[1][0] (-p[2][1] + p[3][1])) +
    p[0][0] p[1][1] p[3][2] - p[1][1] p[2][0] p[3][2] - p[0][0] p[2][1] p[3][2] +
    p[1][0] p[2][1] p[3][2] + p[0][1] (-p[2][2] p[3][0] +
       p[1][2] (-p[2][0] + p[3][0]) + p[1][0] (p[2][2] - p[3][2]) + p[2][0] p[3][2])),
 1/120 (p[0][0]^2 + p[1][0]^2 + p[2][0]^2 + p[2][0] p[3][0] + p[3][0]^2 +
    p[1][0] (p[2][0] + p[3][0]) + p[0][0] (p[1][0] + p[2][0] + p[3][0]))
  (p[0][0] p[1][2] p[2][1] - p[0][0] p[1][1] p[2][2] - p[1][2] p[2][1] p[3][0] +
    p[1][1] p[2][2] p[3][0] - p[0][0] p[1][2] p[3][1] +
    p[1][2] p[2][0] p[3][1] + p[0][0] p[2][2] p[3][1] - p[1][0] p[2][2] p[3][1] +
    p[0][2] (p[1][1] (p[2][0] - p[3][0]) + p[2][1] p[3][0] - p[2][0] p[3][1] +
       p[1][0] (-p[2][1] + p[3][1])) + p[0][0] p[1][1] p[3][2] - p[1][1] p[2][0] p[3][2] -
    p[0][0] p[2][1] p[3][2] + p[1][0] p[2][1] p[3][2] + p[0][1] (-p[2][2] p[3][0] +
       p[1][2] (-p[2][0] + p[3][0]) + p[1][0] (p[2][2] - p[3][2]) + p[2][0] p[3][2])),
 -1/120 (2 p[1][0] p[1][1] + p[1][1] p[2][0] + p[1][0] p[2][1] + 2 p[2][0] p[2][1] +
    p[1][1] p[3][0] + p[2][1] p[3][0] + p[0][1] (p[1][0] + p[2][0] + p[3][0]) +
    p[1][0] p[3][1] + p[2][0] p[3][1] + 2 p[3][0] p[3][1] +
    p[0][0] (2 p[0][1] + p[1][1] + p[2][1] + p[3][1]))
  (-p[0][0] p[1][2] p[2][1] + p[0][0] p[1][1] p[2][2] + p[1][2] p[2][1] p[3][0] -
    p[1][1] p[2][2] p[3][0] + p[0][0] p[1][2] p[3][1] - p[1][2] p[2][0] p[3][1] -
    p[0][0] p[2][2] p[3][1] + p[1][0] p[2][2] p[3][1] + p[0][2] (-p[2][1] p[3][0] +
       p[1][1] (-p[2][0] + p[3][0]) + p[1][0] (p[2][1] - p[3][1]) + p[2][0] p[3][1]) -
    p[0][0] p[1][1] p[3][2] + p[1][1] p[2][0] p[3][2] + p[0][0] p[2][1] p[3][2] -
    p[1][0] p[2][1] p[3][2] + p[0][1] (p[1][2] (p[2][0] - p[3][0]) +
       p[2][2] p[3][0] - p[2][0] p[3][2] + p[1][0] (-p[2][2] + p[3][2]))),
 1/120 (2 p[1][0] p[1][2] + p[1][2] p[2][0] + p[1][0] p[2][2] + 2 p[2][0] p[2][2] +
    p[1][2] p[3][0] + p[2][2] p[3][0] + p[0][2] (p[1][0] + p[2][0] + p[3][0]) +
    p[1][0] p[3][2] + p[2][0] p[3][2] + 2 p[3][0] p[3][2] +
    p[0][0] (2 p[0][2] + p[1][2] + p[2][2] + p[3][2]))
  (p[0][0] p[1][2] p[2][1] - p[0][0] p[1][1] p[2][2] - p[1][2] p[2][1] p[3][0] +
    p[1][1] p[2][2] p[3][0] - p[0][0] p[1][2] p[3][1] +
    p[1][2] p[2][0] p[3][1] + p[0][0] p[2][2] p[3][1] - p[1][0] p[2][2] p[3][1] +
    p[0][2] (p[1][1] (p[2][0] - p[3][0]) + p[2][1] p[3][0] - p[2][0] p[3][1] +
       p[1][0] (-p[2][1] + p[3][1])) + p[0][0] p[1][1] p[3][2] - p[1][1] p[2][0] p[3][2] -
    p[0][0] p[2][1] p[3][2] + p[1][0] p[2][1] p[3][2] + p[0][1] (-p[2][2] p[3][0] +
       p[1][2] (-p[2][0] + p[3][0]) + p[1][0] (p[2][2] - p[3][2]) + p[2][0] p[3][2])),
 1/120 (p[0][1]^2 + p[1][1]^2 + p[2][1]^2 + p[2][1] p[3][1] + p[3][1]^2 +
```

```
---
    p[1][1] (p[2][1] + p[3][1]) + p[0][1] (p[1][1] + p[2][1] + p[3][1]))
  (p[0][0] p[1][2] p[2][1] - p[0][0] p[1][1] p[2][2] - p[1][2] p[2][1] p[3][0] +
    p[1][1] p[2][2] p[3][0] - p[0][0] p[1][2] p[3][1] +
    p[1][2] p[2][0] p[3][1] + p[0][0] p[2][2] p[3][1] - p[1][0] p[2][2] p[3][1] +
    p[0][2] (p[1][1] (p[2][0] - p[3][0]) + p[2][1] p[3][0] - p[2][0] p[3][1] +
        p[1][0] (-p[2][1] + p[3][1])) + p[0][0] p[1][1] p[3][2] - p[1][1] p[2][0] p[3][2] -
    p[0][0] p[2][1] p[3][2] + p[1][0] p[2][1] p[3][2] + p[0][1] (-p[2][2] p[3][0] +
        p[1][2] (-p[2][0] + p[3][0]) + p[1][0] (p[2][2] - p[3][2]) + p[2][0] p[3][2])),
 1/120 (2 p[1][1] p[1][2] + p[1][2] p[2][1] + p[1][1] p[2][2] + 2 p[2][1] p[2][2] +
    p[1][2] p[3][1] + p[2][2] p[3][1] + p[0][2] (p[1][1] + p[2][1] + p[3][1]) +
    p[1][1] p[3][2] + p[2][1] p[3][2] + 2 p[3][1] p[3][2] +
    p[0][1] (2 p[0][2] + p[1][2] + p[2][2] + p[3][2]))
  (p[0][0] p[1][2] p[2][1] - p[0][0] p[1][1] p[2][2] - p[1][2] p[2][1] p[3][0] +
    p[1][1] p[2][2] p[3][0] - p[0][0] p[1][2] p[3][1] +
    p[1][2] p[2][0] p[3][1] + p[0][0] p[2][2] p[3][1] - p[1][0] p[2][2] p[3][1] +
    p[0][2] (p[1][1] (p[2][0] - p[3][0]) + p[2][1] p[3][0] - p[2][0] p[3][1] +
        p[1][0] (-p[2][1] + p[3][1])) + p[0][0] p[1][1] p[3][2] - p[1][1] p[2][0] p[3][2] -
    p[0][0] p[2][1] p[3][2] + p[1][0] p[2][1] p[3][2] + p[0][1] (-p[2][2] p[3][0] +
        p[1][2] (-p[2][0] + p[3][0]) + p[1][0] (p[2][2] - p[3][2]) + p[2][0] p[3][2])),
 1/120 (p[0][2]^2 + p[1][2]^2 + p[2][2]^2 + p[2][2] p[3][2] + p[3][2]^2 +
    p[1][2] (p[2][2] + p[3][2]) + p[0][2] (p[1][2] + p[2][2] + p[3][2]))
  (p[0][0] p[1][2] p[2][1] - p[0][0] p[1][1] p[2][2] - p[1][2] p[2][1] p[3][0] +
    p[1][1] p[2][2] p[3][0] - p[0][0] p[1][2] p[3][1] +
    p[1][2] p[2][0] p[3][1] + p[0][0] p[2][2] p[3][1] - p[1][0] p[2][2] p[3][1] +
    p[0][2] (p[1][1] (p[2][0] - p[3][0]) + p[2][1] p[3][0] - p[2][0] p[3][1] +
        p[1][0] (-p[2][1] + p[3][1])) + p[0][0] p[1][1] p[3][2] - p[1][1] p[2][0] p[3][2] -
    p[0][0] p[2][1] p[3][2] + p[1][0] p[2][1] p[3][2] + p[0][1] (-p[2][2] p[3][0] +
        p[1][2] (-p[2][0] + p[3][0]) + p[1][0] (p[2][2] - p[3][2]) + p[2][0] p[3][2]))}
```

Check

```
b2[x_, y_] := {1, x, y, x^2/2, x y, y^2/2};
b2int = Integrate[Integrate[b2[x, y], {x, 0, 1 - y}], {y, 0, 1}];
pts = {{0, 0}, {1, 0}, {0, 1}} ;
val = basis2IntTri /. prules [pts];
val - b2int // Simplify
```

```
{0, 0, 0, 0, 0, 0}
```

```
b2[x_, y_] := {1, x, y, x^2/2, x y, y^2/2};
b2int = Integrate[b2[x, y], {x, 0, 1}, {y, 0, 1}];
pts = {{0, 0}, {1, 0}, {1, 1}, {0, 1}} ; No. It's very common .
   val = basis2IntQuad /. prules [pts];
val - b2int // SimplifyNo.
```

... Part: The expression 1 + i$ cannot be used as a part specification.

... Part: The expression 1 + j$ cannot be used as a part specification.

```
{0, 0, 0, 0, 0, 0}
```

```
b3[x_, y_, z_] := {1, x, y, z};
b3int = Integrate[b3[x, y, z], {x, 0, 1}, {y, 0, 1}, {z, 0, 1}];
pts =
  {{0, 0, 0}, {1, 0, 0}, {1, 1, 0}, {0, 1, 0}, {0, 0, 1}, {1, 0, 1}, {1, 1, 1}, {0, 1, 1}};
val = basis3IntHex /. prules [pts];
val - b3int // Simplify
```

... Part: The expression 1 + i$ cannot be used as a part specification.

```
{0, 0, 0, 0}
```

```
b3[x_, y_, z_] := {1, x, y, z};
b3int =
  Integrate[ Integrate[Integrate[b3[x, y, z], {x, 0, 1 - y}], {y, 0, 1}], {z, 0, 1}];
pts = {{0, 0, 0}, {1, 0, 0}, {0, 1, 0}, {0, 0, 1}, {1, 0, 1}, {0, 1, 1}};
val = basis3IntWedge /. prules [pts];
val - b3int // Simplify
```

... Part: The expression 1 + i$ cannot be used as a part specification.

```
{0, 0, 0, 0}
```

```
b3[x_, y_, z_] := basis3Quad[{x, y, z}];
b3int =
  Integrate[ Integrate[Integrate[b3[x, y, z], {x, 0, 1 - y - z}], {y, 0, 1 - z}], {z, 0, 1}];
pts = {{0, 0, 0}, {1, 0, 0}, {0, 1, 0}, {0, 0, 1}};
val = basis3IntTet /. prules [pts];
val - b3int // Simplify
```

... Part: The expression 1 + i$ cannot be used as a part specification.

```
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0}
```

```
codesub = {a_^b_ → pow[a, b]}
```

```
{a_^b_ → pow[a, b]}
```

Print results

```
SetDirectory["/scratch/gad/ngc/code/remap/math"]

/scratch/gad/ngc/code/remap/math

Do[ Print[InputForm[basis2IntTri[[i]]] /. codesub, "\n"], {i, Length[basis2IntTri]}]

Do[ Print[InputForm[basis2IntQuad[[i]]] /. codesub, "\n"], {i, Length[basis2IntQuad]}]

Do[ Print[InputForm[basis3IntHex[[i]]] /. codesub, "\n"], {i, Length[basis3IntHex]}]

Do[ Print["result[" <> ToString[i - 1] <> "][0]=\n",
   InputForm[basis3IntWedge[[i]]] /. codesub, ";\n"], {i, Length[basis3IntWedge]}];

Do[ Print["result[" <> ToString[i - 1] <> "][0]=\n",
  InputForm[basis3IntTet[[i]]] /. codesub, ";\n"], {i, Length[basis3IntTet]}]

result[0][0]=
(p[0][0] * p[1][2] * p[2][1] - p[0][0] * p[1][1] * p[2][2] - p[1][2] * p[2][1] * p[3][0] +
    p[1][1] * p[2][2] * p[3][0] - p[0][0] * p[1][2] * p[3][1] +
    p[1][2] * p[2][0] * p[3][1] + p[0][0] * p[2][2] * p[3][1] -
    p[1][0] * p[2][2] * p[3][1] + p[0][2] * (p[1][1] * (p[2][0] - p[3][0]) +
       p[2][1] * p[3][0] - p[2][0] * p[3][1] + p[1][0] * (-p[2][1] + p[3][1])) +
    p[0][0] * p[1][1] * p[3][2] - p[1][1] * p[2][0] * p[3][2] -
    p[0][0] * p[2][1] * p[3][2] + p[1][0] * p[2][1] * p[3][2] +
    p[0][1] * (-(p[2][2] * p[3][0]) + p[1][2] * (-p[2][0] + p[3][0]) +
       p[1][0] * (p[2][2] - p[3][2]) + p[2][0] * p[3][2])) / 6;

result[1][0]=
((p[0][0] + p[1][0] + p[2][0] + p[3][0]) *
    (p[0][0] * p[1][2] * p[2][1] - p[0][0] * p[1][1] * p[2][2] -
      p[1][2] * p[2][1] * p[3][0] + p[1][1] * p[2][2] * p[3][0] -
      p[0][0] * p[1][2] * p[3][1] + p[1][2] * p[2][0] * p[3][1] +
      p[0][0] * p[2][2] * p[3][1] - p[1][0] * p[2][2] * p[3][1] +
      p[0][2] * (p[1][1] * (p[2][0] - p[3][0]) + p[2][1] * p[3][0] -
         p[2][0] * p[3][1] + p[1][0] * (-p[2][1] + p[3][1])) +
      p[0][0] * p[1][1] * p[3][2] - p[1][1] * p[2][0] * p[3][2] -
      p[0][0] * p[2][1] * p[3][2] + p[1][0] * p[2][1] * p[3][2] +
      p[0][1] * (-(p[2][2] * p[3][0]) + p[1][2] * (-p[2][0] + p[3][0]) +
         p[1][0] * (p[2][2] - p[3][2]) + p[2][0] * p[3][2]))) / 24;
```

```
result[2][0]=
((p[0][1] + p[1][1] + p[2][1] + p[3][1]) *
     (p[0][0] * p[1][2] * p[2][1] - p[0][0] * p[1][1] * p[2][2] -
       p[1][2] * p[2][1] * p[3][0] + p[1][1] * p[2][2] * p[3][0] -
       p[0][0] * p[1][2] * p[3][1] + p[1][2] * p[2][0] * p[3][1] +
       p[0][0] * p[2][2] * p[3][1] - p[1][0] * p[2][2] * p[3][1] +
       p[0][2] * (p[1][1] * (p[2][0] - p[3][0]) + p[2][1] * p[3][0] -
          p[2][0] * p[3][1] + p[1][0] * (-p[2][1] + p[3][1])) +
       p[0][0] * p[1][1] * p[3][2] - p[1][1] * p[2][0] * p[3][2] -
       p[0][0] * p[2][1] * p[3][2] + p[1][0] * p[2][1] * p[3][2] +
       p[0][1] * (-(p[2][2] * p[3][0]) + p[1][2] * (-p[2][0] + p[3][0]) +
          p[1][0] * (p[2][2] - p[3][2]) + p[2][0] * p[3][2]))) / 24;

result[3][0]=
((p[0][2] + p[1][2] + p[2][2] + p[3][2]) *
     (p[0][0] * p[1][2] * p[2][1] - p[0][0] * p[1][1] * p[2][2] -
       p[1][2] * p[2][1] * p[3][0] + p[1][1] * p[2][2] * p[3][0] -
       p[0][0] * p[1][2] * p[3][1] + p[1][2] * p[2][0] * p[3][1] +
       p[0][0] * p[2][2] * p[3][1] - p[1][0] * p[2][2] * p[3][1] +
       p[0][2] * (p[1][1] * (p[2][0] - p[3][0]) + p[2][1] * p[3][0] -
          p[2][0] * p[3][1] + p[1][0] * (-p[2][1] + p[3][1])) +
       p[0][0] * p[1][1] * p[3][2] - p[1][1] * p[2][0] * p[3][2] -
       p[0][0] * p[2][1] * p[3][2] + p[1][0] * p[2][1] * p[3][2] +
       p[0][1] * (-(p[2][2] * p[3][0]) + p[1][2] * (-p[2][0] + p[3][0]) +
          p[1][0] * (p[2][2] - p[3][2]) + p[2][0] * p[3][2]))) / 24;

result[4][0]=
((pow[p[0][0], 2] + pow[p[1][0], 2] +
       pow[p[2][0], 2] + pow[p[3][0], 2] + p[2][0] * p[3][0] +
       p[1][0] * (p[2][0] + p[3][0]) + p[0][0] * (p[1][0] + p[2][0] + p[3][0])) *
     (p[0][0] * p[1][2] * p[2][1] - p[0][0] * p[1][1] * p[2][2] -
       p[1][2] * p[2][1] * p[3][0] + p[1][1] * p[2][2] * p[3][0] -
       p[0][0] * p[1][2] * p[3][1] + p[1][2] * p[2][0] * p[3][1] +
       p[0][0] * p[2][2] * p[3][1] - p[1][0] * p[2][2] * p[3][1] +
       p[0][2] * (p[1][1] * (p[2][0] - p[3][0]) + p[2][1] * p[3][0] -
          p[2][0] * p[3][1] + p[1][0] * (-p[2][1] + p[3][1])) +
       p[0][0] * p[1][1] * p[3][2] - p[1][1] * p[2][0] * p[3][2] -
       p[0][0] * p[2][1] * p[3][2] + p[1][0] * p[2][1] * p[3][2] +
       p[0][1] * (-(p[2][2] * p[3][0]) + p[1][2] * (-p[2][0] + p[3][0]) +
          p[1][0] * (p[2][2] - p[3][2]) + p[2][0] * p[3][2]))) / 120;
```

```
result[5][0]=
-((2*p[1][0]*p[1][1] + p[1][1]*p[2][0] + p[1][0]*p[2][1] +
       2*p[2][0]*p[2][1] + p[1][1]*p[3][0] + p[2][1]*p[3][0] +
       p[0][1]*(p[1][0] + p[2][0] + p[3][0]) + p[1][0]*p[3][1] + p[2][0]*p[3][1] +
       2*p[3][0]*p[3][1] + p[0][0]*(2*p[0][1] + p[1][1] + p[2][1] + p[3][1]))*
     (-(p[0][0]*p[1][2]*p[2][1]) + p[0][0]*p[1][1]*p[2][2] +
       p[1][2]*p[2][1]*p[3][0] - p[1][1]*p[2][2]*p[3][0] +
       p[0][0]*p[1][2]*p[3][1] - p[1][2]*p[2][0]*p[3][1] -
       p[0][0]*p[2][2]*p[3][1] + p[1][0]*p[2][2]*p[3][1] +
       p[0][2]*(-(p[2][1]*p[3][0]) + p[1][1]*(-p[2][0] + p[3][0]) +
          p[1][0]*(p[2][1] - p[3][1]) + p[2][0]*p[3][1]) -
       p[0][0]*p[1][1]*p[3][2] + p[1][1]*p[2][0]*p[3][2] +
       p[0][0]*p[2][1]*p[3][2] - p[1][0]*p[2][1]*p[3][2] +
       p[0][1]*(p[1][2]*(p[2][0] - p[3][0]) + p[2][2]*p[3][0] -
          p[2][0]*p[3][2] + p[1][0]*(-p[2][2] + p[3][2]))))/120;

result[6][0]=
((2*p[1][0]*p[1][2] + p[1][2]*p[2][0] + p[1][0]*p[2][2] +
      2*p[2][0]*p[2][2] + p[1][2]*p[3][0] + p[2][2]*p[3][0] +
      p[0][2]*(p[1][0] + p[2][0] + p[3][0]) + p[1][0]*p[3][2] + p[2][0]*p[3][2] +
      2*p[3][0]*p[3][2] + p[0][0]*(2*p[0][2] + p[1][2] + p[2][2] + p[3][2]))*
    (p[0][0]*p[1][2]*p[2][1] - p[0][0]*p[1][1]*p[2][2] -
      p[1][2]*p[2][1]*p[3][0] + p[1][1]*p[2][2]*p[3][0] -
      p[0][0]*p[1][2]*p[3][1] + p[1][2]*p[2][0]*p[3][1] +
      p[0][0]*p[2][2]*p[3][1] - p[1][0]*p[2][2]*p[3][1] +
      p[0][2]*(p[1][1]*(p[2][0] - p[3][0]) + p[2][1]*p[3][0] -
         p[2][0]*p[3][1] + p[1][0]*(-p[2][1] + p[3][1])) +
      p[0][0]*p[1][1]*p[3][2] - p[1][1]*p[2][0]*p[3][2] -
      p[0][0]*p[2][1]*p[3][2] + p[1][0]*p[2][1]*p[3][2] +
      p[0][1]*(-(p[2][2]*p[3][0]) + p[1][2]*(-p[2][0] + p[3][0]) +
         p[1][0]*(p[2][2] - p[3][2]) + p[2][0]*p[3][2])))/120;
```

```
result[7][0]=
((pow[p[0][1], 2] + pow[p[1][1], 2] +
      pow[p[2][1], 2] + pow[p[3][1], 2] + p[2][1] * p[3][1] +
      p[1][1] * (p[2][1] + p[3][1]) + p[0][1] * (p[1][1] + p[2][1] + p[3][1])) *
    (p[0][0] * p[1][2] * p[2][1] - p[0][0] * p[1][1] * p[2][2] -
      p[1][2] * p[2][1] * p[3][0] + p[1][1] * p[2][2] * p[3][0] -
      p[0][0] * p[1][2] * p[3][1] + p[1][2] * p[2][0] * p[3][1] +
      p[0][0] * p[2][2] * p[3][1] - p[1][0] * p[2][2] * p[3][1] +
      p[0][2] * (p[1][1] * (p[2][0] - p[3][0]) + p[2][1] * p[3][0] -
         p[2][0] * p[3][1] + p[1][0] * (-p[2][1] + p[3][1])) +
      p[0][0] * p[1][1] * p[3][2] - p[1][1] * p[2][0] * p[3][2] -
      p[0][0] * p[2][1] * p[3][2] + p[1][0] * p[2][1] * p[3][2] +
      p[0][1] * (-(p[2][2] * p[3][0]) + p[1][2] * (-p[2][0] + p[3][0]) +
         p[1][0] * (p[2][2] - p[3][2]) + p[2][0] * p[3][2]))) / 120;

result[8][0]=
((2 * p[1][1] * p[1][2] + p[1][2] * p[2][1] + p[1][1] * p[2][2] +
      2 * p[2][1] * p[2][2] + p[1][2] * p[3][1] + p[2][2] * p[3][1] +
      p[0][2] * (p[1][1] + p[2][1] + p[3][1]) + p[1][1] * p[3][2] + p[2][1] * p[3][2] +
      2 * p[3][1] * p[3][2] + p[0][1] * (2 * p[0][2] + p[1][2] + p[2][2] + p[3][2])) *
    (p[0][0] * p[1][2] * p[2][1] - p[0][0] * p[1][1] * p[2][2] -
      p[1][2] * p[2][1] * p[3][0] + p[1][1] * p[2][2] * p[3][0] -
      p[0][0] * p[1][2] * p[3][1] + p[1][2] * p[2][0] * p[3][1] +
      p[0][0] * p[2][2] * p[3][1] - p[1][0] * p[2][2] * p[3][1] +
      p[0][2] * (p[1][1] * (p[2][0] - p[3][0]) + p[2][1] * p[3][0] -
         p[2][0] * p[3][1] + p[1][0] * (-p[2][1] + p[3][1])) +
      p[0][0] * p[1][1] * p[3][2] - p[1][1] * p[2][0] * p[3][2] -
      p[0][0] * p[2][1] * p[3][2] + p[1][0] * p[2][1] * p[3][2] +
      p[0][1] * (-(p[2][2] * p[3][0]) + p[1][2] * (-p[2][0] + p[3][0]) +
         p[1][0] * (p[2][2] - p[3][2]) + p[2][0] * p[3][2]))) / 120;
```

```
result[9][0] =
((pow[p[0][2], 2] + pow[p[1][2], 2] +
      pow[p[2][2], 2] + pow[p[3][2], 2] + p[2][2] * p[3][2] +
      p[1][2] * (p[2][2] + p[3][2]) + p[0][2] * (p[1][2] + p[2][2] + p[3][2])) *
    (p[0][0] * p[1][2] * p[2][1] - p[0][0] * p[1][1] * p[2][2] -
      p[1][2] * p[2][1] * p[3][0] + p[1][1] * p[2][2] * p[3][0] -
      p[0][0] * p[1][2] * p[3][1] + p[1][2] * p[2][0] * p[3][1] +
      p[0][0] * p[2][2] * p[3][1] - p[1][0] * p[2][2] * p[3][1] +
      p[0][2] * (p[1][1] * (p[2][0] - p[3][0]) + p[2][1] * p[3][0] -
         p[2][0] * p[3][1] + p[1][0] * (-p[2][1] + p[3][1])) +
      p[0][0] * p[1][1] * p[3][2] - p[1][1] * p[2][0] * p[3][2] -
      p[0][0] * p[2][1] * p[3][2] + p[1][0] * p[2][1] * p[3][2] +
      p[0][1] * (-(p[2][2] * p[3][0]) + p[1][2] * (-p[2][0] + p[3][0]) +
         p[1][0] * (p[2][2] - p[3][2]) + p[2][0] * p[3][2]))) / 120;
```

Transfactors

evaluation

tf1 = transFactor[basis1, {x}]

$\left\{\{1, 0, 0\}, \{c[0], 1, 0\}, \left\{\frac{c[0]^2}{2}, c[0], 1\right\}\right\}$

tf1Lin = transFactor[basis1Lin, {x}]

{{1, 0}, {c[0], 1}}

tf2 = transFactor[basis2, {x, y}]

$\left\{\{1, 0, 0, 0, 0, 0\}, \{c[0], 1, 0, 0, 0, 0\}, \{c[1], 0, 1, 0, 0, 0\}, \left\{\frac{c[0]^2}{2}, c[0], 0, 1, 0, 0\right\}, \{c[0]\, c[1], c[1], c[0], 0, 1, 0\}, \left\{\frac{c[1]^2}{2}, 0, c[1], 0, 0, 1\right\}\right\}$

tf2Lin = transFactor[basis2Lin, {x, y}]

{{1, 0, 0}, {c[0], 1, 0}, {c[1], 0, 1}}

tf3 = transFactor[basis3, {x, y, z}]

{{1, 0, 0, 0}, {c[0], 1, 0, 0}, {c[1], 0, 1, 0}, {c[2], 0, 0, 1}}

```
tf3q = transFactor[basis3Quad, {x, y, z}]
```

$$\left\{\{1, 0, 0, 0, 0, 0, 0, 0, 0, 0\}, \{c[0], 1, 0, 0, 0, 0, 0, 0, 0, 0\}, \{c[1], 0, 1, 0, 0, 0, 0, 0, 0, 0\}, \{c[2], 0, 0, 1, 0, 0, 0, 0, 0, 0\}, \left\{\frac{c[0]^2}{2}, c[0], 0, 0, 1, 0, 0, 0, 0, 0\right\}, \{c[0]\, c[1], c[1], c[0], 0, 0, 1, 0, 0, 0, 0\}, \{c[0]\, c[2], c[2], 0, c[0], 0, 0, 1, 0, 0, 0\}, \left\{\frac{c[1]^2}{2}, 0, c[1], 0, 0, 0, 0, 1, 0, 0\right\}, \{c[1]\, c[2], 0, c[2], c[1], 0, 0, 0, 0, 1, 0\}, \left\{\frac{c[2]^2}{2}, 0, 0, c[2], 0, 0, 0, 0, 0, 1\right\}\right\}$$

test

```
tf1.basis1[{x[0]}] - basis1[{x[0] + c[0]}] // Simplify
{0, 0, 0}

tf1Lin.basis1Lin[{x[0]}] - basis1Lin[{x[0] + c[0]}] // Simplify
{0, 0}

tf2Lin.basis2Lin[{x[0], x[1]}] - basis2Lin[{x[0] + c[0], x[1] + c[1]}] // Simplify
{0, 0, 0}

tf2.basis2[{x[0], x[1]}] - basis2[{x[0] + c[0], x[1] + c[1]}] // Simplify
{0, 0, 0, 0, 0, 0}

tf3.basis3[{x[0], x[1], x[2]}] - basis3[{x[0] + c[0], x[1] + c[1], x[2] + c[2]}] // Simplify
{0, 0, 0, 0}

tf3q.basis3Quad[{x[0], x[1], x[2]}] -
  basis3Quad[{x[0] + c[0], x[1] + c[1], x[2] + c[2]}] // Simplify
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0}
```

print

```
StringJoin[Table[StringJoin[{"tf[", ToString[i - 1], "][", ToString[j - 1],
    "]=", ToString[InputForm[tf1Lin[[i, j]]]], ";\n"}], {i, 2}, {j, 2}]]
tf[0][0]=1;
tf[0][1]=0;
tf[1][0]=c[0];
tf[1][1]=1;
```

```
StringJoin[Table[StringJoin[{"tf[", ToString[i - 1], "][", ToString[j - 1], "]=",
    ToString[InputForm[tf1[[i, j]] /. codesub]], ";\n"}], {i, 3}, {j, 3}]]
```

```
tf[0][0]=1;
tf[0][1]=0;
tf[0][2]=0;
tf[1][0]=c[0];
tf[1][1]=1;
tf[1][2]=0;
tf[2][0]=pow[c[0], 2]/2;
tf[2][1]=c[0];
tf[2][2]=1;
```

```
StringJoin[Table[StringJoin[{"tf[", ToString[i - 1], "][", ToString[j - 1],
    "]=", ToString[InputForm[tf2Lin[[i, j]]]], ";\n"}], {i, 3}, {j, 3}]]
```

```
tf[0][0]=1;
tf[0][1]=0;
tf[0][2]=0;
tf[1][0]=c[0];
tf[1][1]=1;
tf[1][2]=0;
tf[2][0]=c[1];
tf[2][1]=0;
tf[2][2]=1;
```

```
StringJoin[Table[StringJoin[{"tf[", ToString[i - 1], "][", ToString[j - 1], "]=",
   ToString[InputForm[tf2[[i, j]] /. codesub]], ";\n"}], {i, 6}, {j, 6}]]
```

```
tf[0][0]=1;
tf[0][1]=0;
tf[0][2]=0;
tf[0][3]=0;
tf[0][4]=0;
tf[0][5]=0;
tf[1][0]=c[0];
tf[1][1]=1;
tf[1][2]=0;
tf[1][3]=0;
tf[1][4]=0;
tf[1][5]=0;
tf[2][0]=c[1];
tf[2][1]=0;
tf[2][2]=1;
tf[2][3]=0;
tf[2][4]=0;
tf[2][5]=0;
tf[3][0]=pow[c[0], 2]/2;
tf[3][1]=c[0];
tf[3][2]=0;
tf[3][3]=1;
tf[3][4]=0;
tf[3][5]=0;
tf[4][0]=c[0]*c[1];
tf[4][1]=c[1];
tf[4][2]=c[0];
tf[4][3]=0;
tf[4][4]=1;
tf[4][5]=0;
tf[5][0]=pow[c[1], 2]/2;
tf[5][1]=0;
tf[5][2]=c[1];
tf[5][3]=0;
tf[5][4]=0;
tf[5][5]=1;
```

```
StringJoin[Table[StringJoin[{"tf[", ToString[i - 1], "][", ToString[j - 1], "]=",
    ToString[InputForm[tf3[[i, j]] /. codesub]], ";\n"}], {i, 4}, {j, 4}]]
```

```
tf[0][0]=1;
tf[0][1]=0;
tf[0][2]=0;
tf[0][3]=0;
tf[1][0]=c[0];
tf[1][1]=1;
tf[1][2]=0;
tf[1][3]=0;
tf[2][0]=c[1];
tf[2][1]=0;
tf[2][2]=1;
tf[2][3]=0;
tf[3][0]=c[2];
tf[3][1]=0;
tf[3][2]=0;
tf[3][3]=1;
```

```
StringJoin[Table[StringJoin[{"tf[", ToString[i - 1], "][", ToString[j - 1], "]=",
    ToString[InputForm[tf3q[[i, j]] /. codesub]], ";\n"}], {i, 10}, {j, 10}]]
```

```
tf[0][0]=1;
tf[0][1]=0;
tf[0][2]=0;
tf[0][3]=0;
tf[0][4]=0;
tf[0][5]=0;
tf[0][6]=0;
tf[0][7]=0;
tf[0][8]=0;
tf[0][9]=0;
tf[1][0]=c[0];
tf[1][1]=1;
tf[1][2]=0;
tf[1][3]=0;
tf[1][4]=0;
tf[1][5]=0;
tf[1][6]=0;
tf[1][7]=0;
tf[1][8]=0;
tf[1][9]=0;
tf[2][0]=c[1];
tf[2][1]=0;
tf[2][2]=1;
```

```
tf[2][3]=0;
tf[2][4]=0;
tf[2][5]=0;
tf[2][6]=0;
tf[2][7]=0;
tf[2][8]=0;
tf[2][9]=0;
tf[3][0]=c[2];
tf[3][1]=0;
tf[3][2]=0;
tf[3][3]=1;
tf[3][4]=0;
tf[3][5]=0;
tf[3][6]=0;
tf[3][7]=0;
tf[3][8]=0;
tf[3][9]=0;
tf[4][0]=pow[c[0], 2]/2;
tf[4][1]=c[0];
tf[4][2]=0;
tf[4][3]=0;
tf[4][4]=1;
tf[4][5]=0;
tf[4][6]=0;
tf[4][7]=0;
tf[4][8]=0;
tf[4][9]=0;
tf[5][0]=c[0]*c[1];
tf[5][1]=c[1];
tf[5][2]=c[0];
tf[5][3]=0;
tf[5][4]=0;
tf[5][5]=1;
tf[5][6]=0;
tf[5][7]=0;
tf[5][8]=0;
tf[5][9]=0;
tf[6][0]=c[0]*c[2];
tf[6][1]=c[2];
tf[6][2]=0;
tf[6][3]=c[0];
tf[6][4]=0;
tf[6][5]=0;
tf[6][6]=1;
tf[6][7]=0;
```

```
tf[6][8]=0;
tf[6][9]=0;
tf[7][0]=pow[c[1], 2]/2;
tf[7][1]=0;
tf[7][2]=c[1];
tf[7][3]=0;
tf[7][4]=0;
tf[7][5]=0;
tf[7][6]=0;
tf[7][7]=1;
tf[7][8]=0;
tf[7][9]=0;
tf[8][0]=c[1]*c[2];
tf[8][1]=0;
tf[8][2]=c[2];
tf[8][3]=c[1];
tf[8][4]=0;
tf[8][5]=0;
tf[8][6]=0;
tf[8][7]=0;
tf[8][8]=1;
tf[8][9]=0;
tf[9][0]=pow[c[2], 2]/2;
tf[9][1]=0;
tf[9][2]=0;
tf[9][3]=c[2];
tf[9][4]=0;
tf[9][5]=0;
tf[9][6]=0;
tf[9][7]=0;
tf[9][8]=0;
tf[9][9]=1;
```

Example of rotated and stretched quad and hex

```
deform2 = DiagonalMatrix[{5.2, .17}].RotationMatrix[Pi / 2 * .63]
```

```
{{2.85492, -4.3462}, {0.142087, 0.0933339}}
```

```
Det[deform2]
```

```
0.884
```

```
Det[deform2] - Times @@ {5.2, .17}
```

```
0.
```

```
deform3 = DiagonalMatrix[{5.2, .17, 2.03}].RotationMatrix[Pi/2*.63, {1., -2., 1.63}]
```

```
{{3.16119, -3.17272, -2.64211},
 {0.0636729, 0.133385, -0.08399}, {1.42121, 0.223387, 1.43218}}
```

```
Det[deform3]
```

```
1.79452
```

```
Det[deform3] - Times @@ {5.2, .17, 2.03}
```

-1.33227×10^{-15}

```
InputForm[deform2]
```

```
{{2.8549186535902855, -4.346198279115005}, {0.14208725143260595, 0.0933338790596824}}
```

```
InputForm[deform3]
```

```
{{3.1611889909865836, -3.1727215693209625, -2.6421056009990864}, {0.0636728533375156, 0
```