TENSOR ALGEBRA

Ph. Zysset, April 20th 2008

Real number space

```
n1=Sin[t]*Cos[u];
n2=Sin[t]*Sin[u];
n3=Cos[t];
```

Vector spaces

Vectors or rank one tensors

Canonical

```
e1={1,0,0};
e2={0,1,0};
e3={0,0,1};
```

Arguments (triclinic)

```
a=Table["a"<>ToString[i],{i,1,3}];
b=Table["b"<>ToString[i],{i,1,3}];
c=Table["c"<>ToString[i],{i,1,3}];
d=Table["d"<>ToString[i],{i,1,3}];
```

```
as = Table["a" <> ToString[i], {i, 1, 6}];
bs = Table["b" <> ToString[i], {i, 1, 6}];
cs = Table["c" <> ToString[i], {i, 1, 6}];
```

```
au = Table["a" <> ToString[i], {i, 1, 9}];
bu = Table["b" <> ToString[i], {i, 1, 9}];
cu = Table["c" <> ToString[i], {i, 1, 9}];
```

```
aw = Table["a" <> ToString[i], {i, 1, 21}];
bw = Table["b" <> ToString[i], {i, 1, 21}];
cw = Table["c" <> ToString[i], {i, 1, 21}];
```

Variables

```
x = Table["x" <> ToString[i], {i, 1, 3}];
y = Table["y" <> ToString[i], {i, 1, 3}];
z = Table["z" <> ToString[i], {i, 1, 3}];
```

```
xs = Table["x" <> ToString[i], {i, 1, 6}];
ys = Table["y" <> ToString[i], {i, 1, 6}];
zs = Table["z" <> ToString[i], {i, 1, 6}];
```

```
xu = Table["x" <> ToString[i], {i, 1, 9}];
yu = Table["y" <> ToString[i], {i, 1, 9}];
zu = Table["z" <> ToString[i], {i, 1, 9}];
```

Numerical

```
xnum={1.0,2.5,-2.0};
ynum={-1.0,3.5,2.5};
znum={3.0,-1.5,2.0};
```

Special

```
n={n1,n2,n3};
```

Operations

```
x.y;
```

```
dist1[a_,b_]:=Sqrt[(b-a).(b-a)]
```

```
norm1[a_]:=Sqrt[a.a]
```

```
Cross[x,y];
```

```
dyad[a_,b_]:=Outer[Times,a,b]
```

```
nvn[a_]:=(a.n)*a
```

```
ntn[a_]:=a-(a.n)a
```

Second order tensor space

Second order tensors

Canonical basis

```
Do[Evaluate[Symbol["ee" <> ToString[i] <> ToString[j]]] =
  Evaluate[dyad[Symbol["e" <> ToString[i]], Symbol["e" <> ToString[j]]]], {i, 1, 3}, {j, 1, 3}]
```

Set::setraw : Cannot assign to raw object 1. ≫

Set::setraw : Cannot assign to raw object 0. ≫

Set::setraw : Cannot assign to raw object 0. ≫

General::stop : Further output of Set::setraw will be suppressed during this calculation . ≫

Arguments

```
aa=Table["a"<>ToString[i]<>ToString[j],{i,1,3},{j,1,3}];
bb=Table["b"<>ToString[i]<>ToString[j],{i,1,3},{j,1,3}];
cc=Table["c"<>ToString[i]<>ToString[j],{i,1,3},{j,1,3}];
dd=Table["d"<>ToString[i]<>ToString[j],{i,1,3},{j,1,3}];
ff=Table["f"<>ToString[i]<>ToString[j],{i,1,3},{j,1,3}];
```

```
aas = Table["a" <> ToString[i] <> ToString[j], {i, 1, 6}, {j, 1, 6}];
bbs = Table["b" <> ToString[i] <> ToString[j], {i, 1, 6}, {j, 1, 6}];
ccs = Table["c" <> ToString[i] <> ToString[j], {i, 1, 6}, {j, 1, 6}];
```

```
aau = Table["a" <> ToString[i] <> ToString[j], {i, 1, 9}, {j, 1, 9}];
bbu = Table["b" <> ToString[i] <> ToString[j], {i, 1, 9}, {j, 1, 9}];
ccu = Table["c" <> ToString[i] <> ToString[j], {i, 1, 9}, {j, 1, 9}];
```

Variables

```
xx = Table["x" <> ToString[i] <> ToString[j], {i, 1, 3}, {j, 1, 3}];
yy = Table["y" <> ToString[i] <> ToString[j], {i, 1, 3}, {j, 1, 3}];
zz = Table["z" <> ToString[i] <> ToString[j], {i, 1, 3}, {j, 1, 3}];
```

```
xxu = Table["x" <> ToString[i] <> ToString[j], {i, 1, 9}, {j, 1, 9}];
yyu = Table["y" <> ToString[i] <> ToString[j], {i, 1, 9}, {j, 1, 9}];
zzu = Table["z" <> ToString[i] <> ToString[j], {i, 1, 9}, {j, 1, 9}];
```

```
xxnum={{1.0,2.5,1.5},{-2.0,3.0,1.5},{1.5,-1.0,-2.5}};
yynum={{-2.0,-0.5,-1.0},{1.5,1.0,-2.5},{3.0,1.0,0.5}};
zznum={{3.0,0.5,-0.5},{1.0,3.5,-1.0},{-2.5,1.5,2.0}};
```

Normal

```
nn=dyad[n,n];
```

Fabric tensors

```
mm1 = dyad[e1, e1];
mm2 = dyad[e2, e2];
mm3 = dyad[e3, e3];
mm4 = Sqrt[2] / 2 * (dyad[e2, e3] + dyad[e3, e2]);
mm5 = Sqrt[2] / 2 * (dyad[e3, e1] + dyad[e1, e3]);
mm6 = Sqrt[2] / 2 * (dyad[e1, e2] + dyad[e2, e1]);
```

```
mm = μ1 * mm1 + μ2 * mm2 + μ3 * mm3;
```

```
mma := dyad[a, a]
mmb := dyad[b, b];
mmc := dyad[c, c];
mmbc := dyad[b, c] + dyad[c, b];
mmca := dyad[c, a] + dyad[a, c];
mmab := dyad[a, b] + dyad[b, a];
```

Isomorphism of second rank tensors with a dimension 9 vector space

Projection

```
pru2[aa_] := Table[aa[[Floor[(i - 1) / 3 + 1], Mod[i - 1, 3] + 1]], {i, 1, 9}]
```

```
pru2[aa]
```

```
{a11, a12, a13, a21, a22, a23, a31, a32, a33}
```

Elevation

```
blu2[au_] := Table[au[[3 * (i - 1) + j]], {i, 1, 3}, {j, 1, 3}]
```

```
blu2[au]
```

```
{{a1, a2, a3}, {a4, a5, a6}, {a7, a8, a9}}
```

Symmetric second rank tensors

Variables

```
ee={{e11,e12,e31},{e12,e22,e23},{e31,e23,e33}};
```

```
ss={{s11,s12,s31},{s12,s22,s23},{s31,s23,s33}};
```

```
Simplify[ee - Transpose[ee]]
```

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

```
Simplify[ss - Transpose[ss]]
```

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

Numerical

```
eenum={{1.0,0.5,0.5},{0.5,2.0,1.0},{0.5,1.0,1.5}};
ssnum={{2.0,0.0,0.0},{0.0,1.0,0.0},{0.0,0.0,1.5}};
```

Isomorphism of symmetric second rank tensors with a dimension 6 vector space

Projection

```
proj2[aa_]:={aa[[1,1]],aa[[2,2]],aa[[3,3]],
1/Sqrt[2]*(aa[[2,3]]+aa[[3,2]]),1/Sqrt[2]*(aa[[1,3]]+aa[[3,1]]),1/Sqrt[2]*(aa[[1,2]]+aa[[2,1]])}
```

Elevation

```
blow2[a_]:={{a[[1]],a[[6]]/Sqrt[2],a[[5]]/Sqrt[2]},
            {a[[6]]/Sqrt[2],a[[2]],a[[4]]/Sqrt[2]},
            {a[[5]]/Sqrt[2],a[[4]]/Sqrt[2],a[[3]]}}
```

Wrong projection

```
pro2[aa_]:={aa[[1,1]],aa[[2,2]],aa[[3,3]],
1/2*(aa[[2,3]]+aa[[3,2]]),1/2*(aa[[1,3]]+aa[[3,1]]),1/2*(aa[[1,2]]+aa[[2,1]])}
```

Wrong blow up

```
blo2[a_]:={{a[[1]],a[[6]],a[[5]]},
            {a[[6]],a[[2]],a[[4]]},
            {a[[5]],a[[4]],a[[3]]}}
```

```
blo2[{e11,e22,e33,e23,e31,e12}]
```

```
{{e11, e12, e31}, {e12, e22, e23}, {e31, e23, e33}}
```

Isomorphism of symmetric second rank tensors with a dimension 21 vector space

Projection

```
prsj2[aas_]:={aas[[1,1]],aas[[2,2]],aas[[3,3]],aas[[4,4]],aas[[5,5]],aas[[6,6]],
Sqrt[2]*aas[[5,6]],Sqrt[2]*aas[[4,6]],Sqrt[2]*aas[[3,6]],Sqrt[2]*aas[[2,6]],Sqrt[2]*aas[[1,6]],Sqrt
```

Blow up

```
blsw2[aw_]:={{aw[[1]],aw[[21]]/Sqrt[2],aw[[20]]/Sqrt[2],aw[[18]]/Sqrt[2],aw[[15]]/Sqrt[2],aw[[11]]/Sqrt[2
```

Antisymmetric second rank tensor

```
anti={{0,a12,-a31},{-a12,0.0,a23},{a31,-a23,0}};
```

```
antinum={{0.0,0.5,0.5},{-0.5,0.0,1.0},{-0.5,-1.0,0}};
```

```
pra2[aa_]:={aa[[2,3]],aa[[3,1]],aa[[1,2]]}
```

```
bla2[a_]:={{0,a[[3]],-a[[2]]},
           {-a[[3]],0,a[[1]]},
           {a[[2]],-a[[1]],0}}
```

Addition

Neutral tensor

```
ne2={{0.0,0.0,0.0},{0.0,0.0,0.0},{0.0,0.0,0.0}};
```

Internal Product

Identity

```
id2 = dyad[e1, e1] + dyad[e2, e2] + dyad[e3, e3];
```

Cotensor

```
cot[aa_] := Det[aa] * Inverse[aa]
```

Scalar Product, distance and norm

```
t222[aa_,bb_]:=Sum[Inner[Times,aa,Transpose[bb,{2,1}],Plus]\
[[i,i]],{i,Length[aa[[1]]]}]
```

```
dist2[aa_,bb_]:=Sqrt[t222[aa-bb,aa-bb]]
```

```
norm2[aa_]:=Sqrt[t222[aa,aa]]
```

```
normln2[aa_] := Sqrt[Sum[Log[Eigenvalues[aa][[i]]]^2, {i, Length[aa[[1]]]}]]
```

Unitary transformations

Rotations

```
rr[α_,β_,γ_]:=\
{{Cos[α]Cos[β]Cos[γ]-Sin[α]Sin[γ],
 -Cos[α]Cos[β]Sin[γ]-Sin[α]Cos[γ],
  Cos[α]Sin[β]},
 {Sin[α]Cos[β]Cos[γ]+Cos[α]Sin[γ],
 -Sin[α]Cos[β]Sin[γ]+Cos[α]Cos[γ],
  Sin[α]Sin[β]},
  {-Sin[β]Cos[γ],Sin[β]Sin[γ],Cos[β]}}
```

```
rr1[t_] := {{1, 0, 0}, {0, Cos[t], -Sin[t]}, {0, Sin[t], Cos[t]}}
```

```
rr2[t_] := {{Cos[t], 0, Sin[t]}, {0, 1, 0}, {-Sin[t], 0, Cos[t]}}
```

```
rr3[t_] := {{Cos[t], -Sin[t], 0}, {Sin[t], Cos[t], 0}, {0, 0, 1}}
```

Invariants

```
Tr[xx];
```

```
sec[aa_]:=1/2*(Tr[aa]*Tr[aa]-Tr[aa.aa])
```

```
Det[xx]
```

- x13 x22 x31 + x12 x23 x31 + x13 x21 x32 - x11 x23 x32 - x12 x21 x33 + x11 x22 x33

```
inv1[aa_]:=Tr[aa]
```

```
inv2[aa_] := 3 * sec[aa] - Tr[aa] * Tr[aa]
```

```
inv3[aa_] := 27 * Det[aa] - 9 sec[aa] * Tr[aa] + 2 * Tr[aa]^3
```

```
inv0[aa_]:=4*inv2[aa]^3+inv3[aa]^2
```

```
invc[aa_]:=inv3[aa]+Sqrt[inv0[aa]]
```

Rivlin identity

```
Simplify[Tr[aa.bb.aa.bb]+2Tr[aa.aa.bb.bb]
-2*Tr[bb.aa.bb]*Tr[aa]
-2*Tr[aa.aa.bb]*Tr[bb]
-Tr[aa.bb]^2
+2*Tr[aa.bb]*Tr[bb]*Tr[aa]
-1/2*Tr[bb.bb]*Tr[aa.aa]
+1/2*Tr[bb.bb]*Tr[aa]^2
+1/2*Tr[bb]^2*Tr[aa.aa]
-1/2*Tr[aa]^2*Tr[bb]^2]
```

```
0
```

Von Mises equivalent stress

```
vmises[ss_] := Sqrt[3 / 2] * norm2[dev2[ss]]
```

Second order tensor products

Rank one product

```
tcro[aa_,bb_]:=Outer[Times,aa,bb]
```

Overline product

```
tove[aa_,bb_]:=Transpose[Outer[Times,aa,bb],{1,4,2,3}]
```

Underline product

```
tund[aa_,bb_]:=Transpose[Outer[Times,aa,bb],{1,3,2,4}]
```

Symetric product

```
tdou[aa_,bb_]:=1/2*(Transpose[Outer[Times,aa,bb],{1,3,2,4}]+
                 Transpose[Outer[Times,aa,bb],{1,4,2,3}])
```

Theorems

Rivlin theorem

```
rivlin[aa_,bb_,cc_]:=aa.bb.cc+aa.cc.bb+bb.aa.cc+bb.cc.aa+cc.aa.bb+cc.bb.aa-(bb.cc+cc.b
```

Cayley-Hamilton theorem

```
Simplify[(xx.xx.xx)-(xx.xx)*Tr[xx]+
1/2*xx*(Tr[xx]*Tr[xx]-Tr[xx.xx])-Det[xx]*id2]
```

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

Decomposition RU

```
fuu[aa_]:=Sum[Sqrt[Eigenvalues[Transpose[aa].aa][[i]]]/((Eigenvectors[Transpose[aa].aa][[i]]).(Ei
```

```
frr[aa_]:=aa.Inverse[fuu[aa]]
```

Decomposition in trace and deviator

Deviator

```
dev2[aa_]:=aa-1/3*Tr[aa]*id2
```

Decomposition check

```
Simplify[ee-(dev2[ee]+1/3*Tr[ee]*id)]
```

$$\left\{\left\{-\frac{1}{3}(e11+e22+e33)(-1+id),\ -\frac{1}{3}(e11+e22+e33)\,id,\ -\frac{1}{3}(e11+e22+e33)\,id\right\},\right.$$
$$\left\{-\frac{1}{3}(e11+e22+e33)\,id,\ -\frac{1}{3}(e11+e22+e33)(-1+id),\ -\frac{1}{3}(e11+e22+e33)\,id\right\},$$
$$\left.\left\{-\frac{1}{3}(e11+e22+e33)\,id,\ -\frac{1}{3}(e11+e22+e33)\,id,\ -\frac{1}{3}(e11+e22+e33)(-1+id)\right\}\right\}$$

Decomposition in symmetric and antisymmetric tensors

Third order product

```
t322[aaa_,bb_]:=Sum[Transpose[Inner[Times,aaa,Transpose
[bb,{2,1}],Plus],{3,2,1}][[i,i]],{i,Length[bb[[1]]]}]
```

Levi-Civita tensor

```
eps={{{0,0,0},{0,0,1},{0,-1,0}},
        {{0,0,-1},{0,0,0},{1,0,0}},
        {{0,1,0},{-1,0,0},{0,0,0}}};
```

Decomposition

```
Simplify[xx-(1/2*(xx+Transpose[xx])+
1/2*eps.t322[eps,xx])]
```

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

Fourth order tensor space

General fourth order tensors

Canonical

```
Do[Evaluate[Symbol["eeee" <> ToString[i] <> ToString[j] <> ToString[k] <> ToString[l]]] =
  Evaluate[dyad[Symbol["ee" <> ToString[i] <> ToString[j]],
    Symbol["ee" <> ToString[k] <> ToString[l]]]], {i, 1, 3}, {j, 1, 3}, {k, 1, 3}, {l, 1, 3}]
```

```
Set::setraw : Cannot assign to raw object 1. >>
Set::setraw : Cannot assign to raw object 0. >>
Set::setraw : Cannot assign to raw object 0. >>
General::stop : Further output of Set::setraw will be suppressed during this calculation . >>
```

Arguments

```
aaaa=Table["a"<>ToString[i]<>ToString[j]<>ToString[k]<>ToString[l],{i,1,3},{j,1,3},{k,1,3},
bbbb=Table["b"<>ToString[i]<>ToString[j]<>ToString[k]<>ToString[l],{i,1,3},{j,1,3},{k,1,3},
cccc=Table["b"<>ToString[i]<>ToString[j]<>ToString[k]<>ToString[l],{i,1,3},{j,1,3},{k,1,3},
```

Variables

```
xxxx=Table["x"<>ToString[i]<>ToString[j]<>ToString[k]<>ToString[l],{i,1,3},{j,1,3},{k,1,3},
yyyy=Table["y"<>ToString[i]<>ToString[j]<>ToString[k]<>ToString[l],{i,1,3},{j,1,3},{k,1,3},
zzzz=Table["z"<>ToString[i]<>ToString[j]<>ToString[k]<>ToString[l],{i,1,3},{j,1,3},{k,1,3},
```

```
xxxxnum={{{{2,0,1},{0,0,0},{0,0,0}},{{0,1,0},{0,0,0},{0,0,0}},
{{0,0,1},{0,1,0},{0,0,0}}},{{{0,0,0},   {1,0,  0}, {0,0,0}},
{{0,1,0},{0,3,0},{0,0,0}},{{0,0,0},{0,0,1},{0,0,0}}},
{{{0,0,0},  {1,0,  0},  {0,0,0}},{{0,1,0},{0,3,0},{0,0,0}},
{{0,0,0},{0,0,0},{0,0,1}}}};
yyyynum={{{{3,0,0},{0,0,0},{0,1,0}},{{0,1,0},{0,0,0},{0,0,0}},
{{0,0,1},{0,0,0},{0,0,0}}},{{{0,1,0},   {1,0,  0}, {0,1,0}},
{{0,0,0},{0,1,0},{0,0,1}},{{0,0,0},{0,1,1},{0,0,0}}},
{{{0,0,0},  {1,0,  0},  {0,0,0}},{{0,1,0},{0,3,0},{0,0,0}},
{{0,1,0},{0,0,0},{0,0,2}}}};
```

Isomorphism of fourth order tensors with R9xR9

Projection

```
pru4[aaaa_] := Table[aaaa[[Floor[(m - 1) / 3 + 1],
    Mod[m - 1, 3] + 1, Floor[(n - 1) / 3 + 1], Mod[n - 1, 3] + 1]], {m, 1, 9}, {n, 1, 9}]
```

Expansion

```
blu4[aau_] := Table[aau[[3 * (i - 1) + j, 3 * (k - 1) + l]], {i, 1, 3}, {j, 1, 3}, {k, 1, 3}, {l, 1, 3}]
```

Symmetric fourth order tensors

Variables

```
eeee=
{{{{e1111,e1112,e1131},{e1112,e1122,e1123},{e1131,e1123,e1133}},
  {{e1112,e1212,e1231},{e1212,e1222,e1223},{e1231,e1223,e1233}},
  {{e1131,e1231,e3131},{e1231,e3122,e3123},{e3131,e3123,e3133}}},
 {{{e1112,e1212,e1231},{e1212,e1222,e1223},{e1231,e1223,e1233}},
  {{e1122,e1222,e3122},{e1222,e2222,e2223},{e3122,e2223,e2233}},
  {{e1123,e1223,e3123},{e1223,e2223,e2323},{e3123,e2323,e2333}}},
 {{{e1131,e1231,e3131},{e1231,e3122,e3123},{e3131,e3123,e3133}},
  {{e1123,e1223,e3123},{e1223,e2223,e2323},{e3123,e2323,e2333}},
  {{e1133,e1233,e3133},{e1233,e2233,e2333},{e3133,e2333,e3333}}}};
```

```
ssss=
{{{{s1111,s1112,s1131},{s1112,s1122,s1123},{s1131,s1123,s1133}},
  {{s1112,s1212,s1231},{s1212,s1222,s1223},{s1231,s1223,s1233}},
  {{s1131,s1231,s3131},{s1231,s3122,s3123},{s3131,s3123,s3133}}},
 {{{s1112,s1212,s1231},{s1212,s1222,s1223},{s1231,s1223,s1233}},
  {{s1122,s1222,s3122},{s1222,s2222,s2223},{s3122,s2223,s2233}},
  {{s1123,s1223,s3123},{s1223,s2223,s2323},{s3123,s2323,s2333}}},
 {{{s1131,s1231,s3131},{s1231,s3122,s3123},{s3131,s3123,s3133}},
  {{s1123,s1223,s3123},{s1223,s2223,s2323},{s3123,s2323,s2333}},
  {{s1133,s1233,s3133},{s1233,s2233,s2333},{s3133,s2333,s3333}}}};
```

Numerical

```
eeeenum={{{{2,0,0},{0,0,0},{0,0,0}},{{0,1,0},{0,0,0},{0,0,0}},
 {{0,0,1},{0,0,0},{0,0,0}}},
{{{0,0,0},{1,0,0},{0,0,0}},{{0,0,0},{0,3,0},{0,0,0}},
{{0,0,0},{0,0,1},{0,0,0}}},
{{{0,0,0},{0,0,0},{1,0,0}},{{0,0,0},{0,0,0},{0,1,0}},
{{0,0,0},{0,0,0},{0,0,1}}}};
ssssnum={{{{1,0,0},{0,0,0},{0,0,0}},{{0,1,0},{0,0,0},{0,0,0}},
 {{0,0,1},{0,0,0},{0,0,0}}},
{{{0,0,0},{1,0,0},{0,0,0}},{{0,0,0},{0,3,0},{0,0,0}},
{{0,0,0},{0,0,1},{0,0,0}}},
{{{0,0,0},{0,0,0},{1,0,0}},{{0,0,0},{0,0,0},{0,1,0}},
{{0,0,0},{0,0,0},{0,0,3}}}};
```

Isomorphism of fourth order tensors having two minor symetries with R6xR6

Invariant projection

```
proj4[aaaa_]:=\
{{aaaa[[1,1,1,1]],aaaa[[1,1,2,2]],aaaa[[1,1,3,3]],
 1/Sqrt[2]*(aaaa[[1,1,2,3]]+aaaa[[1,1,3,2]]),1/Sqrt[2]*(aaaa[[1,1,3,1]]+aaaa[[1,1,1,3]]),
 1/Sqrt[2]*(aaaa[[1,1,1,2]]+aaaa[[1,1,2,1]])},
 {aaaa[[2,2,1,1]],aaaa[[2,2,2,2]],aaaa[[2,2,3,3]],
 1/Sqrt[2]*(aaaa[[2,2,2,3]]+aaaa[[2,2,3,2]]),1/Sqrt[2]*(aaaa[[2,2,3,1]]+aaaa[[2,2,1,3]]),
 1/Sqrt[2]*(aaaa[[2,2,1,2]]+aaaa[[2,2,2,1]])},
 {aaaa[[3,3,1,1]],aaaa[[3,3,2,2]],aaaa[[3,3,3,3]],
 1/Sqrt[2]*(aaaa[[3,3,2,3]]+aaaa[[3,3,3,2]]),1/Sqrt[2]*(aaaa[[3,3,3,1]]+aaaa[[3,3,1,3]]),
 1/Sqrt[2]*(aaaa[[3,3,1,2]]+aaaa[[3,3,2,1]])},
{1/Sqrt[2]*(aaaa[[2,3,1,1]]+aaaa[[3,2,1,1]]),1/Sqrt[2]*(aaaa[[2,3,2,2]]+aaaa[[3,2,2,2]]),
 1/Sqrt[2]*(aaaa[[2,3,3,3]]+aaaa[[3,2,3,3]]),
1/2*(aaaa[[2,3,2,3]]+aaaa[[2,3,3,2]]+aaaa[[3,2,2,3]]+aaaa[[3,2,3,2]]),1/2*(aaaa[[2,3,3,1]]+aaaa[[2
{1/Sqrt[2]*(aaaa[[3,1,1,1]]+aaaa[[1,3,1,1]]),1/Sqrt[2]*(aaaa[[3,1,2,2]]+aaaa[[1,3,2,2]]),
 1/Sqrt[2]*(aaaa[[3,1,3,3]]+aaaa[[1,3,3,3]]),
1/2*(aaaa[[3,1,2,3]]+aaaa[[3,1,3,2]]+aaaa[[1,3,2,3]]+aaaa[[1,3,3,2]]),1/2*(aaaa[[3,1,3,1]]+aaaa[[3
{1/Sqrt[2]*(aaaa[[1,2,1,1]]+aaaa[[2,1,1,1]]),1/Sqrt[2]*(aaaa[[1,2,2,2]]+aaaa[[2,1,2,2]]),
 1/Sqrt[2]*(aaaa[[1,2,3,3]]+aaaa[[2,1,3,3]]),
1/2*(aaaa[[1,2,2,3]]+aaaa[[1,2,3,2]]+aaaa[[2,1,2,3]]+aaaa[[2,1,3,2]]),1/2*(aaaa[[1,2,3,1]]+aaaa[[1
```

Simple projection

```
pro4[aaaa_]:=\
{{aaaa[[1,1,1,1]],aaaa[[1,1,2,2]],aaaa[[1,1,3,3]],
 1/2*(aaaa[[1,1,2,3]]+aaaa[[1,1,3,2]]),1/2*(aaaa[[1,1,3,1]]+aaaa[[1,1,1,3]]),
 1/2*(aaaa[[1,1,1,2]]+aaaa[[1,1,2,1]])},
 {aaaa[[2,2,1,1]],aaaa[[2,2,2,2]],aaaa[[2,2,3,3]],
 1/2*(aaaa[[2,2,2,3]]+aaaa[[2,2,3,2]]),1/2*(aaaa[[2,2,3,1]]+aaaa[[2,2,1,3]]),
 1/2*(aaaa[[2,2,1,2]]+aaaa[[2,2,2,1]])},
 {aaaa[[3,3,1,1]],aaaa[[3,3,2,2]],aaaa[[3,3,3,3]],
 1/2*(aaaa[[3,3,2,3]]+aaaa[[3,3,3,2]]),1/2*(aaaa[[3,3,3,1]]+aaaa[[3,3,1,3]]),
 1/2*(aaaa[[3,3,1,2]]+aaaa[[3,3,2,1]])},
{1/2*(aaaa[[2,3,1,1]]+aaaa[[3,2,1,1]]),1/2*(aaaa[[2,3,2,2]]+aaaa[[3,2,2,2]]),
 1/2*(aaaa[[2,3,3,3]]+aaaa[[3,2,3,3]]),
1/4*(aaaa[[2,3,2,3]]+aaaa[[2,3,3,2]]+aaaa[[3,2,2,3]]+aaaa[[3,2,3,2]]),1/4*(aaaa[[2,3,3,1]]+aaaa[[2
{1/2*(aaaa[[3,1,1,1]]+aaaa[[1,3,1,1]]),1/2*(aaaa[[3,1,2,2]]+aaaa[[1,3,2,2]]),
 1/2*(aaaa[[3,1,3,3]]+aaaa[[1,3,3,3]]),
1/4*(aaaa[[3,1,2,3]]+aaaa[[3,1,3,2]]+aaaa[[1,3,2,3]]+aaaa[[1,3,3,2]]),1/4*(aaaa[[3,1,3,1]]+aaaa[[3
{1/2*(aaaa[[1,2,1,1]]+aaaa[[2,1,1,1]]),1/2*(aaaa[[1,2,2,2]]+aaaa[[2,1,2,2]]),
 1/2*(aaaa[[1,2,3,3]]+aaaa[[2,1,3,3]]),
1/4*(aaaa[[1,2,2,3]]+aaaa[[1,2,3,2]]+aaaa[[2,1,2,3]]+aaaa[[2,1,3,2]]),1/4*(aaaa[[1,2,3,1]]+aaaa[[1
```

Invariant expansion

```
blow4[aau_]:={{{{aau[[1,1]],aau[[1,6]]/Sqrt[2],aau[[1,5]]/Sqrt[2]},
{aau[[1,6]]/Sqrt[2],aau[[1,2]],aau[[1,4]]/Sqrt[2]},
{aau[[1,5]]/Sqrt[2],aau[[1,4]]/Sqrt[2],aau[[1,3]]}},
{{aau[[6,1]]/Sqrt[2],aau[[6,6]]/2,aau[[6,5]]/2},
{aau[[6,6]]/2,aau[[6,2]]/Sqrt[2],aau[[6,4]]/2},
{aau[[6,5]]/2,aau[[6,4]]/2,aau[[6,3]]/Sqrt[2]}},
{{aau[[5,1]]/Sqrt[2],aau[[5,6]]/2,aau[[5,5]]/2},
{aau[[5,6]]/2,aau[[5,2]]/Sqrt[2],aau[[5,4]]/2},
{aau[[5,5]]/2,aau[[5,4]]/2,aau[[5,3]]/Sqrt[2]}}},
{{{aau[[6,1]]/Sqrt[2],aau[[6,6]]/2,aau[[6,5]]/2},
{aau[[6,6]]/2,aau[[6,2]]/Sqrt[2],aau[[6,4]]/2},
{aau[[6,5]]/2,aau[[6,4]]/2,aau[[6,3]]/Sqrt[2]}},
{{aau[[2,1]],aau[[2,6]]/Sqrt[2],aau[[2,5]]/Sqrt[2]},
{aau[[2,6]]/Sqrt[2],aau[[2,2]],aau[[2,4]]/Sqrt[2]},
{aau[[2,5]]/Sqrt[2],aau[[2,4]]/Sqrt[2],aau[[2,3]]}},
{{aau[[4,1]]/Sqrt[2],aau[[4,6]]/2,aau[[4,5]]/2},
{aau[[4,6]]/2,aau[[4,2]]/Sqrt[2],aau[[4,4]]/2},
{aau[[4,5]]/2,aau[[4,4]]/2,aau[[4,3]]/Sqrt[2]}}},
{{{aau[[5,1]]/Sqrt[2],aau[[5,6]]/2,aau[[5,5]]/2},
{aau[[5,6]]/2,aau[[5,2]]/Sqrt[2],aau[[5,4]]/2},
{aau[[5,5]]/2,aau[[5,4]]/2,aau[[5,3]]/Sqrt[2]}},
{{aau[[4,1]]/Sqrt[2],aau[[4,6]]/2,aau[[4,5]]/2},
{aau[[4,6]]/2,aau[[4,2]]/Sqrt[2],aau[[4,4]]/2},
{aau[[4,5]]/2,aau[[4,4]]/2,aau[[4,3]]/Sqrt[2]}},
{{aau[[3,1]],aau[[3,6]]/Sqrt[2],aau[[3,5]]/Sqrt[2]},
{aau[[3,6]]/Sqrt[2],aau[[3,2]],aau[[3,4]]/Sqrt[2]},
{aau[[3,5]]/Sqrt[2],aau[[3,4]]/Sqrt[2],aau[[3,3]]}}}}
```

Simple extension

```
blo4[aau_]:={{{{aau[[1,1]],aau[[1,6]],aau[[1,5]]},
{aau[[1,6]],aau[[1,2]],aau[[1,4]]},
{aau[[1,5]],aau[[1,4]],aau[[1,3]]}},
{{aau[[6,1]],aau[[6,6]],aau[[6,5]]},
{aau[[6,6]],aau[[6,2]],aau[[6,4]]},
{aau[[6,5]],aau[[6,4]],aau[[6,3]]}},
{{aau[[5,1]],aau[[5,6]],aau[[5,5]]},
{aau[[5,6]],aau[[5,2]],aau[[5,4]]},
{aau[[5,5]],aau[[5,4]],aau[[5,3]]}}},
{{{aau[[6,1]],aau[[6,6]],aau[[6,5]]},
{aau[[6,6]],aau[[6,2]],aau[[6,4]]},
{aau[[6,5]],aau[[6,4]],aau[[6,3]]}},
{{aau[[2,1]],aau[[2,6]],aau[[2,5]]},
{aau[[2,6]],aau[[2,2]],aau[[2,4]]},
{aau[[2,5]],aau[[2,4]],aau[[2,3]]}},
{{aau[[4,1]],aau[[4,6]],aau[[4,5]]},
{aau[[4,6]],aau[[4,2]],aau[[4,4]]},
{aau[[4,5]],aau[[4,4]],aau[[4,3]]}}},
{{{aau[[5,1]],aau[[5,6]],aau[[5,5]]},
{aau[[5,6]],aau[[5,2]],aau[[5,4]]},
{aau[[5,5]],aau[[5,4]],aau[[5,3]]}},
{{aau[[4,1]],aau[[4,6]],aau[[4,5]]},
{aau[[4,6]],aau[[4,2]],aau[[4,4]]},
{aau[[4,5]],aau[[4,4]],aau[[4,3]]}},
{{aau[[3,1]],aau[[3,6]],aau[[3,5]]},
{aau[[3,6]],aau[[3,2]],aau[[3,4]]},
{aau[[3,5]],aau[[3,4]],aau[[3,3]]}}}}
```

Completely symmetric fourth order tensors

Variables

```
eeeecst=
{{{{e1111,e1112,e1113},{e1112,e1122,e1123},{e1113,e1123,e1133}},
  {{e1112,e1122,e1123},{e1122,e1222,e1223},{e1123,e1223,e1233}},
  {{e1113,e1123,e1133},{e1123,e1223,e1233},{e1133,e1233,e1333}}},
 {{{e1112,e1122,e1123},{e1122,e1222,e1223},{e1123,e1223,e1233}},
  {{e1122,e1222,e1223},{e1222,e2222,e2223},{e1223,e2223,e2233}},
  {{e1123,e1223,e1233},{e1223,e2223,e2233},{e1233,e2233,e2333}}},
 {{{e1113,e1123,e1133},{e1123,e1223,e1233},{e1133,e1233,e1333}},
  {{e1123,e1223,e1233},{e1223,e2223,e2233},{e1233,e2233,e2333}},
  {{e1133,e1233,e1333},{e1233,e2233,e2333},{e1333,e2333,e3333}}}};
```

Addition

Neutral tensor

```
ne4=\
{{{{0,0,0},{0,0,0},{0,0,0}},
  {{0,0,0},{0,0,0},{0,0,0}},
  {{0,0,0},{0,0,0},{0,0,0}}},
 {{{0,0,0},{0,0,0},{0,0,0}},
  {{0,0,0},{0,0,0},{0,0,0}},
  {{0,0,0},{0,0,0},{0,0,0}}},
 {{{0,0,0},{0,0,0},{0,0,0}},
  {{0,0,0},{0,0,0},{0,0,0}},
  {{0,0,0},{0,0,0},{0,0,0}}}};
```

Internal product

```
t442[aaaa_,bbbb_]:=\
Sum[Transpose[Inner[Times,aaaa,Transpose[bbbb],Plus],
{3,4,1,2,5,6}][[i,i]],{i,3}]
```

Linear transformations of second rank tensors

```
t422[aaaa_,bb_]:=\
Sum[Transpose[Inner[Times,aaaa,Transpose[bb],Plus],
{3,4,1,2}][[i,i]],{i,3}]
```

Identity and special tensors

Rank three identity tensor

```
id4 = tund[id2, id2];
```

Transposing tensor

```
tr4 = tove[id2, id2];
```

Rank one traceor tensor

```
tc4 = tcro[id2, id2];
```

Symetric identity tensor

```
id4sym := tdou[id2, id2]
```

Deviator tensor extractor

```
dc4 := id4 - 1 / 3 * tcro[id2, id2]
```

Symmetric deviator tensor extractor

```
dc4sym := id4sym - 1 / 3 * tcro[id2, id2]
```

Scalar product, distance and norm

```
t444[aaaa_,bbbb_]:=\
Sum[Sum[Transpose[Sum[Transpose[Inner[Times,aaaa,
Transpose[bbbb,{4,2,3,1}],Plus],{1,6,3,4,5,2}]
[[i,i]],{i,3}],{1,3,2,4}][[j,j]],{j,3}][[k,k]],{k,3}]
```

```
dist4[aaaa_,bbbb_]:=Sqrt[t444[aaaa-bbbb,aaaa-bbbb]]
```

```
norm4[aaaa_]:=Sqrt[t444[aaaa,aaaa]]
```

Unitary transformations

Rotation of a fourth rank tensor

```
t442[t442[Transpose[tund[rr[ξ,ψ,ζ],rr[ξ,ψ,ζ]],{3,4,1,2}],xxxx],tund[rr[ξ,ψ,ζ],rr[ξ,ψ,ζ]]];
```

Decomposition of symmetric fourth rank tensor

Definitions

```
reuss[aaaa_] := t422[aaaa, id2]
```

```
voigt[aaaa_] := t422[Transpose[aaaa, {1, 3, 2, 4}], id2]
```

Fourth order deviator of a fully traceless tensor (Spencer, 1970)

```
dev4fs[eeee_]:=eeee-1/7*(tcro[id2,reuss[eeee]]+tcro[reuss[eeee],id2]+
                              2*(tdou[id2,reuss[eeee]]+tdou[reuss[eeee],id2]))+
                1/35*Tr[reuss[eeee]]*(tcro[id2,id2]+2*tdou[id2,id2])
```

```
Simplify[t422[dev4fs[eeee],id2]]
```

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

Fourth order decomposition (Cowin, 1989)

```
dec4alpha[aaaa_] := 1 / 15 * (2 * Tr[reuss[aaaa]] - Tr[voigt[aaaa]])
```

```
dec4beta[aaaa_] := 1 / 30 * (3 * Tr[voigt[aaaa]] - Tr[reuss[aaaa]])
```

```
dec4aa[aaaa_] := 5 / 7 * dev2[reuss[aaaa]] - 4 / 7 * dev2[voigt[aaaa]]
```

```
dec4bb[aaaa_] := 3 / 7 * dev2[voigt[aaaa]] - 2 / 7 * dev2[reuss[aaaa]]
```

```
dev4[aaaa_] :=
 aaaa - dec4alpha[aaaa] * tc4 - 2 * dec4beta[aaaa] * id4sym - tcro[id2, dec4aa[aaaa]] -
  tcro[dec4aa[aaaa], id2] - 2 * tdou[id2, dec4bb[aaaa]] - 2 * tdou[dec4bb[aaaa], id2]
```

Acoustic tensor

```
acousten[ssss_, n_] := t422[Transpose[ssss, {1, 3, 2, 4}], tcro[n, n]]
```

Fourth order tensor products

Major products

```
tsund[aaaa_, bbbb_] := Transpose[Outer[Times, aaaa, bbbb], {1, 5, 2, 6, 3, 7, 4, 8}]
```

```
tsove[aaaa_, bbbb_] := Transpose[Outer[Times, aaaa, bbbb], {1, 7, 2, 8, 3, 5, 4, 6}]
```

Minor products, first term left

```
t12354678[aaaa_, bbbb_] := Transpose[Outer[Times, aaaa, bbbb], {1, 2, 3, 5, 4, 6, 7, 8}]
```

```
t12365478[aaaa_, bbbb_] := Transpose[Outer[Times, aaaa, bbbb], {1, 2, 3, 6, 4, 5, 7, 8}]
```

```
t12375648[aaaa_, bbbb_] := Transpose[Outer[Times, aaaa, bbbb], {1, 2, 3, 7, 5, 6, 4, 8}]
```

Minor products, first term right

```
t15342678[aaaa_, bbbb_] := Transpose[Outer[Times, aaaa, bbbb], {1, 5, 3, 4, 2, 6, 7, 8}]
```

```
t16345278[aaaa_, bbbb_] := Transpose[Outer[Times, aaaa, bbbb], {1, 6, 3, 4, 2, 5, 7, 8}]
```

```
t17345628[aaaa_, bbbb_] := Transpose[Outer[Times, aaaa, bbbb], {1, 7, 3, 4, 5, 6, 2, 8}]
```

Minor products, second term left

```
t13645278[aaaa_, bbbb_] := Transpose[Outer[Times, aaaa, bbbb], {1, 3, 6, 4, 5, 2, 7, 8}]
```

```
t13542678[aaaa_, bbbb_] := Transpose[Outer[Times, aaaa, bbbb], {1, 3, 5, 4, 2, 6, 7, 8}]
```

```
t13845672[aaaa_, bbbb_] := Transpose[Outer[Times, aaaa, bbbb], {1, 3, 8, 4, 5, 6, 7, 2}]

t14365278[aaaa_, bbbb_] := Transpose[Outer[Times, aaaa, bbbb], {1, 4, 3, 6, 5, 2, 7, 8}]

t14352678[aaaa_, bbbb_] := Transpose[Outer[Times, aaaa, bbbb], {1, 4, 3, 5, 2, 6, 7, 8}]

t14385672[aaaa_, bbbb_] := Transpose[Outer[Times, aaaa, bbbb], {1, 4, 3, 8, 5, 6, 7, 2}]
```

Minor products, second term right

```
t15243678[aaaa_, bbbb_] := Transpose[Outer[Times, aaaa, bbbb], {1, 5, 2, 4, 3, 6, 7, 8}]

t15324678[aaaa_, bbbb_] := Transpose[Outer[Times, aaaa, bbbb], {1, 5, 3, 2, 4, 6, 7, 8}]

t16245378[aaaa_, bbbb_] := Transpose[Outer[Times, aaaa, bbbb], {1, 6, 2, 4, 5, 3, 7, 8}]

t16325478[aaaa_, bbbb_] := Transpose[Outer[Times, aaaa, bbbb], {1, 6, 3, 2, 5, 4, 7, 8}]

t17245638[aaaa_, bbbb_] := Transpose[Outer[Times, aaaa, bbbb], {1, 7, 2, 4, 5, 6, 3, 8}]

t17325648[aaaa_, bbbb_] := Transpose[Outer[Times, aaaa, bbbb], {1, 7, 3, 2, 5, 6, 4, 8}]
```

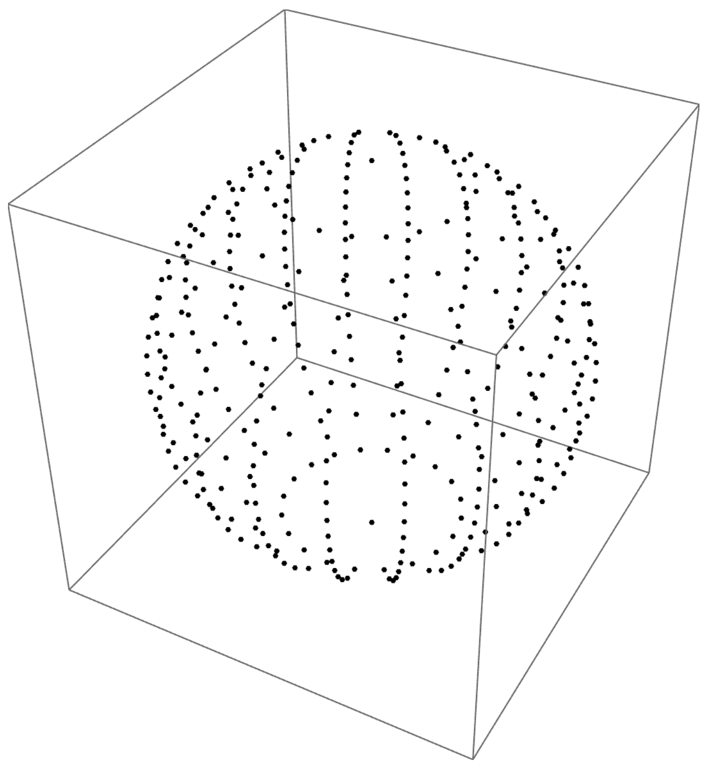
MIL

Definitions

Essentials

```
e1 = {1, 0, 0}; e2 = {0, 1, 0}; e3 = {0, 0, 1};

a = {Sin[theta] * Cos[phi], Sin[theta] * Sin[phi], Cos[theta]};
```

```
Graphics3D[
 Point[Flatten[Table[{Sqrt[1 - zeta ^ 2] * Cos[phi], Sqrt[1 - zeta ^ 2] * Sin[phi], zeta},
    {zeta, -1, 1, 2 / 20}, {phi, 0, 2 Pi, 2 Pi / 20}], 1]]]
```



```
rr[α_, β_, γ_] := \
{{Cos[α] Cos[β] Cos[γ] - Sin[α] Sin[γ],
 -Cos[α] Cos[β] Sin[γ] - Sin[α] Cos[γ],
  Cos[α] Sin[β]},
 {Sin[α] Cos[β] Cos[γ] + Cos[α] Sin[γ],
 -Sin[α] Cos[β] Sin[γ] + Cos[α] Cos[γ],
  Sin[α] Sin[β]},
  {-Sin[β] Cos[γ], Sin[β] Sin[γ], Cos[β]}}
```

Basis functions of spherical harmonics

```
ffsphh[a_] := tcro[a, a] - 1 / 3 * id2;
```

```
ffffsphh[a_] := tcro[tcro[a, a], tcro[a, a]] - 1 / 7 * (tcro[tcro[a, a], id2] +
tcro[id2, tcro[a, a]] + tund[tcro[a, a], id2] + tund[id2, tcro[a, a]] +
tove[tcro[a, a], id2] + tove[id2, tcro[a, a]]) + 1 / 35 * (tcro[id2, id2] +
tund[id2, id2] + tove[id2, id2]);
```

Check

Example of a hexahedral bar with dimensions 1x1x5mm

```
nface = 6;
```

```
normals = {e3, -e3, e1, -e1, e2, -e2};
```

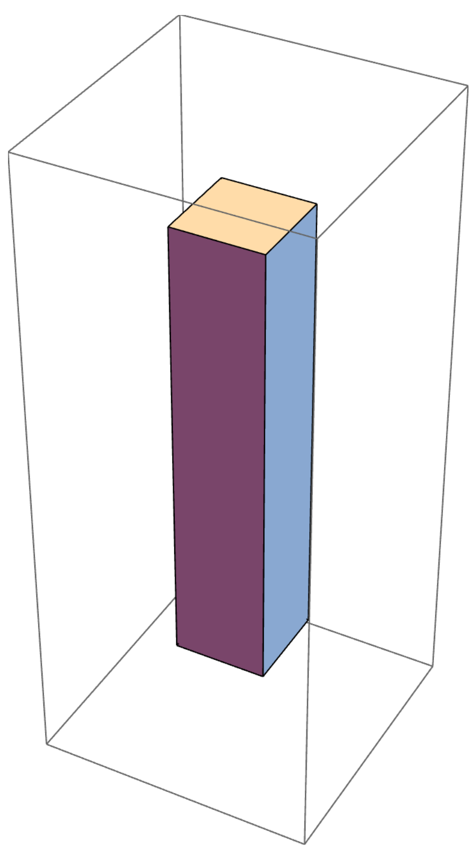
```
areas = {1, 1, 5, 5, 5, 5};
```

```
volume = 5.0;
```

```
area = Sum[areas[[i]], {i, 1, nface}]
```

```
22
```

```
Graphics3D[Hexahedron[
   {{0, 0, 0}, {1, 0, 0}, {1, 1, 0}, {0, 1, 0}, {0, 0, 5}, {1, 0, 5}, {1, 1, 5}, {0, 1, 5}}],
  PlotRange -> {{-1, 2}, {-1, 2}, {-1, 6}}]
```



```
mil[a_] :=
  2 * volume / Sum[areas[[i]] * Sqrt[a.Outer[Times, normals[[i]], normals[[i]]].a], {i, 1, nface}]
```

```
mil[e1]
```

1.

```
mil[e2]
```

1.

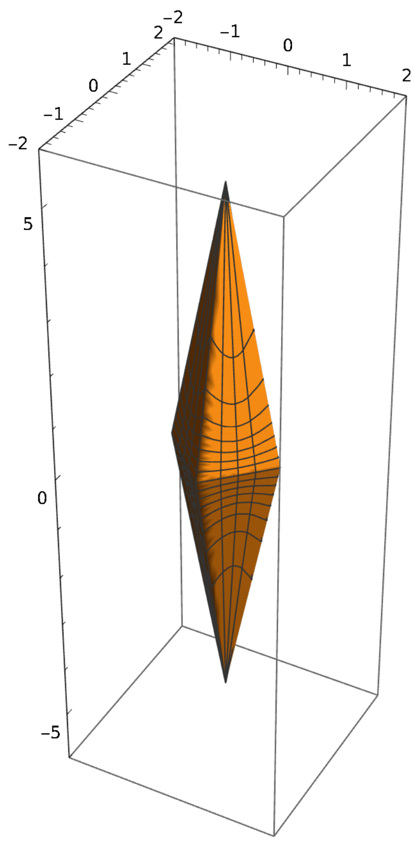
```
mil[e3]
```

5.

```
daratiomil = N[mil[e3]/mil[e1]]
```

5.

```
ParametricPlot3D [mil[a] * a, {theta , 0, Pi},
 {phi , 0, 2 Pi}, PlotRange → {{-2, 2}, {-2, 2}, {-6, 6}}]
```



TRIFAB

```
fabricinv2approx =
  Sum[areas[[i]]/2/volume * Outer[Times , normals[[i]], normals[[i]]], {i, 1, nface}].
   Sum[areas[[i]]/2/volume * Outer[Times , normals[[i]], normals[[i]]], {i, 1, nface}];

trf[a_] := 1/Sqrt[a.fabricinv2approx .a]
```

```
N[trf[e1]]
```

1.

```
N[trf[e2]]
```

1.

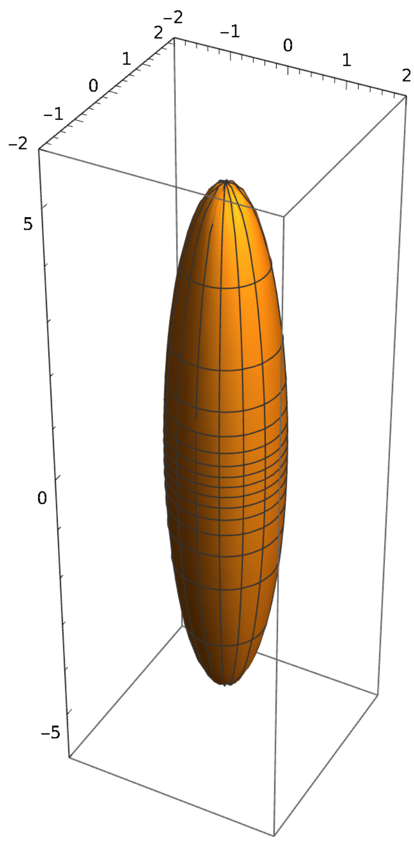
```
N[trf[e3]]
```

5.

```
daratiotrf = N[trf[e3] / trf[e1]]
```

5.

```
ParametricPlot3D [trf[a] * a, {theta , 0, Pi},
 {phi , 0, 2 Pi}, PlotRange → {{-2, 2}, {-2, 2}, {-6, 6}}]
```



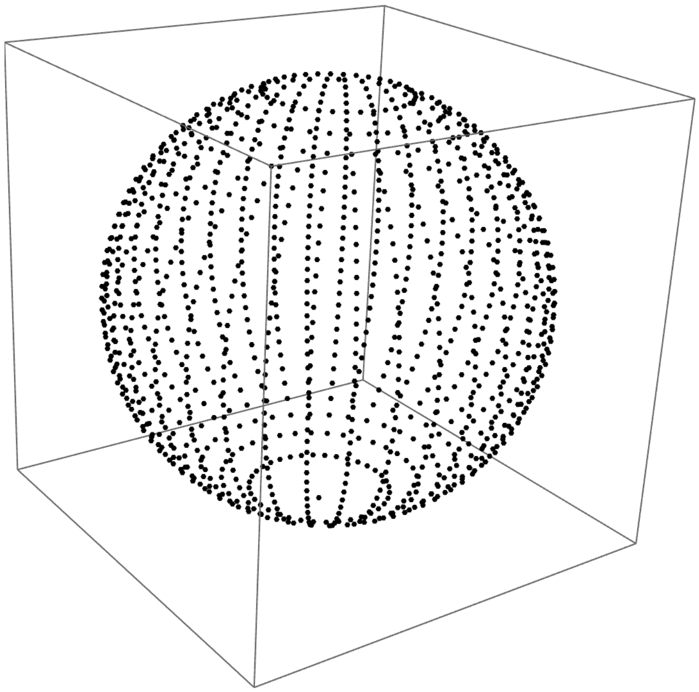
Fit of the original MIL to an ellispsoid

```
fabric[{m1_, m2_, m3_, alpha_, beta_, gamma_}] :=
 m1 * rr[alpha, beta, gamma].Outer[Times, e1, e1].Transpose[rr[alpha, beta, gamma]] +
  m2 * rr[alpha, beta, gamma].Outer[Times, e2, e2].Transpose[rr[alpha, beta, gamma]] +
  m3 * rr[alpha, beta, gamma].Outer[Times, e3, e3].Transpose[rr[alpha, beta, gamma]]
```

```
abar = {Sqrt[1 - zeta ^ 2] * Cos[phi], Sqrt[1 - zeta ^ 2] * Sin[phi], zeta}
```

$\left\{\sqrt{1-\text{zeta}^2}\ \text{Cos[phi]},\ \sqrt{1-\text{zeta}^2}\ \text{Sin[phi]},\ \text{zeta}\right\}$

```
Graphics3D[Point[Flatten[Table[abar, {zeta, -1, 1, 2 / 36}, {phi, 0, 2 Pi, 2 Pi / 36}], 1]]]
```



```
NMinimize[
 {Sum[(mil[abar] - 1 / Sqrt[(abar.fabric[{1 / m1 ^ 2, 1 / m1 ^ 2, 1 / m3 ^ 2, 0, 0, 0}].abar)]) ^ 2,
   {zeta, -1, 1, 2 / 72}, {phi, 0, 2 Pi, 2 Pi / 72}], m1 > 0, m3 > 0}, {m1, m3}]
```

{117.1, {m1 → 0.668848, m3 → 4.92845}}

```
fabricarg = {1 / m1 ^ 2, 1 / m1 ^ 2, 1 / m3 ^ 2, 0, 0, 0} /. %[[2]]
```

{2.23535, 2.23535, 0.0411698, 0, 0, 0}

```
Eigensystem[fabric[fabricarg]]
```

{{2.23535, 2.23535, 0.0411698}, {{0., 1., 0.}, {1., 0., 0.}, {0., 0., 1.}}}

```
milapprox[a_] := 1 / Sqrt[a.fabric[fabricarg].a]
```

```
milapprox[e1]
```

0.668848

```
milapprox[e2]
```

0.668848

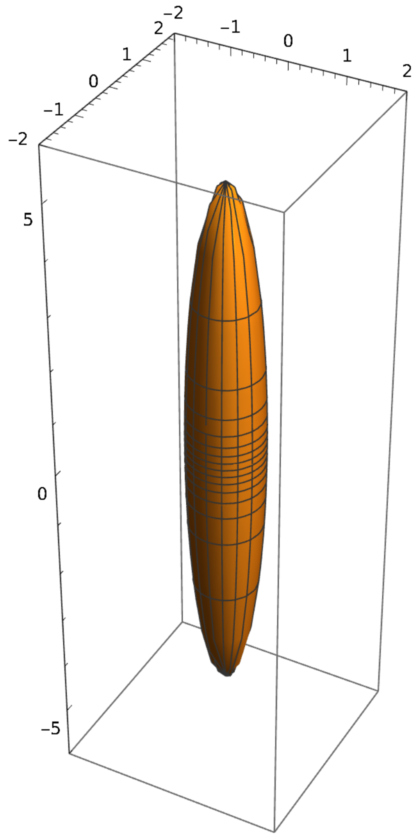
```
milapprox[e3]
```

4.92845

```
daratiomilapprox = milapprox[e3]/milapprox[e1]
```

7.36856

```
ParametricPlot3D[milapprox[a]*a, {theta, 0, Pi},
 {phi, 0, 2 Pi}, PlotRange → {{-2, 2}, {-2, 2}, {-6, 6}}]
```



Decomposition of MIL in spherical harmonics

Coefficients

```
gnum = 1/(4 Pi)*
  NIntegrate[mil[{Sin[theta]*Cos[phi], Sin[theta]*Sin[phi], Cos[theta]}]*Sin[theta],
   {theta, 0, Pi}, {phi, 0, 2 Pi}]
```

0.988552

```
ggnum = 15 / (8 Pi) * NIntegrate [mil[{Sin[theta] * Cos[phi], Sin[theta] * Sin[phi], Cos[theta]}] *
    ffsphh[{Sin[theta] * Cos[phi], Sin[theta] * Sin[phi], Cos[theta]}] *
    Sin[theta], {theta , 0, Pi}, {phi , 0, 2 Pi}]
```

NIntegrate ::slwcon : Numerical integration converging too slowly ; suspect one of the following : singularity , value of the integration is 0, highly oscillatory integrand , or WorkingPrecision too small . ≫

NIntegrate ::eincr : The global error of the strategy GlobalAdaptive has increased more than 2000 times . The global error is expected to decrease monotonically after a number of integrand evaluations . Suspect one of the following : the working precision is insufficient for the specified precision goal ; the integrand is highly oscillatory or it is not a (piecewise) smooth function ; or the true value of the integral is 0. Increasing the value of the GlobalAdaptive option MaxErrorIncreases might lead to a convergent numerical integration . NIntegrate obtained 2.916503843986007`*^-17 and 1.389584655517477`*^-12 for the integral and error estimates . ≫

NIntegrate ::slwcon : Numerical integration converging too slowly ; suspect one of the following : singularity , value of the integration is 0, highly oscillatory integrand , or WorkingPrecision too small . ≫

NIntegrate ::eincr : The global error of the strategy GlobalAdaptive has increased more than 2000 times . The global error is expected to decrease monotonically after a number of integrand evaluations . Suspect one of the following : the working precision is insufficient for the specified precision goal ; the integrand is highly oscillatory or it is not a (piecewise) smooth function ; or the true value of the integral is 0. Increasing the value of the GlobalAdaptive option MaxErrorIncreases might lead to a convergent numerical integration . NIntegrate obtained 1.179069862577986`*^-16 and 7.394499717291283`*^-13 for the integral and error estimates . ≫

NIntegrate ::slwcon : Numerical integration converging too slowly ; suspect one of the following : singularity , value of the integration is 0, highly oscillatory integrand , or WorkingPrecision too small . ≫

General ::stop : Further output of NIntegrate ::slwcon will be suppressed during this calculation . ≫

NIntegrate ::eincr : The global error of the strategy GlobalAdaptive has increased more than 2000 times . The global error is expected to decrease monotonically after a number of integrand evaluations . Suspect one of the following : the working precision is insufficient for the specified precision goal ; the integrand is highly oscillatory or it is not a (piecewise) smooth function ; or the true value of the integral is 0. Increasing the value of the GlobalAdaptive option MaxErrorIncreases might lead to a convergent numerical integration . NIntegrate obtained 2.916503843986007`*^-17 and 1.389584655517477`*^-12 for the integral and error estimates . ≫

General ::stop : Further output of NIntegrate ::eincr will be suppressed during this calculation . ≫

$$\{\{-0.33242, 1.74066\times10^{-17}, 7.03705\times10^{-17}\}, \{1.74066\times10^{-17}, -0.33242, -1.34578\times10^{-16}\}, \{7.03705\times10^{-17}, -1.34578\times10^{-16}, 0.664841\}\}$$

```
ggggnum =
  315 / (32 Pi) * NIntegrate [mil[{Sin[theta] * Cos[phi], Sin[theta] * Sin[phi], Cos[theta]}] *
    ffffsphh[{Sin[theta] * Cos[phi], Sin[theta] * Sin[phi], Cos[theta]}] *
    Sin[theta], {theta , 0, Pi}, {phi , 0, 2 Pi}]
```

NIntegrate ::slwcon : Numerical integration converging too slowly ; suspect one of the following : singularity , value of the integration is 0, highly oscillatory integrand , or WorkingPrecision too small . ≫

NIntegrate::eincr : The global error of the strategy GlobalAdaptive has increased more than 2000 times. The global error is expected to decrease monotonically after a number of integrand evaluations. Suspect one of the following: the working precision is insufficient for the specified precision goal; the integrand is highly oscillatory or it is not a (piecewise) smooth function; or the true value of the integral is 0. Increasing the value of the GlobalAdaptive option MaxErrorIncreases might lead to a convergent numerical integration. NIntegrate obtained -1.4137×10^{-18} and 4.262691713201152`*^-13 for the integral and error estimates. ≫

NIntegrate::slwcon : Numerical integration converging too slowly; suspect one of the following: singularity, value of the integration is 0, highly oscillatory integrand, or WorkingPrecision too small. ≫

NIntegrate::eincr : The global error of the strategy GlobalAdaptive has increased more than 2000 times. The global error is expected to decrease monotonically after a number of integrand evaluations. Suspect one of the following: the working precision is insufficient for the specified precision goal; the integrand is highly oscillatory or it is not a (piecewise) smooth function; or the true value of the integral is 0. Increasing the value of the GlobalAdaptive option MaxErrorIncreases might lead to a convergent numerical integration. NIntegrate obtained -1.26377×10^{-18} and 3.7291239231520534`*^-13 for the integral and error estimates. ≫

NIntegrate::slwcon : Numerical integration converging too slowly; suspect one of the following: singularity, value of the integration is 0, highly oscillatory integrand, or WorkingPrecision too small. ≫

General::stop : Further output of NIntegrate::slwcon will be suppressed during this calculation. ≫

NIntegrate::eincr : The global error of the strategy GlobalAdaptive has increased more than 2000 times. The global error is expected to decrease monotonically after a number of integrand evaluations. Suspect one of the following: the working precision is insufficient for the specified precision goal; the integrand is highly oscillatory or it is not a (piecewise) smooth function; or the true value of the integral is 0. Increasing the value of the GlobalAdaptive option MaxErrorIncreases might lead to a convergent numerical integration. NIntegrate obtained -1.4137×10^{-18} and 4.262691713201152`*^-13 for the integral and error estimates. ≫

General::stop : Further output of NIntegrate::eincr will be suppressed during this calculation. ≫

$$\{\{\{\{0.334723, -4.42963\times10^{-18}, -3.95986\times10^{-18}\}, \{-4.42963\times10^{-18}, -0.0772198, -9.95804\times10^{-18}\}, \{-3.95986\times10^{-18}, -9.95804\times10^{-18}, -0.257503\}\},$$
$$\{\{-4.42963\times10^{-18}, -0.0772198, -9.95804\times10^{-18}\}, \{-0.0772198, 1.96719\times10^{-17}, -2.46251\times10^{-12}\}, \{-9.95804\times10^{-18}, -2.46251\times10^{-12}, -2.08503\times10^{-17}\}\},$$
$$\{\{-3.95986\times10^{-18}, -9.95804\times10^{-18}, -0.257503\}, \{-9.95804\times10^{-18}, -2.46251\times10^{-12}, -2.08503\times10^{-17}\}, \{-0.257503, -2.08503\times10^{-17}, -2.15138\times10^{-17}\}\}\},$$
$$\{\{\{-4.42963\times10^{-18}, -0.0772198, -9.95804\times10^{-18}\}, \{-0.0772198, 1.96719\times10^{-17}, -2.46251\times10^{-12}\}, \{-9.95804\times10^{-18}, -2.46251\times10^{-12}, -2.08503\times10^{-17}\}\},$$
$$\{\{-0.0772198, 1.96719\times10^{-17}, -2.46251\times10^{-12}\}, \{1.96719\times10^{-17}, 0.334723, -8.27059\times10^{-17}\}, \{-2.46251\times10^{-12}, -8.27059\times10^{-17}, -0.257503\}\},$$
$$\{\{-9.95804\times10^{-18}, -2.46251\times10^{-12}, -2.08503\times10^{-17}\}, \{-2.46251\times10^{-12}, -8.27059\times10^{-17}, -0.257503\}, \{-2.08503\times10^{-17}, -0.257503, -4.08733\times10^{-17}\}\}\},$$
$$\{\{\{-3.95986\times10^{-18}, -9.95804\times10^{-18}, -0.257503\}, \{-9.95804\times10^{-18}, -2.46251\times10^{-12}, -2.08503\times10^{-17}\}, \{-0.257503, -2.08503\times10^{-17}, -2.15138\times10^{-17}\}\},$$
$$\{\{-9.95804\times10^{-18}, -2.46251\times10^{-12}, -2.08503\times10^{-17}\}, \{-2.46251\times10^{-12}, -8.27059\times10^{-17}, -0.257503\}, \{-2.08503\times10^{-17}, -0.257503, -4.08733\times10^{-17}\}\},$$
$$\{\{-0.257503, -2.08503\times10^{-17}, -2.15138\times10^{-17}\}, \{-2.08503\times10^{-17}, -0.257503, -4.08733\times10^{-17}\}, \{-2.15138\times10^{-17}, -4.08733\times10^{-17}, 0.515007\}\}\}\}$$

Estimation 2nd order

```
milsphh2[n_] := gnum + t222[ggnum, ffsphh[n]];
```

```
milsphh2[e1]
```

0.656132

```
milsphh2[e2]
```

0.656132

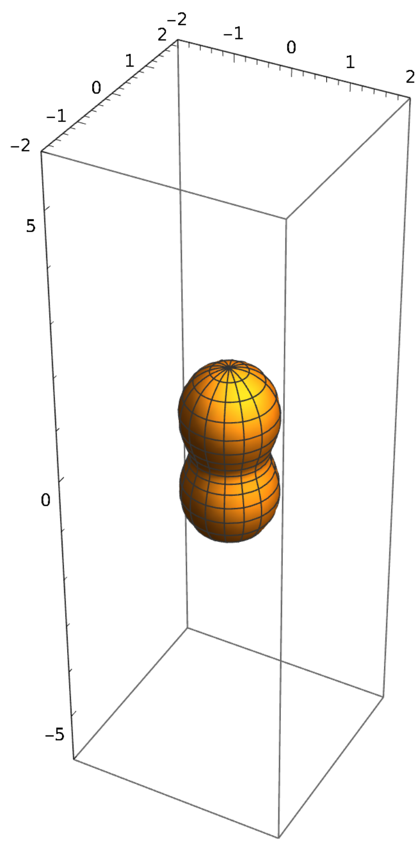
```
milsphh2[e3]
```

1.65339

```
daratiosphh2 = milsphh2[e3]/milsphh2[e1]
```

2.51991

```
ParametricPlot3D[milsphh2[n] * n, {t, 0, Pi},
 {u, 0, 2 Pi}, PlotRange → {{-2, 2}, {-2, 2}, {-6, 6}}]
```



Estimation 4th order

```
milsphh4[n_] := gnum + t222[ggnum, ffsphh[n]] + t444[ggggnum, ffffsphh[n]];
```

```
N[milsphh4[e1]]
```

0.990855

```
N[milsphh4[e2]]
```

0.990855

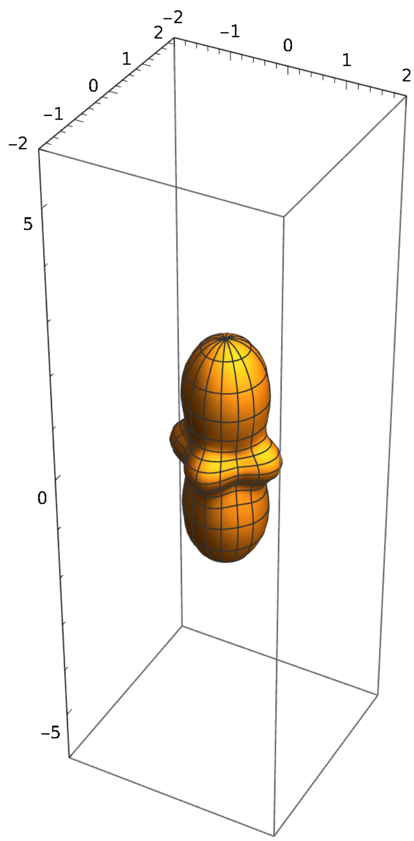
```
N[milsphh4[e3]]
```

2.1684

```
daratiosphh4 = milsphh4[e3]/milsphh4[e1]
```

2.18841

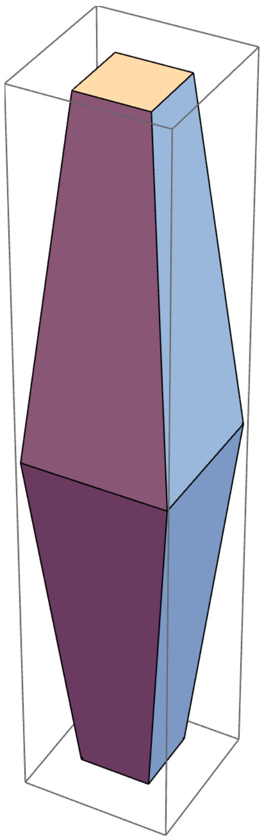
```
ParametricPlot3D[milsphh4[n]*n, {t, 0, Pi},
 {u, 0, 2 Pi}, PlotRange → {{-2, 2}, {-2, 2}, {-6, 6}}]
```



Example of a truncated rhombus with dimensions 1x1x5mm

```
nface = 10;
```

```
Graphics3D[{Hexahedron[{{0, 0, 0}, {1, 0, 0}, {1, 1, 0}, {0, 1, 0},
    {1/4, 1/4, 5/2}, {3/4, 1/4, 5/2}, {3/4, 3/4, 5/2}, {1/4, 3/4, 5/2}}],
  Hexahedron[{{0, 0, 0}, {1, 0, 0}, {1, 1, 0}, {0, 1, 0}, {1/4, 1/4, -5/2},
    {3/4, 1/4, -5/2}, {3/4, 3/4, -5/2}, {1/4, 3/4, -5/2}}]}]
```



```
volume = N[2 * (1/3 * 1 * 5 - 1/3 * 1/4 * 5/2)]
```

2.91667

```
normals = N[{-e3, -Cross[{1, 0, 0}, {1/4, 1/4, 5/2}] / Sqrt[{1/4, 1/4, 5/2}.{1/4, 1/4, 5/2}],
   Cross[{0, 1, 0}, {1/4, 1/4, 5/2}] / Sqrt[{1/4, 1/4, 5/2}.{1/4, 1/4, 5/2}],
   Cross[{1, 0, 0}, {-1/4, -1/4, 5/2}] / Sqrt[{-1/4, -1/4, 5/2}.{-1/4, -1/4, 5/2}],
   -Cross[{0, 1, 0}, {-1/4, -1/4, 5/2}] / Sqrt[{-1/4, -1/4, 5/2}.{-1/4, -1/4, 5/2}],
   Cross[{1, 0, 0}, {1/4, 1/4, 5/2}] / Sqrt[{1/4, 1/4, 5/2}.{1/4, 1/4, 5/2}],
   -Cross[{0, 1, 0}, {1/4, 1/4, 5/2}] / Sqrt[{1/4, 1/4, 5/2}.{1/4, 1/4, 5/2}],
   -Cross[{1, 0, 0}, {-1/4, -1/4, 5/2}] / Sqrt[{-1/4, -1/4, 5/2}.{-1/4, -1/4, 5/2}],
   Cross[{0, 1, 0}, {-1/4, -1/4, 5/2}] / Sqrt[{-1/4, -1/4, 5/2}.{-1/4, -1/4, 5/2}], e3}]
```

```
{{0., 0., -1.}, {0., 0.990148, -0.0990148}, {0.990148, 0., -0.0990148},
 {0., -0.990148, -0.0990148}, {-0.990148, 0., -0.0990148},
 {0., -0.990148, 0.0990148}, {-0.990148, 0., 0.0990148},
 {0., 0.990148, 0.0990148}, {0.990148, 0., 0.0990148}, {0., 0., 1.}}
```

```
areas = N[{1/4, 1/2*Sqrt[Cross[{1, 0, 0}, {1/4, 1/4, 5}].Cross[{1, 0, 0}, {1/4, 1/4, 5}]] -
    1/2*Sqrt[Cross[{1/2, 0, 0}, {0, 0, 5/2}].Cross[{1/2, 0, 0}, {0, 0, 5/2}]],
   1/2*Sqrt[Cross[{1, 0, 0}, {1/4, 1/4, 5}].Cross[{1, 0, 0}, {1/4, 1/4, 5}]] -
    1/2*Sqrt[Cross[{1/2, 0, 0}, {0, 0, 5/2}].Cross[{1/2, 0, 0}, {0, 0, 5/2}]],
   1/2*Sqrt[Cross[{1, 0, 0}, {1/4, 1/4, 5}].Cross[{1, 0, 0}, {1/4, 1/4, 5}]] -
    1/2*Sqrt[Cross[{1/2, 0, 0}, {0, 0, 5/2}].Cross[{1/2, 0, 0}, {0, 0, 5/2}]],
   1/2*Sqrt[Cross[{1, 0, 0}, {1/4, 1/4, 5}].Cross[{1, 0, 0}, {1/4, 1/4, 5}]] -
    1/2*Sqrt[Cross[{1/2, 0, 0}, {0, 0, 5/2}].Cross[{1/2, 0, 0}, {0, 0, 5/2}]],
   1/2*Sqrt[Cross[{1, 0, 0}, {1/4, 1/4, 5}].Cross[{1, 0, 0}, {1/4, 1/4, 5}]] -
    1/2*Sqrt[Cross[{1/2, 0, 0}, {0, 0, 5/2}].Cross[{1/2, 0, 0}, {0, 0, 5/2}]],
   1/2*Sqrt[Cross[{1, 0, 0}, {1/4, 1/4, 5}].Cross[{1, 0, 0}, {1/4, 1/4, 5}]] -
    1/2*Sqrt[Cross[{1/2, 0, 0}, {0, 0, 5/2}].Cross[{1/2, 0, 0}, {0, 0, 5/2}]],
   1/2*Sqrt[Cross[{1, 0, 0}, {1/4, 1/4, 5}].Cross[{1, 0, 0}, {1/4, 1/4, 5}]] -
    1/2*Sqrt[Cross[{1/2, 0, 0}, {0, 0, 5/2}].Cross[{1/2, 0, 0}, {0, 0, 5/2}]],
   1/2*Sqrt[Cross[{1, 0, 0}, {1/4, 1/4, 5}].Cross[{1, 0, 0}, {1/4, 1/4, 5}]] -
    1/2*Sqrt[Cross[{1/2, 0, 0}, {0, 0, 5/2}].Cross[{1/2, 0, 0}, {0, 0, 5/2}]], 1/4}]
```

```
{0.25, 1.87812, 1.87812, 1.87812, 1.87812, 1.87812, 1.87812, 1.87812, 1.87812, 0.25}
```

```
Sum[areas[[i]], {i, 1, nface}]
```

```
15.525
```

```
mil[a_] :=
 2*volume/Sum[areas[[i]]*Sqrt[a.Outer[Times, normals[[i]], normals[[i]]].a], {i, 1, nface}]
```

```
N[mil[e1]]
```

```
0.784211
```

```
N[mil[e2]]
```

```
0.784211
```

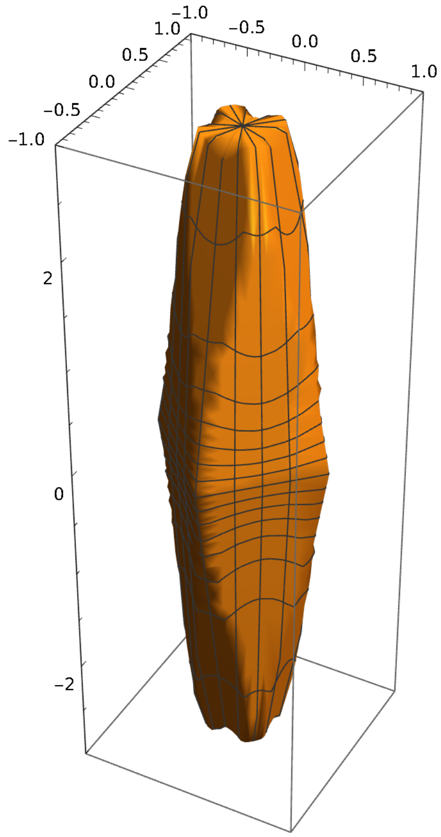
```
N[mil[e3]]
```

```
2.93472
```

```
daratio = N[mil[e3]/mil[e1]]
```

```
3.74226
```

```
ParametricPlot3D [mil[a] * a, {theta , 0, Pi},
 {phi , 0, 2 Pi}, PlotRange → {{-1, 1}, {-1, 1}, {-3, 3}}]
```



TRIFAB

```
fabricinv2approx =
  Sum[areas[[i]] / 2 / volume * Outer[Times , normals[[i]], normals[[i]]], {i , 1, nface}].
   Sum[areas[[i]] / 2 / volume * Outer[Times , normals[[i]], normals[[i]]], {i , 1, nface}];
```

```
milapprox [a_] := 1 / Sqrt[a.fabricinv2approx .a]
```

```
N[milapprox [e1]]
```

```
0.792014
```

```
N[milapprox [e2]]
```

```
0.792014
```

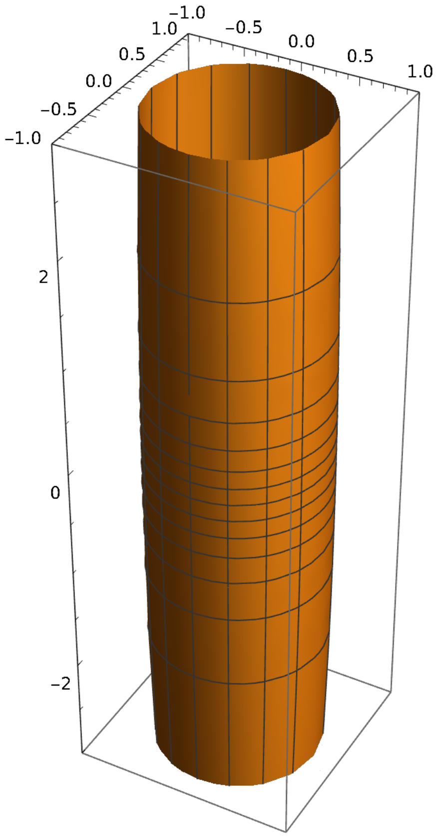
```
N[milapprox [e3]]
```

```
9.01174
```

```
daratio = N[milapprox [e3] / milapprox [e1]]
```

```
11.3783
```

```
ParametricPlot3D [milapprox [a] * a, {theta , 0, Pi},
 {phi , 0, 2 Pi}, PlotRange → {{-1, 1}, {-1, 1}, {-3, 3}}]
```



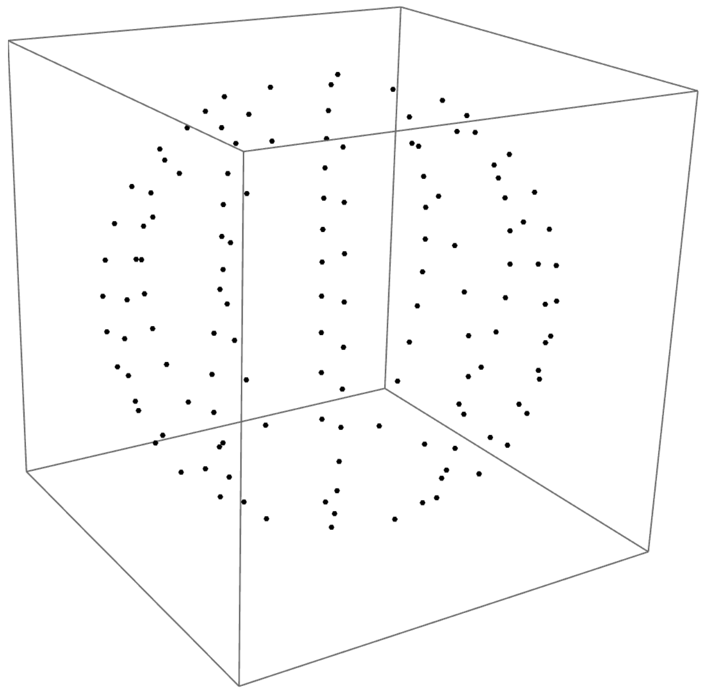
Fit of the original MIL to an ellipsoid

```
fabric[{m1_ , m2_ , m3_ , alpha_ , beta_ , gamma_}] :=
 m1 * rr[alpha , beta , gamma].Outer[Times , e1, e1].Transpose [rr[alpha , beta , gamma]] +
  m2 * rr[alpha , beta , gamma].Outer[Times , e2, e2].Transpose [rr[alpha , beta , gamma]] +
  m3 * rr[alpha , beta , gamma].Outer[Times , e3, e3].Transpose [rr[alpha , beta , gamma]]
```

```
abar = {Sqrt[1 - zeta ^ 2] * Cos[phi], Sqrt[1 - zeta ^ 2] * Sin[phi], zeta}
```

$\left\{\sqrt{1 - \text{zeta}^2}\ \text{Cos[phi]},\ \sqrt{1 - \text{zeta}^2}\ \text{Sin[phi]},\ \text{zeta}\right\}$

```
Graphics3D[Point[Flatten[Table[abar, {zeta, -1, 1, 2/12}, {phi, 0, 2 Pi, 2 Pi/12}], 1]]]
```



```
NMinimize[
 {Sum[(mil[abar] - 1/Sqrt[(abar.fabric[{1/m1^2, 1/m1^2, 1/m3^2, 0, 0, 0}].abar)])^2,
   {zeta, -1, 1, 2/72}, {phi, 0, 2 Pi, 2 Pi/72}], m1 > 0, m3 > 0}, {m1, m3}]
```

{25.6548, {m1 → 0.606181, m3 → 2.94014}}

```
fabricarg = {1/m1^2, 1/m1^2, 1/m3^2, 0, 0, 0} /. %[[2]]
```

{2.72141, 2.72141, 0.115681, 0, 0, 0}

```
Eigensystem[fabric[fabricarg]]
```

{{2.72141, 2.72141, 0.115681}, {{0., 1., 0.}, {1., 0., 0.}, {0., 0., 1.}}}

```
mil2[a_] := 1/Sqrt[a.fabric[fabricarg].a]
```

```
mil2[e1]
```

0.606181

```
mil2[e2]
```

0.606181

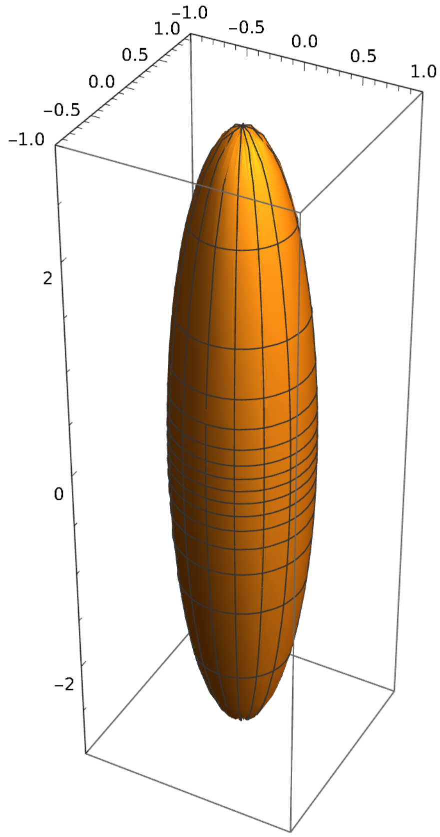
```
mil2[e3]
```

2.94014

```
daratioell = mil2[e3]/mil2[e1]
```

4.85027

```
ParametricPlot3D [mil2[a] * a, {theta , 0, Pi},
 {phi , 0, 2 Pi}, PlotRange → {{-1, 1}, {-1, 1}, {-3, 3}}]
```



Decomposition of MIL in spherical harmonics

Coefficients

```
gnum = 1 / (4 Pi) *
  NIntegrate [mil[{Sin[theta] * Cos[phi], Sin[theta] * Sin[phi], Cos[theta]}] * Sin[theta],
   {theta , 0, Pi}, {phi , 0, 2 Pi}]
```

NIntegrate ::slwcon : Numerical integration converging too slowly ; suspect one of the following : singularity , value of the integration is 0, highly oscillatory integrand , or WorkingPrecision too small . ≫

```
0.847795
```

```
ggnum = 15 / (8 Pi) * NIntegrate[mil[{Sin[theta] * Cos[phi], Sin[theta] * Sin[phi], Cos[theta]}] *
    ffsphh[{Sin[theta] * Cos[phi], Sin[theta] * Sin[phi], Cos[theta]}] *
    Sin[theta], {theta, 0, Pi}, {phi, 0, 2 Pi}]
```

NIntegrate::slwcon : Numerical integration converging too slowly; suspect one of the following: singularity, value of the integration is 0, highly oscillatory integrand, or WorkingPrecision too small. ≫

NIntegrate::eincr : The global error of the strategy GlobalAdaptive has increased more than 2000 times. The global error is expected to decrease monotonically after a number of integrand evaluations. Suspect one of the following: the working precision is insufficient for the specified precision goal; the integrand is highly oscillatory or it is not a (piecewise) smooth function; or the true value of the integral is 0. Increasing the value of the GlobalAdaptive option MaxErrorIncreases might lead to a convergent numerical integration. NIntegrate obtained -0.591876 and 1.8681919528114675`*^-6 for the integral and error estimates. ≫

NIntegrate::slwcon : Numerical integration converging too slowly; suspect one of the following: singularity, value of the integration is 0, highly oscillatory integrand, or WorkingPrecision too small. ≫

NIntegrate::eincr : The global error of the strategy GlobalAdaptive has increased more than 2000 times. The global error is expected to decrease monotonically after a number of integrand evaluations. Suspect one of the following: the working precision is insufficient for the specified precision goal; the integrand is highly oscillatory or it is not a (piecewise) smooth function; or the true value of the integral is 0. Increasing the value of the GlobalAdaptive option MaxErrorIncreases might lead to a convergent numerical integration. NIntegrate obtained 1.9817075430014492`*^-10 and 1.7365690635600095`*^-8 for the integral and error estimates. ≫

NIntegrate::slwcon : Numerical integration converging too slowly; suspect one of the following: singularity, value of the integration is 0, highly oscillatory integrand, or WorkingPrecision too small. ≫

General::stop : Further output of NIntegrate::slwcon will be suppressed during this calculation. ≫

NIntegrate::eincr : The global error of the strategy GlobalAdaptive has increased more than 2000 times. The global error is expected to decrease monotonically after a number of integrand evaluations. Suspect one of the following: the working precision is insufficient for the specified precision goal; the integrand is highly oscillatory or it is not a (piecewise) smooth function; or the true value of the integral is 0. Increasing the value of the GlobalAdaptive option MaxErrorIncreases might lead to a convergent numerical integration. NIntegrate obtained 2.116918688381253`*^-9 and 1.0839553250196608`*^-8 for the integral and error estimates. ≫

General::stop : Further output of NIntegrate::eincr will be suppressed during this calculation. ≫

$\{\{-0.35325, 1.18274\times10^{-10}, 1.26344\times10^{-9}\}, \{1.18274\times10^{-10}, -0.35325, -5.12365\times10^{-12}\}, \{1.26344\times10^{-9}, -5.12365\times10^{-12}, 0.7065\}\}$

```
ggggnum =
  315 / (32 Pi) * NIntegrate[mil[{Sin[theta] * Cos[phi], Sin[theta] * Sin[phi], Cos[theta]}] *
    ffffsphh[{Sin[theta] * Cos[phi], Sin[theta] * Sin[phi], Cos[theta]}] *
    Sin[theta], {theta, 0, Pi}, {phi, 0, 2 Pi}]
```

NIntegrate::slwcon : Numerical integration converging too slowly; suspect one of the following: singularity, value of the integration is 0, highly oscillatory integrand, or WorkingPrecision too small. ≫

NIntegrate::eincr : The global error of the strategy GlobalAdaptive has increased more than 2000 times. The global error is expected to decrease monotonically after a number of integrand evaluations. Suspect one of the following: the working precision is insufficient for the specified precision goal; the integrand is highly oscillatory or it is not a (piecewise) smooth function; or the true value of the integral is 0. Increasing the value of the GlobalAdaptive option MaxErrorIncreases might lead to a convergent numerical integration. NIntegrate obtained 0.09972253511072629` and 4.728295237892957`*^-7 for the integral and error estimates. ≫

NIntegrate::slwcon : Numerical integration converging too slowly; suspect one of the following: singularity, value of the integration is 0, highly oscillatory integrand, or WorkingPrecision too small. ≫

NIntegrate::eincr : The global error of the strategy GlobalAdaptive has increased more than 2000 times. The global error is expected to decrease monotonically after a number of integrand evaluations. Suspect one of the following: the working precision is insufficient for the specified precision goal; the integrand is highly oscillatory or it is not a (piecewise) smooth function; or the true value of the integral is 0. Increasing the value of the GlobalAdaptive option MaxErrorIncreases might lead to a convergent numerical integration. NIntegrate obtained 5.655931880401091`*^-11 and 7.092510888958495`*^-9 for the integral and error estimates. ≫

NIntegrate::slwcon : Numerical integration converging too slowly; suspect one of the following: singularity, value of the integration is 0, highly oscillatory integrand, or WorkingPrecision too small. ≫

General::stop : Further output of NIntegrate::slwcon will be suppressed during this calculation. ≫

NIntegrate::eincr : The global error of the strategy GlobalAdaptive has increased more than 2000 times. The global error is expected to decrease monotonically after a number of integrand evaluations. Suspect one of the following: the working precision is insufficient for the specified precision goal; the integrand is highly oscillatory or it is not a (piecewise) smooth function; or the true value of the integral is 0. Increasing the value of the GlobalAdaptive option MaxErrorIncreases might lead to a convergent numerical integration. NIntegrate obtained 3.4824569234596092`*^-9 and 3.3747444918016807`*^-9 for the integral and error estimates. ≫

General::stop : Further output of NIntegrate::eincr will be suppressed during this calculation. ≫

$\{\{\{\{0.312467, 1.77221\times10^{-10}, 1.09118\times10^{-8}\}, \{1.77221\times10^{-10}, -0.051426, -5.20208\times10^{-12}\},$
$\{1.09118\times10^{-8}, -5.20208\times10^{-12}, -0.261042\}\},$
$\{\{1.77221\times10^{-10}, -0.051426, -5.20208\times10^{-12}\}, \{-0.051426, 6.35835\times10^{-10},$
$-5.8798\times10^{-10}\}, \{-5.20208\times10^{-12}, -5.8798\times10^{-10}, 3.35436\times10^{-10}\}\},$
$\{\{1.09118\times10^{-8}, -5.20208\times10^{-12}, -0.261042\}, \{-5.20208\times10^{-12}, -5.8798\times10^{-10},$
$3.35436\times10^{-10}\}, \{-0.261042, 3.35436\times10^{-10}, 3.59693\times10^{-11}\}\}\},$
$\{\{\{1.77221\times10^{-10}, -0.051426, -5.20208\times10^{-12}\}, \{-0.051426, 6.35835\times10^{-10},$
$-5.8798\times10^{-10}\}, \{-5.20208\times10^{-12}, -5.8798\times10^{-10}, 3.35436\times10^{-10}\}\},$
$\{\{-0.051426, 6.35835\times10^{-10}, -5.8798\times10^{-10}\}, \{6.35835\times10^{-10}, 0.312467, 5.5134\times10^{-11}\},$
$\{-5.8798\times10^{-10}, 5.5134\times10^{-11}, -0.261042\}\},$
$\{\{-5.20208\times10^{-12}, -5.8798\times10^{-10}, 3.35436\times10^{-10}\}, \{-5.8798\times10^{-10},$
$5.5134\times10^{-11}, -0.261042\}, \{3.35436\times10^{-10}, -0.261042, 4.96551\times10^{-12}\}\}\},$
$\{\{\{1.09118\times10^{-8}, -5.20208\times10^{-12}, -0.261042\}, \{-5.20208\times10^{-12}, -5.8798\times10^{-10},$
$3.35436\times10^{-10}\}, \{-0.261042, 3.35436\times10^{-10}, 3.59693\times10^{-11}\}\},$
$\{\{-5.20208\times10^{-12}, -5.8798\times10^{-10}, 3.35436\times10^{-10}\}, \{-5.8798\times10^{-10},$
$5.5134\times10^{-11}, -0.261042\}, \{3.35436\times10^{-10}, -0.261042, 4.96551\times10^{-12}\}\},$
$\{\{-0.261042, 3.35436\times10^{-10}, 3.59693\times10^{-11}\}, \{3.35436\times10^{-10}, -0.261042, 4.96551\times10^{-12}\},$
$\{3.59693\times10^{-11}, 4.96551\times10^{-12}, 0.522082\}\}\}\}$

Estimation 2nd order

```
milsphh2[n_] := gnum + t222[ggnum, ffsphh[n]];
```

```
milsphh2[e1]
```

0.494545

```
milsphh2[e2]
```

0.494545

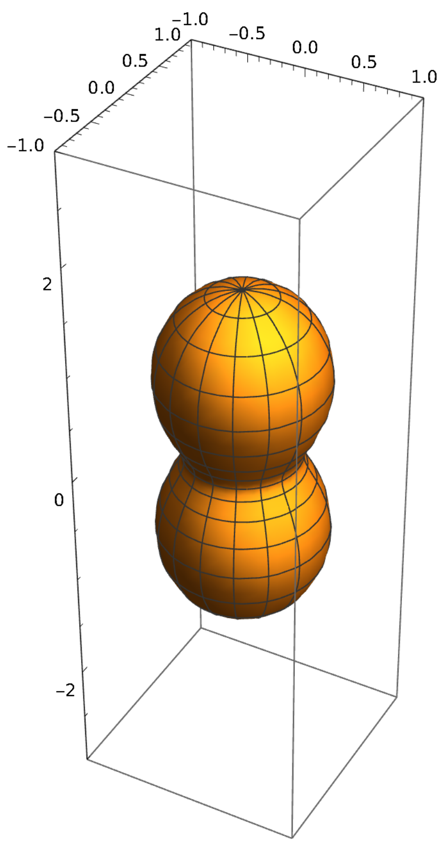
```
milsphh2[e3]
```

1.5543

```
daratiosphh2 = milsphh2[e3]/milsphh2[e1]
```

3.14288

```
ParametricPlot3D[milsphh2[n]*n, {t, 0, Pi},
 {u, 0, 2 Pi}, PlotRange → {{-1, 1}, {-1, 1}, {-3, 3}}]
```



Estimation 4th order

```
milsphh4[n_] := gnum + t222[ggnum, ffsphh[n]] + t444[ggggnum, ffffsphh[n]];
```

```
milsphh4[e1]
```

0.807012

```
milsphh4[e2]
```

0.807012

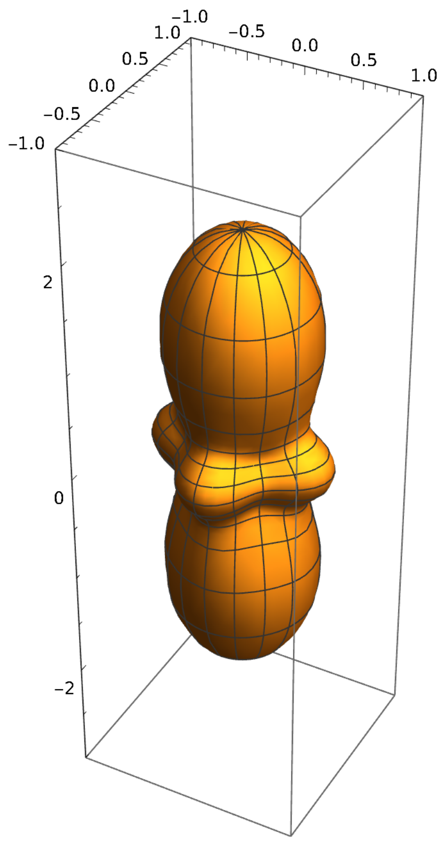
```
milsphh4[e3]
```

2.07638

```
daratiosphh4 = milsphh4[e3]/milsphh4[e1]
```

2.57292

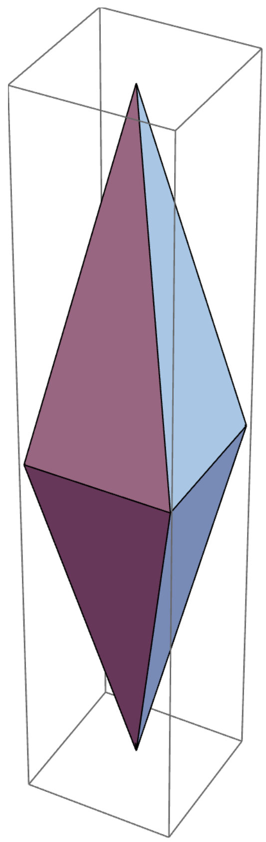
```
ParametricPlot3D[milsphh4[n]*n, {t, 0, Pi},
 {u, 0, 2 Pi}, PlotRange → {{-1, 1}, {-1, 1}, {-3, 3}}]
```



Example of a rhombus with dimensions 1x1x5mm

```
nface = 8;
```

```
Graphics3D[{Pyramid[{{0, 0, 0}, {1, 0, 0}, {1, 1, 0}, {0, 1, 0}, {1/2, 1/2, 5/2}}],
  Pyramid[{{0, 0, 0}, {1, 0, 0}, {1, 1, 0}, {0, 1, 0}, {1/2, 1/2, -5/2}}]}]
```



```
volume = N[2/3 * 5/2]
```

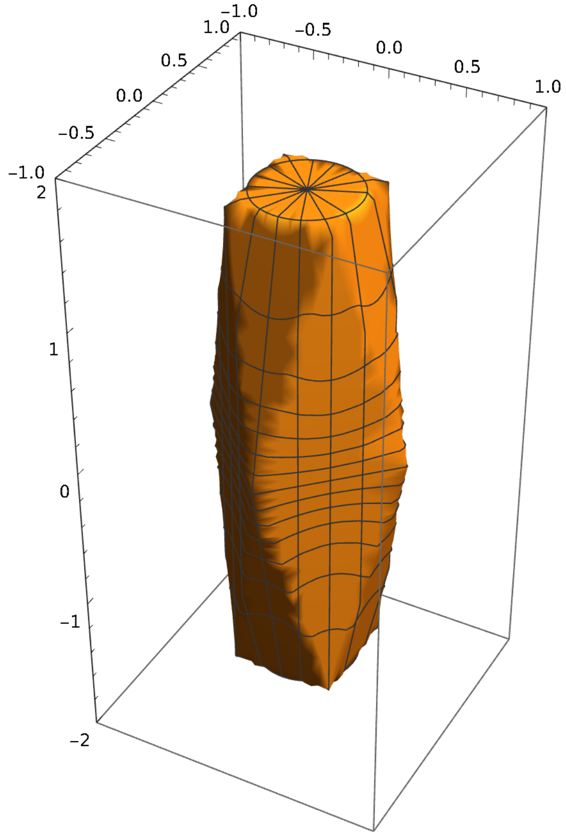
```
1.66667
```

```
normals = N[{-Cross[{1, 0, 0}, {1/2, 1/2, 5/2}]/Sqrt[{1/2, 1/2, 5/2}.{1/2, 1/2, 5/2}],
   Cross[{0, 1, 0}, {1/2, 1/2, 5/2}]/Sqrt[{1/2, 1/2, 5/2}.{1/2, 1/2, 5/2}],
   Cross[{1, 0, 0}, {-1/2, -1/2, 5/2}]/Sqrt[{-1/2, -1/2, 5/2}.{-1/2, -1/2, 5/2}],
   -Cross[{0, 1, 0}, {-1/2, -1/2, 5/2}]/Sqrt[{-1/2, -1/2, 5/2}.{-1/2, -1/2, 5/2}],
   Cross[{1, 0, 0}, {1/2, 1/2, 5/2}]/Sqrt[{1/2, 1/2, 5/2}.{1/2, 1/2, 5/2}],
   -Cross[{0, 1, 0}, {1/2, 1/2, 5/2}]/Sqrt[{1/2, 1/2, 5/2}.{1/2, 1/2, 5/2}],
   -Cross[{1, 0, 0}, {-1/2, -1/2, 5/2}]/Sqrt[{-1/2, -1/2, 5/2}.{-1/2, -1/2, 5/2}],
   Cross[{0, 1, 0}, {-1/2, -1/2, 5/2}]/Sqrt[{-1/2, -1/2, 5/2}.{-1/2, -1/2, 5/2}]}]
```

```
{{0., 0.96225, -0.19245}, {0.96225, 0., -0.19245},
 {0., -0.96225, -0.19245}, {-0.96225, 0., -0.19245}, {0., -0.96225, 0.19245},
 {-0.96225, 0., 0.19245}, {0., 0.96225, 0.19245}, {0.96225, 0., 0.19245}}
```

```
areas = N[{1/2 * Sqrt[Cross[{1, 0, 0}, {1/2, 1/2, 5/2}].Cross[{1, 0, 0}, {1/2, 1/2, 5/2}]],
   1/2 * Sqrt[Cross[{1, 0, 0}, {1/2, 1/2, 5/2}].Cross[{1, 0, 0}, {1/2, 1/2, 5/2}]],
   1/2 * Sqrt[Cross[{1, 0, 0}, {1/2, 1/2, 5/2}].Cross[{1, 0, 0}, {1/2, 1/2, 5/2}]],
   1/2 * Sqrt[Cross[{1, 0, 0}, {1/2, 1/2, 5/2}].Cross[{1, 0, 0}, {1/2, 1/2, 5/2}]],
   1/2 * Sqrt[Cross[{1, 0, 0}, {1/2, 1/2, 5/2}].Cross[{1, 0, 0}, {1/2, 1/2, 5/2}]],
   1/2 * Sqrt[Cross[{1, 0, 0}, {1/2, 1/2, 5/2}].Cross[{1, 0, 0}, {1/2, 1/2, 5/2}]],
   1/2 * Sqrt[Cross[{1, 0, 0}, {1/2, 1/2, 5/2}].Cross[{1, 0, 0}, {1/2, 1/2, 5/2}]],
   1/2 * Sqrt[Cross[{1, 0, 0}, {1/2, 1/2, 5/2}].Cross[{1, 0, 0}, {1/2, 1/2, 5/2}]]}]

{1.27475, 1.27475, 1.27475, 1.27475, 1.27475, 1.27475, 1.27475, 1.27475}

Sum[areas[[i]], {i, 1, nface}]

10.198

mil[a_] :=
 2 * volume / Sum[areas[[i]] * Sqrt[a.Outer[Times, normals[[i]], normals[[i]]].a], {i, 1, nface}]

N[mil[e1]]

0.679366

N[mil[e2]]

0.679366

N[mil[e3]]

1.69842

daratio = N[mil[e3] / mil[e1]]

2.5
```

```
ParametricPlot3D [mil[a] * a, {theta , 0, Pi},
 {phi , 0, 2 Pi}, PlotRange → {{-1, 1}, {-1, 1}, {-2, 2}}]
```



TRIFAB

```
fabricinv2approx =
  Sum[areas[[i]] / 2 / volume * Outer[Times , normals[[i]], normals[[i]]], {i , 1, nface}].
   Sum[areas[[i]] / 2 / volume * Outer[Times , normals[[i]], normals[[i]]], {i , 1, nface}];
```

```
milapprox[a_] := 1 / Sqrt[a.fabricinv2approx .a]
```

```
N[milapprox[e1]]
```

```
0.706018
```

```
N[milapprox[e2]]
```

```
0.706018
```

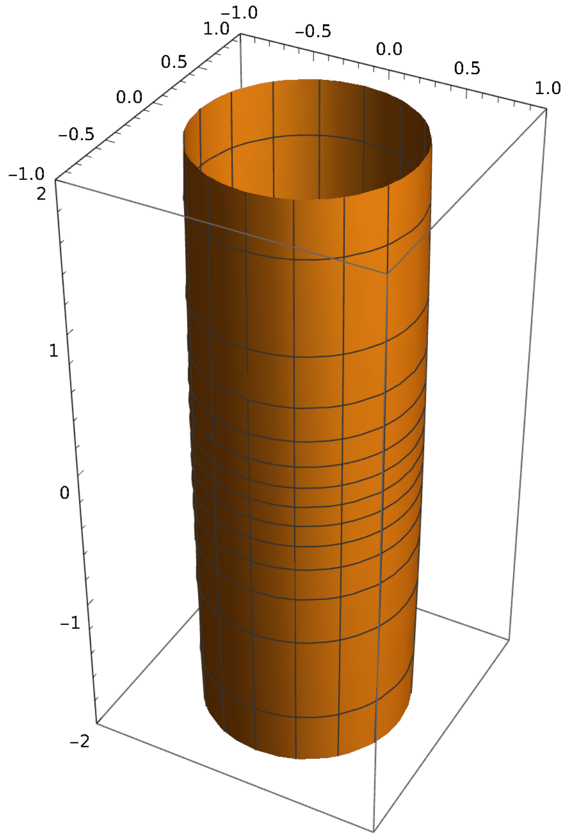
```
N[milapprox[e3]]
```

```
8.82523
```

```
daratio = N[milapprox[e3] / milapprox[e1]]
```

```
12.5
```

```
ParametricPlot3D [milapprox [a] * a, {theta , 0, Pi},
 {phi , 0, 2 Pi}, PlotRange → {{-1, 1}, {-1, 1}, {-2, 2}}]
```



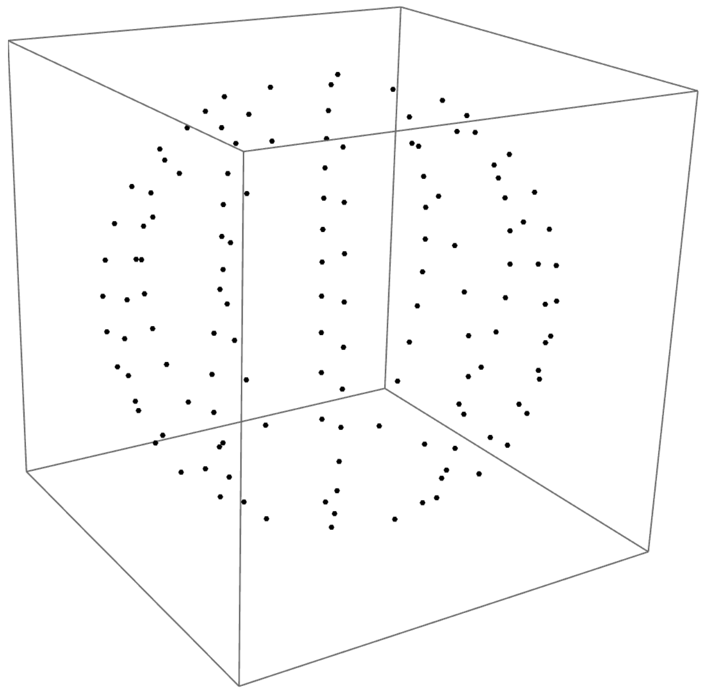
Fit of the original MIL to an ellispsoid

```
fabric[{m1_ , m2_ , m3_ , alpha_ , beta_ , gamma_}] :=
 m1 * rr[alpha , beta , gamma].Outer[Times , e1, e1].Transpose [rr[alpha , beta , gamma]] +
  m2 * rr[alpha , beta , gamma].Outer[Times , e2, e2].Transpose [rr[alpha , beta , gamma]] +
  m3 * rr[alpha , beta , gamma].Outer[Times , e3, e3].Transpose [rr[alpha , beta , gamma]]
```

```
abar = {Sqrt[1 - zeta ^ 2] * Cos[phi], Sqrt[1 - zeta ^ 2] * Sin[phi], zeta}
```

$$\left\{\sqrt{1-\text{zeta}^2}\ \text{Cos[phi]},\ \sqrt{1-\text{zeta}^2}\ \text{Sin[phi]},\ \text{zeta}\right\}$$

```
Graphics3D[Point[Flatten[Table[abar, {zeta, -1, 1, 2/12}, {phi, 0, 2 Pi, 2 Pi/12}], 1]]]
```



```
NMinimize[
 {Sum[(mil[abar] - 1/Sqrt[(abar.fabric[{1/m1^2, 1/m1^2, 1/m3^2, 0, 0, 0}].abar)])^2,
   {zeta, -1, 1, 2/90}, {phi, 0, 2 Pi, 2 Pi/90}], m1 > 0, m3 > 0}, {m1, m3}]
```

```
{39.1031, {m1 → 0.55709, m3 → 1.86687}}
```

```
fabricarg = {1/m1^2, 1/m1^2, 1/m3^2, 0, 0, 0} /. %[[2]]
```

```
{3.22218, 3.22218, 0.286926, 0, 0, 0}
```

```
Eigensystem[fabric[fabricarg]]
```

```
{{3.22218, 3.22218, 0.286926}, {{0., 1., 0.}, {1., 0., 0.}, {0., 0., 1.}}}
```

```
mil2[a_] := 1/Sqrt[a.fabric[fabricarg].a]
```

```
mil2[e1]
```

```
0.55709
```

```
mil2[e2]
```

```
0.55709
```

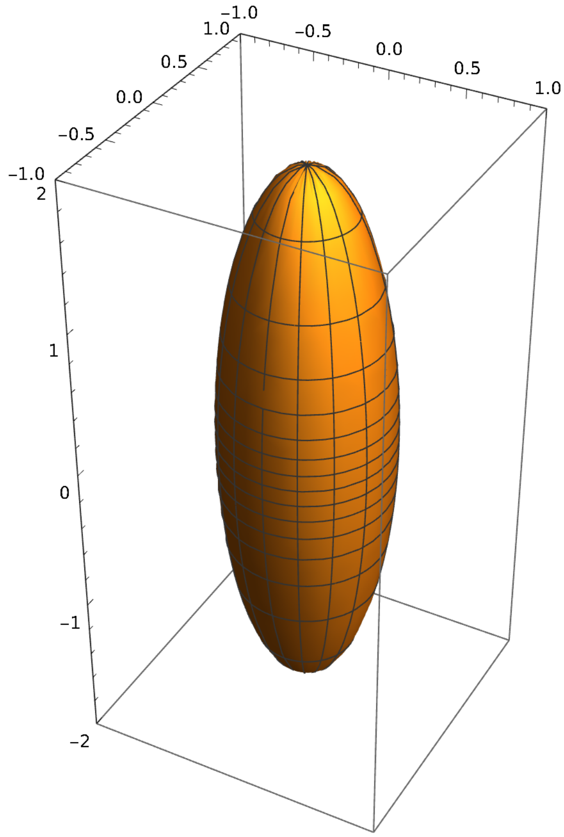
```
mil2[e3]
```

```
1.86687
```

```
daratio = mil2[e3]/mil2[e1]
```

```
3.35112
```

```
ParametricPlot3D [mil2[a] * a, {theta , 0, Pi},
 {phi , 0, 2 Pi}, PlotRange → {{-1, 1}, {-1, 1}, {-2, 2}}]
```



Decomposition of original MIL in spherical harmonics

Coefficients

```
gnum = 1 / (4 Pi) *
  NIntegrate [mil[{Sin[theta] * Cos[phi], Sin[theta] * Sin[phi], Cos[theta]}] * Sin[theta],
   {theta , 0, Pi}, {phi , 0, 2 Pi}]
```

NIntegrate ::slwcon : Numerical integration converging too slowly ; suspect one of the following : singularity , value of the integration is 0, highly oscillatory integrand , or WorkingPrecision too small . ≫

```
0.736671
```

```
ggnum = 15 / (8 Pi) * NIntegrate [mil[{Sin[theta] * Cos[phi], Sin[theta] * Sin[phi], Cos[theta]}] *
    ffsphh [{Sin[theta] * Cos[phi], Sin[theta] * Sin[phi], Cos[theta]}] *
    Sin[theta], {theta , 0, Pi}, {phi , 0, 2 Pi}]
```

NIntegrate ::slwcon : Numerical integration converging too slowly ; suspect one of the following : singularity , value of the integration is 0, highly oscillatory integrand , or WorkingPrecision too small . ≫

NIntegrate ::eincr : The global error of the strategy GlobalAdaptive has increased more than 2000 times . The global error is expected to decrease monotonically after a number of integrand evaluations . Suspect one of the following : the working precision is insufficient for the specified precision goal ; the integrand is highly oscillatory or it is not a (piecewise) smooth function ; or the true value of the integral is 0. Increasing the value of the GlobalAdaptive option MaxErrorIncreases might lead to a convergent numerical integration . NIntegrate obtained –0.474352 and 2.083444156114494`*^–6 for the integral and error estimates . ≫

NIntegrate ::slwcon : Numerical integration converging too slowly ; suspect one of the following : singularity , value of the integration is 0, highly oscillatory integrand , or WorkingPrecision too small . ≫

NIntegrate ::eincr : The global error of the strategy GlobalAdaptive has increased more than 2000 times . The global error is expected to decrease monotonically after a number of integrand evaluations . Suspect one of the following : the working precision is insufficient for the specified precision goal ; the integrand is highly oscillatory or it is not a (piecewise) smooth function ; or the true value of the integral is 0. Increasing the value of the GlobalAdaptive option MaxErrorIncreases might lead to a convergent numerical integration . NIntegrate obtained 9.454783470292658`*^–11 and 5.4193548135148445`*^–8 for the integral and error estimates . ≫

NIntegrate ::slwcon : Numerical integration converging too slowly ; suspect one of the following : singularity , value of the integration is 0, highly oscillatory integrand , or WorkingPrecision too small . ≫

General ::stop : Further output of NIntegrate ::slwcon will be suppressed during this calculation . ≫

NIntegrate ::eincr : The global error of the strategy GlobalAdaptive has increased more than 2000 times . The global error is expected to decrease monotonically after a number of integrand evaluations . Suspect one of the following : the working precision is insufficient for the specified precision goal ; the integrand is highly oscillatory or it is not a (piecewise) smooth function ; or the true value of the integral is 0. Increasing the value of the GlobalAdaptive option MaxErrorIncreases might lead to a convergent numerical integration . NIntegrate obtained -1.20078×10^{-8} and 2.162185027648949`*^–8 for the integral and error estimates . ≫

General ::stop : Further output of NIntegrate ::eincr will be suppressed during this calculation . ≫

$$\{\{-0.283108, 5.64291\times10^{-11}, -7.16664\times10^{-9}\}, \{5.64291\times10^{-11}, -0.283108, 1.17955\times10^{-16}\}, \{-7.16664\times10^{-9}, 1.17955\times10^{-16}, 0.566217\}\}$$

```
ggggnum =
  315 / (32 Pi) * NIntegrate [mil[{Sin[theta] * Cos[phi], Sin[theta] * Sin[phi], Cos[theta]}] *
    ffffsphh [{Sin[theta] * Cos[phi], Sin[theta] * Sin[phi], Cos[theta]}] *
    Sin[theta], {theta , 0, Pi}, {phi , 0, 2 Pi}]
```

NIntegrate ::slwcon : Numerical integration converging too slowly ; suspect one of the following : singularity , value of the integration is 0, highly oscillatory integrand , or WorkingPrecision too small . ≫

NIntegrate ::eincr : The global error of the strategy GlobalAdaptive has increased more than 2000 times . The global error is expected to decrease monotonically after a number of integrand evaluations . Suspect one of the following : the working precision is insufficient for the specified precision goal ; the integrand is highly oscillatory or it is not a (piecewise) smooth function ; or the true value of the integral is 0. Increasing the value of the GlobalAdaptive option MaxErrorIncreases might lead to a convergent numerical integration . NIntegrate obtained 0.07019550344451209` and 5.03522526932154`*^-7 for the integral and error estimates . ≫

NIntegrate ::slwcon : Numerical integration converging too slowly ; suspect one of the following : singularity , value of the integration is 0, highly oscillatory integrand , or WorkingPrecision too small . ≫

NIntegrate ::eincr : The global error of the strategy GlobalAdaptive has increased more than 2000 times . The global error is expected to decrease monotonically after a number of integrand evaluations . Suspect one of the following : the working precision is insufficient for the specified precision goal ; the integrand is highly oscillatory or it is not a (piecewise) smooth function ; or the true value of the integral is 0. Increasing the value of the GlobalAdaptive option MaxErrorIncreases might lead to a convergent numerical integration . NIntegrate obtained 4.5633688895222614`*^-11 and 1.9348489644151374`*^-8 for the integral and error estimates . ≫

NIntegrate ::slwcon : Numerical integration converging too slowly ; suspect one of the following : singularity , value of the integration is 0, highly oscillatory integrand , or WorkingPrecision too small . ≫

General ::stop : Further output of NIntegrate ::slwcon will be suppressed during this calculation . ≫

NIntegrate ::eincr : The global error of the strategy GlobalAdaptive has increased more than 2000 times . The global error is expected to decrease monotonically after a number of integrand evaluations . Suspect one of the following : the working precision is insufficient for the specified precision goal ; the integrand is highly oscillatory or it is not a (piecewise) smooth function ; or the true value of the integral is 0. Increasing the value of the GlobalAdaptive option MaxErrorIncreases might lead to a convergent numerical integration . NIntegrate obtained 1.5247950453846139`*^-10 and 7.65224824140596`*^-9 for the integral and error estimates . ≫

General ::stop : Further output of NIntegrate ::eincr will be suppressed during this calculation . ≫

$$\{\{\{\{0.219948, 1.42987\times10^{-10}, 4.77774\times10^{-10}\}, \{1.42987\times10^{-10}, -0.0562, 1.03705\times10^{-11}\}, \{4.77774\times10^{-10}, 1.03705\times10^{-11}, -0.163748\}\},$$
$$\{\{1.42987\times10^{-10}, -0.0562, 1.03705\times10^{-11}\}, \{-0.0562, -8.02852\times10^{-10}, -2.31708\times10^{-10}\}, \{1.03705\times10^{-11}, -2.31708\times10^{-10}, 5.23819\times10^{-11}\}\},$$
$$\{\{4.77774\times10^{-10}, 1.03705\times10^{-11}, -0.163748\}, \{1.03705\times10^{-11}, -2.31708\times10^{-10}, 5.23819\times10^{-11}\}, \{-0.163748, 5.23819\times10^{-11}, -3.17038\times10^{-12}\}\}\},$$
$$\{\{\{1.42987\times10^{-10}, -0.0562, 1.03705\times10^{-11}\}, \{-0.0562, -8.02852\times10^{-10}, -2.31708\times10^{-10}\}, \{1.03705\times10^{-11}, -2.31708\times10^{-10}, 5.23819\times10^{-11}\}\},$$
$$\{\{-0.0562, -8.02852\times10^{-10}, -2.31708\times10^{-10}\}, \{-8.02852\times10^{-10}, 0.219948, 3.82536\times10^{-11}\}, \{-2.31708\times10^{-10}, 3.82536\times10^{-11}, -0.163748\}\},$$
$$\{\{1.03705\times10^{-11}, -2.31708\times10^{-10}, 5.23819\times10^{-11}\}, \{-2.31708\times10^{-10}, 3.82536\times10^{-11}, -0.163748\}, \{5.23819\times10^{-11}, -0.163748, -4.47549\times10^{-11}\}\}\},$$
$$\{\{\{4.77774\times10^{-10}, 1.03705\times10^{-11}, -0.163748\}, \{1.03705\times10^{-11}, -2.31708\times10^{-10}, 5.23819\times10^{-11}\}, \{-0.163748, 5.23819\times10^{-11}, -3.17038\times10^{-12}\}\},$$
$$\{\{1.03705\times10^{-11}, -2.31708\times10^{-10}, 5.23819\times10^{-11}\}, \{-2.31708\times10^{-10}, 3.82536\times10^{-11}, -0.163748\}, \{5.23819\times10^{-11}, -0.163748, -4.47549\times10^{-11}\}\},$$
$$\{\{-0.163748, 5.23819\times10^{-11}, -3.17038\times10^{-12}\}, \{5.23819\times10^{-11}, -0.163748, -4.47549\times10^{-11}\}, \{-3.17038\times10^{-12}, -4.47549\times10^{-11}, 0.327496\}\}\}\}$$

Estimation 2nd order

```
milsphh2[n_] := gnum + t222[ggnum, ffsphh[n]];
```

```
milsphh2[e1]
```

0.453563

```
milsphh2[e2]
```

0.453563

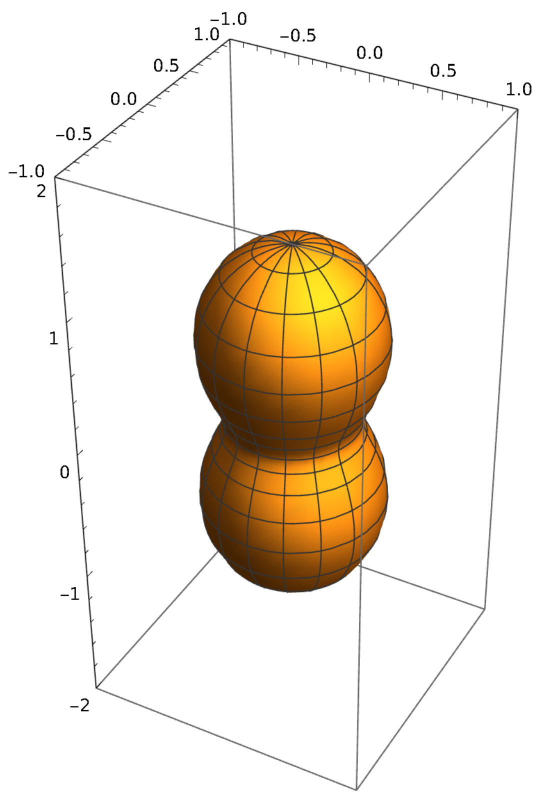
```
milsphh2[e3]
```

1.30289

```
daratiosphh2 = milsphh2[e3]/milsphh2[e1]
```

2.87256

```
ParametricPlot3D[milsphh2[n]*n, {t, 0, Pi},
 {u, 0, 2 Pi}, PlotRange → {{-1, 1}, {-1, 1}, {-2, 2}}]
```



Estimation 4th order

```
milsphh4[n_] := gnum + t222[ggnum, ffsphh[n]] + t444[ggggnum, ffffsphh[n]];
```

```
milsphh4[e1]
```

0.673511

```
milsphh4[e2]
```

0.673511

```
milsphh4[e3]
```

1.63038

```
daratio4sphh4 = milsphh4[e3]/milsphh4[e1]
```

2.42072

```
ParametricPlot3D[milsphh4[n]*n, {t, 0, Pi},
 {u, 0, 2 Pi}, PlotRange → {{-1, 1}, {-1, 1}, {-2, 2}}]
```