

[ > **restart:**

# PARAMETRIC SURFACES

parametric.mws -- updated August 20, 1999

R. M. Kiehn.

rkiehn2352@aol.com

Lecture Notes at <http://www22.pair.com/csdc/pdf/defects2.pdf>

Maple download at <http://www22.pair.com/csdc/maple/parametric.zip>

[ > **with(linalg):with(plots):with(liesymm):with(diffforms): setup(u,v):  
deform(Z=0,u=0,v=0,p=const,q=const);**

Parametric 2-surfaces in 3D support a frame matrix which is always *affine* Torsion free (torsion of type 1 defined as Particle Affine torsion), and can be **adjusted** such as to have zero *adjoint or expansion with twist* Torsion (torsion of type 2 or Wave Affine torsion). Not only are the *affine* Torsion coefficients (which depend upon the existence of little omega - see notes) equal to zero for parametric surfaces, but also the *adjoint or twisted* Torsion coefficients (which depend upon the existence of Big Omega - see notes) can be made to vanish by a unique choice of scale (the Holder euclidean norm) for the adjoint (normal) field ( $q=1,p=2$ )

**Think of the expansion with twist type of torsion in terms of a Bourdon tube in the shape of a spiral-helix. As the interior volume expands, the endpoints experience a rotational twist relative to one another.**

In the general case of a projective Frame matrix, the (two species of) Torsion coefficients are measures of how accurately changes in the position vector and changes in the adjoint normal field are confined to the surface. When these changes are not confined to the surface, then the system is said to have a defect. There are two species of defects. The twisted torsion defect is that idea which started the concept of Gauge theories (H. Weyl about 1919) in physics. However, it is often confused with the notion of the affine torsion defect (which has been applied to theory of crystal dislocations by Kondo about 1950)

When the *adjoint or twisted* Torsion coefficients are non-zero, the adjoint field can experience changes which are NOT confined to the surface. Such defects are the source of tip "vortices" in airplane wings of finite span. The circulation profile is not planar in a transverse direction to the wingspan (finite wing). In fact, the flow lines about the top of the wing and the flow lines about the bottom of the wing do not meet at the trailing edge! They are torsionally displaced. For an infinite wing, the circulation profile is planar, and there is no drag. For a finite wing, the circulation profile has a twist. In such situations the big Omega is not zero.

The idea of parametric surfaces is to build a global basis Frame of N linearly independent vectors, whose first N-1 columns are composed of the "tangent" vectors (the partials of the N dimensional position vector with respect to the N-1 parameters). These first N-1 vectors are effectively the elements of a Jacobian (non-square) matrix for the given parametric map.

The Frame matrix adds to this system of N-1 tangent (column) vectors, a vector composed of components constructed from the column of adjoint coefficients (all N-1 sub-determinants of the N-1 tangent vectors). This adjoint vector can have an arbitrary (projective) scale, and yet linear independence of the Frame is

preserved. The Frame matrix induces (pulls back) a metric that can be used on the N-1 dimensional surface.

Note that the Gibbs cross-product is equivalent to the adjoint construction procedure (in terms of 2x2 sub-determinants) in N= 3 dimensions. However, the adjoint (Grassman sub\_determinant) process works as well in N dimensions, where the Gibbs product does not.

## Special Case: Monge SURFACES in 3D

Monge surfaces are a special form of the parametric surface (although Osserman calls such surfaces "non-parametric") where two of the three coordinates are identified with the two parameters, and the third coordinate with the arbitrary (surface) function  $Z(u,v)$ . The construction is still a map from 2 to 3 D, with the 3 D position vector having components as given below:

Define a position vector in R3 as three functions parametrized by two variables  $u,v$ , and the Monge function  $Z(u,v)$

```
> R:=[u,v,Z(u,v)];d(R);
```

Define two tangent vectors,  $Y_u$  and  $Y_v$

```
> Yu:=[diff(R[1],u),diff(R[2],u),diff(R[3],u)];
```

```
> Yv:=[diff(R[1],v),diff(R[2],v),diff(R[3],v)];
```

```
> F2:=array([[Yu[1],Yv[1],0],[Yu[2],Yv[2],0],[Yu[3],Yv[3],1] ] );
```

Use the Grassman adjoint method to define the "adjoint" field. (equivalent to the Gibbs cross product in 3D)

```
> Grass:=(transpose(adjoint(F2)));det(Grass);
```

```
> NNU:=evalm([Grass[1,3],Grass[2,3],Grass[3,3]]);
```

```
>
```

The normal field can be scaled by suitable functions which will influence a number of the results below

Scale the (adjoint) normal field here by  $\rho$  (the particular choice - the euclidean choice- shown below will eliminate the adjoint twisted Torsion coefficients.)

```
> magn:=(NNU[1]^p+NNU[2]^p+NNU[3]^p)^(q/p);
```

```
>
```

```
>
```

Any other choice of the scaling factor  $\rho$  will determine the adjoint twisted Torsion coefficients of the system.

When the position vector is parametrically defined, the 1-form, small  $\omega$  (see notes <http://www.uh.edu/~rkiehn/pdf/defects2.pdf>), vanishes identically, and therefore there is no affine (translational dislocation shear) Torsion for the parametric surfaces.

However, the adjoint twisted Torsion coefficients exist unless the normal field is scaled by the quadratic factor given above (the Gauss map). As seen below, the determinant of the Frame matrix is also determined by the scaling factor of the adjoint normal field, and for the special case where the adjoint twisted (exact) 1-form  $\Omega$  vanishes, the determinant is not unimodular.

Note the scaling could have been chosen to leave the Frame matrix unimodular or even involutory!

```
> p:=2;magn;
```

```

[ choose another renormalization above.
[ > NNS:=evalm([factor(NNU[1]),factor(NNU[2]),simplify(factor(NNU[3]))]/magn);
[ Construct the Cartan Frame matrix from the two "tangent" vectors and the adjoint normal:
[ > FF:=array([[Yu[1],Yv[1],NNS[1]],[Yu[2],Yv[2],NNS[2]],[Yu[3],Yv[3],NNS[3]]  ]]);
[ The Repere Mobile or FRAME MATRIX, FF. note that the frame matrix is not orthonormal!!
[ > fdetFF:=simplify(factor(det(FF)));
[ > Gmn:=innerprod(transpose(FF),FF);
[ The parametric frame induces a metric on the pullback of the euclidean 3-space which partitions the
[ surface and the normal fields. The scaling in the normal direction (rho) is not necessarily a constant and
[ will support expansions or contractions in the normal direction. This phenomena will lead to what is
[ called "adjoint twisted Torsion" ( not affine Torsion, see below). When the G33 coefficient is equal to one,
[ the induced metric matrix has a fixed point.
[ Other interesting choices for rho will be examined below.
[ The determinant of the Frame is non-zero globally, hence an inverse always exists. Hence the Cartan
[ matrix of connection 1-forms is globally defined.
[ > FFINVD:=evalm(FF^(-1));
[ The 1-form components of the differential position vector dR can be re-expressed with respect to the Basis
[ Frame, F.
[ > dRF:=innerprod(FFINVD,d(R));
[ The result is a column matrix of 1-forms:
[ > sigma1:=wcollect(dRF[1]);
[ > sigma2:=wcollect(dRF[2]);
[ Note that sigma1 is du and sigma2 is dv for a parametric Monge surfaces!!
[ > omega:=(wcollect(dRF[3]));
[ Note that this term (small omega) vanishes identically for a parametric Monge surface !!
[ The result is independent from the choice of scaling!!
[ Hence parametric Monge surfaces exhibit no TORSION!! of the Affine type
[ ( that is, Monge surfaces do not support translational shear defects!)
[ Monge surfaces do not describe crystal dislocations.
[
[ The result is also true for any parametric Monge 2-surface in 3D. It is also true for any N-1 parametric
[ surface in N-D. Parametric Monge surfaces do not support affine Torsion.
[
[ Now compute the Cartan matrix of connection 1-forms from C=[F(inverse)] times d[F]
[ > dFF:=array([[d(FF[1,1]),d(FF[1,2]),d(FF[1,3])],[d(FF[2,1]),d(FF[2,2]),d(FF[2,3])
[ ],[d(FF[3,1]),d(FF[3,2]),d(FF[3,3])]]):
[ > cartan:=evalm(FFINVD*dFF);
[ The interior connection coefficients (can be Christoffel symbols on the parameter space -- see notes at
[ http://www.uh.edu/~rkiehn/pdf/repere.pdf)
[ Note that the Big Gamma (interior connection) coefficients on the subspace do not depend upon the scaling
[ factor rho.
[ > Gamma11:=(wcollect(cartan[1,1]));
[ > Gamma12:=(wcollect(cartan[1,2]));
[ > Gamma21:=(wcollect(cartan[2,1]));
[ > Gamma22:=(wcollect(cartan[2,2]));
[ The second fundamental form or shape matrix comes from the third row of the Cartan matrix
[ > h1:=wcollect(factor(cartan[3,1]));

```

```
[ > gamma1:=simplify(factor(cartan[1,3]));
[ > h2:=(wcollect(factor(cartan[3,2])));
[ > gamma2:=(wcollect(factor(cartan[2,3])));
```

It is apparent that the components of gamma and h (vector valued 1-forms) are (in general) not proportional. The components are proportional if the Frame matrix is an element of the orthogonal group. Such constraints impose severe conditions on the connection 2-forms on the interior surface.

However, the "inner product" 2-form  $\langle h | \wedge | \gamma \rangle$  for a parametric surface must vanish.

```
[ > ZZZ:=simplify(wcollect(h1&^gamma1+h2&^gamma2));
```

Next compute the "expansion" circulation defect (twist) 1-form (Big Omega in the notes): This expansion concept is related to the G33 component of the induced metric field not being unity.

The "expansion circulation" twist for the parametric surface will show up as a non-zero entry in the [3,3] slot of the Cartan Matrix. This coefficient (Big Omega) is always an exact differential for parametric and Monge surfaces. The circulation twist is not zero unless the special scaling is subsumed that forces the 33 component of the induced metric to be equal to 1.

If the scaling is arbitrary, the parametric (Monge) surfaces will admit circulation defects (Torsion of the second kind due to circulation defects).

What happens when the scaling is chosen such that is minus 1 ?

```
[ > Omega:=wcollect(factor((wcollect(simplify(cartan[3,3]))));dOmega:=d(Omega);
```

Note that if  $q = 1$ , then big Omega is zero.

(Omega vanishes when the scaling is adjusted to give a unit normal field. ( $q=1$ ))

In all cases, the Big Omega 1-form is exact and has a primitive function,  $\Phi(u,v)$ . Hence any curve such that  $\Phi(u,v) = \text{constant}$  is very special, for then the Big Omega (abnormality) term vanishes.

$\Omega = d(\Phi(u,v))$

```
[ > shape11:=-simplify(gamma1&^d(v)/d(u)&^d(v));
[ > shape12:=-simplify(gamma1&^d(u)/d(v)&^d(u));
[ > shape21:=-simplify(gamma2&^d(v)/d(u)&^d(v));
[ > shape22:=-simplify(gamma2&^d(u)/d(v)&^d(u));
[ >
[ > SHAPE:=array([[shape11,shape12],[shape21,shape22]]);
[ > HH:=factor(trace(SHAPE)/2);
[ > print(`Mean Curvature is `,simplify(factor(subs((HH)))));
[ > KK:=simplify(det(SHAPE));
[ > print(`Gauss Curvature is `,subs(factor(KK)));
```

Note that the zeros of the Mean and Gauss curvatures do not depend upon the scaling coefficient. The abnormality Torsion does not effect flatness or minimal surfaces.

## CARTAN TORSION AND CURVATURE

Next compute the induced Cartan Torsion 2-forms and the Cartan Curvature 2-forms on the 2D subspace

```
[ > Sigma1:=simplify(omega&^gamma1);Sigma2:=simplify(omega&^gamma2);
[ > Phi1:=simplify(Omega&^h1);Phi2:=simplify(Omega&^h2);
```

It is characteristic of Monge (and other) parametric surfaces to exhibit Torsion 2-forms of the second type (Wave Affine) on the subspace if the Gauss normalization (unit normal field) is NOT assumed. The Particle Affine Torsion 2-forms are zero

```
[ > CURV2FORMS:=evalm(array([[simplify(h1&^gamma1),simplify(h1&^gamma2)],[factor(simplify(h2&^gamma1)),simplify(h2&^gamma2)]]));
```

## EXAMPLE: the Monkey saddle

```
> Z(u,v):=u^3-3*u*v^2;rho(u,v):=magn;
R:=[u,v,Z(u,v)];dR:=[d(R[1]),d(R[2]),d(R[3])];
> FF:=array([[Yu[1],Yv[1],NNS[1]],[Yu[2],Yv[2],NNS[2]],[Yu[3],Yv[3],NNS[3]] ]):
```

The Repere Mobile or FRAME MATRIX, FF. note that the frame matrix is not orthonormal!!

```
> detFF:=simplify(factor(det(FF))):
> Gmn:=innerprod(transpose(FF),FF):
> FFINVD:=evalm(FF^(-1)):
```

The 1-form components of the differential position vector dR can be re-expressed with respect to the Basis Frame, F.

```
> dRF:=innerprod(FFINVD,[dR[1],dR[2],dR[3]]):
```

The result is a column matrix of 1-forms:

```
> sigma1:=wcollect(dRF[1]);
> sigma2:=wcollect(dRF[2]);
```

Note that sigma1 is du and sigma2 is dv for a parametric Monge surfaces!!

```
> omega:=(wcollect(dRF[3]));
```

Note that this term (small omega) vanishes identically for a parametric Monge surface !!

```
> dFF:=array([[d(FF[1,1]),d(FF[1,2]),d(FF[1,3])],[d(FF[2,1]),d(FF[2,2]),d(FF[2,3])
],[d(FF[3,1]),d(FF[3,2]),d(FF[3,3])]]):
> cartan:=evalm(FFINVD*dFF):
```

The interior connection coefficients (can be Christoffel symbols on the parameter space -- see notes at <http://www.uh.edu/~rkiehn/pdf/defects2.pdf>)

Note that the Big Gamma (interior connection) coefficients on the subspace do not depend upon the scaling factor rho.

```
> Gamma11:=(wcollect(cartan[1,1])):
> Gamma12:=(wcollect(cartan[1,2])):
> Gamma21:=(wcollect(cartan[2,1])):
> Gamma22:=(wcollect(cartan[2,2])):
```

The second fundamental form or shape matrix comes from the third row of the Cartan matrix

```
> h1:=wcollect(factor(cartan[3,1])):
> gamma1:=wcollect(factor(cartan[1,3])):
> h2:=wcollect(factor(cartan[3,2])):
> gamma2:=wcollect(factor(cartan[2,3])):
```

```
> Omega:=(simplify(factor(cartan[3,3])));d(Omega);omega;
```

(Omega vanishes when the scaling rho is proportional to magnitude of the normal field.

The Omega 1-form is exact and has a primitive function

Phi(u,v) = ln(magn/rho). Hence any curve such that Phi(u,v) = constant is very special, for then the Big Omega (abnormality) term vanishes.

```
>
> shape11:=-simplify(gamma1&d(v)/d(u)&d(v));
> shape12:=-simplify(gamma1&d(u)/d(v)&d(u));
> shape21:=-simplify(gamma2&d(v)/d(u)&d(v));
> shape22:=-simplify(gamma2&d(u)/d(v)&d(u));
```

```
[ >
[ > SHAPE:=array([[shape11,shape12],[shape21,shape22]]):
[ > HH:=factor(trace(SHAPE)/2):
[ > print(`Mean Curvature is `,factor(HH));
[ > KK:=simplify(det(SHAPE)):
[ > print(`Gauss Curvature is `,factor(KK));
[ Choose q=1 (Gauss norm)
[ > HHgauss:=simplify(subs(q=1,HH));
[ > KKgauss:=simplify(subs(q=1,KK));
[ >
[ >
```

Note that the zeros of the Mean and Gauss curvatures do not depend upon the scaling coefficient. The abnormality Torsion does not effect flatness or minimal surfaces.

## CARTAN TORSION AND CURVATURE

Next compute the induced Cartan Torsion 2-forms and the Cartan Curvature 2-forms on the 2D subspace

```
[ > Sigma1:=simplify(omega^gamma1);Sigma2:=simplify(omega^gamma2);
[ > Phi1:=simplify(Omega^h1);Phi2:=simplify(Omega^h2);
[ >
[ > CURV2FORMS:=evalm(array([[simplify(h1^gamma1),simplify(h1^gamma2)],[factor(simplify(h2^gamma1)),simplify(h2^gamma2)]]));
```

## The Monge surface can also be considered as a implicit surface in 4D

A good exercise for the 4D version of a Repere Mobile.

The tangent vectors are computed algebraically.

```
[ > restart:
[ > # 3D IMPLICIT SURFACE defined by a Phi(x,y,z) = 0.
[ > with(liesyymm): setup(x,y,z,u):
[ > with (linalg):
```

The Monkey saddle as an implicit surface  $\phi(x,y,z)$  mapped to zero. Consider this as a projection from 4D

```
[ > a:=1;b:=1;c:=1;Phi:=z-x^3+3*x*y^2;
```

Construct the gradient of the surface function to give a 4 dimensional "surface gradient" vector in the space of variables  $\{x,y,z,u\}$ .

```
[ > Dr:=( [d(x),d(y),d(z),d(u)] ):ZZ:=grad(Phi,[x,y,z]);
```

Define an Action 1-form on the space of 4 variables. The first three components are the gradient of  $\phi$  (the implicit function which is set to zero to define the surface). The fourth component is the "magnitude" function. A first choice for the "magnitude" is the euclidean norm of the gradient. Other choices can be made. The choices which are homogeneous of degree 1 in the component functions of the gradient always lead to a shape matrix (3x3) with a zero determinant. Hence the resulting two surface is globally without a 3 volume.

```
[ > NN:=grad(Phi,[x,y,z,u]);magn:=simplify(innerprod(NN,NN)^(1/2),trig);ACTION:=wcollect(innerprod(NN,Dr)-magn*d(u));FIELD:=factor(wcollect(d(ACTION)));
```

```

[ > FT:=array([[1,0,0,NN[1]/magn],[0,1,0,NN[2]/magn],[0,0,1,NN[3]/magn],[-NN[1]/magn
,-NN[2]/magn,-NN[3]/magn,1]]):
The 4D frame is composed of the components of the algebraic vectors associated with the 1-form of Action.
note that the 44 component has been selected as unity. The fundamental 1-form has been "scaled" by
dividing through by magn. The N-1 = 3 tangent vectors are generated algebraically in terms of the
associated vectors which annihilate the 1-form of Action. The 4D Frame (Repere Mobile) is completed by
the components of the Action 1-form. This technique will be used to construct the Repere Mobile for more
general implicit surfaces, where the 1-form of Action does not necessarily yield a unique position vector (
the zero set of the Action may have multiple components.)
See http://www.uh.edu/~rkiehn/pdf/implicit.pdf
[ > FF:=transpose(FT);gmn:=innerprod(FT,FF);
[ > DET:=det(FF);FFINV:=inverse(FF):
[ The Projective Frame matrix so constructed is global as the Det is constant everywhere and non-zero.
[ > DR:=innerprod(FFINV,[d(x),d(y),d(z),d(u)]):sigma1:=(wcollect(factor(simplify(DR[
1]))));sigma2:=wcollect(DR[2]);sigma3:=wcollect(DR[3]);omega:=wcollect(DR[4]);
[ The forms sigma and omega are the 1-forms relative to the basis frame FF. Note that little omega is
proportional to the ACTION 1-form in 4D
[ > VV:=factor(simplify(det(FF),trig));
[ > dFF:=array([[d(FF[1,1]),d(FF[1,2]),d(FF[1,3]),d(FF[1,4]),[d(FF[2,1]),d(FF[2,2])
,d(FF[2,3]),d(FF[2,4]),[d(FF[3,1]),d(FF[3,2]),d(FF[3,3]),d(FF[3,4]),[d(FF[4,1]
),d(FF[4,2]),d(FF[4,3]),d(FF[4,4])]]):
[ > cartan:=evalm(FFINV*dFF);
[ > gamma11:=(wcollect(cartan[1,1]));
[ > gamma12:=(wcollect(cartan[1,2]));
[ > gamma13:=(wcollect(cartan[1,3]));
[ > gamma21:=(wcollect(cartan[2,1]));
[ > gamma22:=(wcollect(cartan[2,2]));
[ > gamma23:=(wcollect(cartan[2,3]));
[ > gamma31:=(wcollect(cartan[3,1]));
[ > gamma32:=(wcollect(cartan[3,2]));
[ > gamma33:=(wcollect(cartan[3,3]));
[ > # second fundamental form and shape matrix
[ > hh1:=(wcollect(factor(cartan[4,1])));
[ >
[ > gg1:=wcollect(simplify(factor(wcollect(cartan[1,4])),trig));
[ > hh2:=wcollect(cartan[4,2]);
[ > gg2:=(wcollect(factor(cartan[2,4])));
[ > hh3:=wcollect(cartan[4,3]);
[ > gg3:=(wcollect(factor(cartan[3,4])));
[ > Omega:=simplify(wcollect(cartan[4,4]));
[ Big Omega vanishes for the specific 4d euclidean norm chosen
[ > shape11:=-factor(gg1^(d(z)&d(y))/d(z)&d(y)&d(x));
[ > shape12:=-factor(gg1^(d(x)&d(z))/d(z)&d(y)&d(x));
[ > shape13:=-factor(gg1^(d(y)&d(x))/d(z)&d(y)&d(x));
[ > shape21:=-factor(gg2^(d(z)&d(y))/d(z)&d(y)&d(x));
[ > shape22:=-factor(gg2^(d(x)&d(z))/d(z)&d(y)&d(x));
[ > shape23:=-factor(gg2^(d(y)&d(x))/d(z)&d(y)&d(x));
[ > shape31:=-factor(gg3^(d(z)&d(y))/d(z)&d(y)&d(x));
[ > shape32:=-factor(gg3^(d(x)&d(z))/d(z)&d(y)&d(x));
[ > shape33:=-factor(gg3^(d(y)&d(x))/d(z)&d(y)&d(x));
[ > SHAPE:=evalm(array([[shape11,shape12,shape13],[shape21,shape22,shape23],[shape31
,shape32,shape33]]));

```

The Shape matrix computed from the Cartan matrix of 1-forms (for the 3D euclidean norm) is precisely the same as the Jacobian of the 3D normalized gradient field. (The 3D gauss map) The similarity invariants are the same. The determinant vanishes (as is to be expected for the 2-surface).

The other two invariants, the trace and the trace of the adjoint of the shape matrix give the mean and the gauss curvature.

```
> DETSHAPE:=simplify(det(SHAPE),trig);determinant_of_the_4D_frame_matrix:=simplify(VV,trig);
```

The mean curvature is the Trace of the Shape matrix (a 3D matrix). The determinant vanishes, and the Gauss curvature is the trace of the adjoint of the Shape matrix. All of these quantities are similarity invariants of the shape matrix.

```
>
> HHT:=factor(simplify(trace(SHAPE)/2,trig));print(`Mean Curvature is `,HHT);
```

```
>
> ADJSH:=adjoint(SHAPE);
> KK:=factor(simplify(trace(ADJSH),trig));
```

```
> print(`Gauss Curvature is `,KK);
```

Compare to the previous computations for  $q = 1$ .

```
>
>
> restart;
>
>
```

## Special Case: Ruled SURFACES in 3D (An Elementary Version of String Theory)

A ruled surface is a doubly parameterized surface of the type constructed from two "curves": the lines can be generated by two singly parameterized vectors such as the position vector and its velocity field. A typical method is to define the a curve  $XX(u)$  and another curve  $YY(u)$ . Then the position vector is defined as the parametric combination,  $R(u,v) = XX(u)+v*YY(u)$ . The vector  $YY$  is defined as the directrix. The example below will define the first curve as a circle in the xy plane, and the directrix as a line meeting the circle at a constant angle. The result is a hyperbolic surface with twisted straight line generators. The surface is said to be ruled. Some ruled surfaces can be rolled out flat; these are called developables. All ruled surfaces for which the Gauss curvature is zero are said to be developable surfaces. The general ruled surface is often called a scroll.

A connection to physical systems is given by the W. Gibbs statement that equilibrium thermodynamic surfaces of mixed phase are ruled surfaces.

```
> restart:with(linalg):with(plots):with(diffforms):with(liesymm): setup(u,v):
> XX:=evalm([a*cos(u),a*sin(u),0]);YY:=evalm([-b*sin(u),b*cos(u),c]);
```



```
[ >
[ > R:=evalm(evalm(XX)+evalm(v*YY));dR:=(evalm([wcollect(d(R[1]),wcollect(d(R[2])),
[ wcollect(d(R[3]))]));
[ > SURF:=(subs(a=1,b=2.5,c=1.0,[R[1],R[2],R[3]]));
[ > plot3d(SURF,u=-2..2,v=-2..2,numpoints=4000,style=PATCHCONTOUR,axes=NORMAL,shadin
[ g=ZGREYSCALE,orientation=[110,65]);
```

Note that the surface has linear straight line generators that are "twisted". However, the envelope of this twisted set of generators does not reveal whether the twist is right handed or left handed. The fact that generators are straight lines and the surface is not a cylinder says something about the fact that there is a twist. The problem is how to determine this fact from a parametric representation.

```
[ >
[ Define two tangent vectors, Yu and Yv
```

```
[ > Yu:=[diff(R[1],u),diff(R[2],u),diff(R[3],u)];
[ > Yv:=[diff(R[1],v),diff(R[2],v),diff(R[3],v)];
```

```
[ > F2:=array([[Yu[1],Yv[1],0],[Yu[2],Yv[2],0],[Yu[3],Yv[3],1] ] );
```

Use the Grassman adjoint method to define the "adjoint" field. (The Gibbs cross product in 3D)

```
[ > Grass:=(transpose(adjoint(F2)));det(Grass);
[ > NNU:=evalm([Grass[1,3],Grass[2,3],Grass[3,3]]);
```

Scale the (adjoint) normal field here by rho (the particular choice shown below will eliminate and abnormality Torsion coefficients.)

```
[ > magn:=(simplify(innerprod(NNU,NNU),trig))^(1/2);
```

```
[ >
[ > NN:=evalm([factor(NNU[1]),factor(NNU[2]),simplify(factor(NNU[3]))]);
```

Construct the Cartan Frame matrix from the two "tangent" vectors and the adjoint normal:

```
[ > #rho(u,v):=magn;
[ >
[ > FF:=array([[Yu[1],Yv[1],NN[1]/rho(u,v)],[Yu[2],Yv[2],NN[2]/rho(u,v)],[Yu[3],Yv[3]
[ ],NN[3]/rho(u,v)] ] );
```

The Repere Mobile or FRAME MATRIX, FF. note that the frame matrix is not orthonormal!!

```
[ > detFF:=simplify((det(FF)));
[ > Gmn:=(innerprod(transpose(FF),FF));G33:=simplify(Gmn[3,3],trig);
```

For the case at hand, the determinant is non-zero globally, hence an inverse always exists.

```
[ >
[ > FFINVD:=evalm(FF^(-1));
```

The 1-form components of the differential position vector with respect to the Basis Frame, F.

```
[ > dRF:=innerprod(FFINVD,[dR[1],dR[2],dR[3]]);
[ > sigma1:=(wcollect(dRF[1]));
[ > sigma2:=(wcollect(dRF[2]));
```

Note that sigma1 is du and sigma2 is dv for a parametric Monge surfaces!!

```
[ > omega:=(wcollect(dRF[3]));
```

Note that this term vanishes for a parametric Monge surface which is homogeneous of degree 1,!! , hence parametric Monge surfaces exhibit no TORSION!! of the Affine type ( that is there is no translational shear defects!)

```
[ Compute the Cartan Matrix of connection forms from C=[F(inverse)] times d[F]
```

```
[ > dFF:=array([[d(FF[1,1]),d(FF[1,2]),d(FF[1,3])],[d(FF[2,1]),d(FF[2,2]),d(FF[2,3])
],[d(FF[3,1]),d(FF[3,2]),d(FF[3,3])]]);
[ > cartan:=evalm(FFINVD*dFF):
```

[ The interior connection coefficients (can be Christoffel symbols on the parameter space

```
[ > Gamma11:=wcollect(simplify(wcollect(cartan[1,1]),trig));
[ > Gamma12:=wcollect(simplify(wcollect(cartan[1,2]),trig));
[ > Gamma21:=wcollect(simplify(wcollect(cartan[2,1]),trig));
[ > Gamma22:=wcollect(simplify(wcollect(cartan[2,2]),trig));
```

[ The second fundamental form or shape matrix comes from the third row of the Cartan matrix

```
[ > h1:=wcollect(simplify(wcollect(cartan[3,1]),trig));
[ > gamma1:=wcollect(simplify(wcollect(cartan[1,3]),trig));
[ > h2:=factor((wcollect(cartan[3,2])));
[ > gamma2:=factor(wcollect(cartan[2,3]));
```

[ Next compute the Expansion Twist:

The expansion twist for the parametric surface will show up as a non-zero entry in the [3,3] slot of the Cartan Matrix. Big Omega is always an exact differential for parametric and Monge surfaces. The expansion twist is not zero unless the euclidean scaling is subsumed.

Therefore implicit Monge surfaces will admit disclination defects (Torsion of the second kind due to rotations)

```
[ > Omega:=wcollect(simplify(wcollect(cartan[3,3]),trig));
[ > Phi(u,v):=simplify(ln((magn)/rho(u,v)),trig);Omegacheck:=wcollect(factor(d(Phi(u
,v)))));
```

```
[ > shape11:=-factor(gamma1*d(v)/d(u)&d(v));
[ > shape12:=-factor(gamma1*d(u)/d(v)&d(u));
[ > shape21:=-factor(gamma2*d(v)/d(u)&d(v));
[ > shape22:=-factor(gamma2*d(u)/d(v)&d(u));
[ >
[ > SHAPE:=array([[shape11,shape12],[shape21,shape22]]):
[ > HH:=factor(trace(SHAPE)/2):
[ > print(`Mean Curvature is `,factor(HH));
[ > KK:=simplify(det(SHAPE)):
[ > print(`Gauss Curvature is `,factor(KK));
```

[ The position vector to this surface generates out a smooth surface that does not describe the fact the the generators of this surface are twisted. And the twisted generators tell nothing about the Big Omega !!

[ The scaling of the normal field generates a non-zero Big Omega but tells nothing of the twisted generators.

[ >

[ Note that the zeros of the Mean and Gauss curvatures do not depend upon the scaling coefficient. The abnormality Torsion does not effect flatness or minimal surfaces.

## CARTAN TORSION AND CURVATURE

Next compute the induced Cartan Torsion 2-forms and the Cartan Curvature 2-forms on the 2D subspace

```
[ > Sigma1:=simplify(omega^gamma1);Sigma2:=simplify(omega^gamma2);
[ > Phi1:=simplify(Omega^h1);Phi2:=simplify(Omega^h2);
[ >
[ > CURV2FORMS:=evalm(array([[simplify(h1^gamma1),simplify(h1^gamma2)],[factor(sim
plify(h2^gamma1)),simplify(h2^gamma2)]]));
```

## EXAMPLE: Whitney Umbrella

```
[ >
[ > restart:with(linalg):with(plots):with(diffforms):with(liesymm): setup(u,v):
Warning, new definition for norm
(The complete output is more than 30 pages. It can be printed by changing the : to a ; at the end of each
program statement)
[ >
[ Define a position vector in R3 as three arbitrary functions U,V,W parametrized by two variables u,v
[ > U(u,v):=(u*v);
[ > V(u,v):=(u);
[ > W(u,v):=(v^2); W(u,v) := sin(1/2 u) sin(v) + cos(1/2 u) sin(2 v)
[ > R:=[U(u,v),V(u,v),W(u,v)]; dR:=evalm(d(R));
[ Insert specific examples here:
[ >
[ Define two tangent vectors, Yu and Yv
[ > Yu:=diff(R,u);
[ > Yv:=diff(R,v);
[ For a given choice of R make a 3D plot.
[ > plot3d(subs(a=2,R),u=-1..1,v=-1..1,orientation=[-150,50],numpoints=2000,style=PA
TCHNOGRID,axes=NORMAL,title=`Whitney Umbrella`,scaling=CONSTRAINED);
[ Use the cross product to define a third independent basis vector for the Cartan frame.
[ >
[ > NNU:=crossprod(Yu,Yv):
[ Scale the (adjoint) normal field here by rho
[ > rho(u,v):=innerprod(NNU,NNU)^(1/2):
[ The choice of the scaling factor rho will determine the abnormality torsion coefficients of the system. As
the position vector is parametrically defined, the 1-form, small omega, vanishes identically, and therefore
there is no affine (translational dislocation shear) torsion for the parametric surfaces. However, the
rotational shear disclination torsion coefficients exist unless the normal field is scaled by the quadratic
factor given above (the Gauss map). Note the scaling could have been chosed to leave the Frame matrix
unimodular or even involutory!
[ > #rho:=1;
[ >
[ > NN:=evalm([factor(NNU[1]),factor(NNU[2]),simplify(factor(NNU[3]))]):
[ Construct the Cartan Frame matrix from the two "tangent" vectors and the adjoint normal:
[ > FF:=array([[Yu[1],Yv[1],NN[1]/rho(u,v)],[Yu[2],Yv[2],NN[2]/rho(u,v)],[Yu[3],Yv[3]
],NN[3]/rho(u,v) ]]):
[ The Repere Mobile or FRAME MATRIX, FF. note that the frame matrix is not orthonormal!!
[ > detFF:=simplify((det(FF))):
[ For the case at hand, the determinant is non-zero globally, hence an inverse always exists.
[ > FFINVD:=evalm(FF^(-1)):
```

```

[ The 1-form components of the differential position vector with respect to the Basis Frame, F.
[ > dRF:=innerprod(FFINVD,[dR[1],dR[2],dR[3]]):
[ > sigma1:=(wcollect(dRF[1]));
[ > sigma2:=(wcollect(dRF[2]));
[ Note that sigma1 is du and sigma2 is dv for a parametric Monge surfaces!!
[ > omega:=(wcollect(dRF[3]));
[ Note that this term vanishes for a parametric Monge surface which is homogeneous of degree 1,!!
, hence parametric Monge surfaces exhibit no TORSION!! of the Affine type ( that is there is no
[ translational shear defects!)
[ Compute the Cartan Matrix of connection forms from C=[F(inverse)] times d[F]
[ > dFF:=array([[d(FF[1,1]),d(FF[1,2]),d(FF[1,3])],[d(FF[2,1]),d(FF[2,2]),d(FF[2,3])
[ ],[d(FF[3,1]),d(FF[3,2]),d(FF[3,3])]]):
[ > cartan:=evalm(FFINVD*dFF):
[ The interior connection coefficients (can be Christoffel symbols on the parameter space
[ > Gamma11:=(wcollect(cartan[1,1]));
[ > Gamma12:=(wcollect(cartan[1,2]));
[ > Gamma21:=(wcollect(cartan[2,1]));
[ > Gamma22:=(wcollect(cartan[2,2]));
[ The second fundamental form or shape matrix comes from the third row of the Cartan matrix
[ > h1:=wcollect(cartan[3,1]):
[ > gamma1:=(wcollect(cartan[1,3]));
[ > h2:=wcollect(cartan[3,2]):
[ > gamma2:=(wcollect(cartan[2,3]));
[ Next compute the twist (BigOmega):
[ The abnormality for the parametric surface will show up as a non-zero entry in the [3,3] slot of the Cartan
[ Matrix. Big Omega is always an exact differential for parametric and Monge surfaces. Big Omega is not
[ zero unless the euclidean scaling is subsumed.
[ Therefore implicit Monge surfaces will admit twist defects (Torsion of the second kind due to rotations)
[ > Omega:=(wcollect(factor((cartan[3,3]))));
[ (Omega vanishes for the euclidean type of normalization.
[
[ > shape11:=-factor(gamma1*d(v)/d(u)&d(v)):
[ > shape12:=-factor(gamma1*d(u)/d(v)&d(u)):
[ > shape21:=-factor(gamma2*d(v)/d(u)&d(v)):
[ > shape22:=-factor(gamma2*d(u)/d(v)&d(u)):
[ >
[ > SHAPE:=array([[shape11,shape12],[shape21,shape22]]);
[ > HH:=factor(trace(SHAPE)/2):
[ > print(`Mean Curvature is `,factor(HH));
[ > KK:=simplify(det(SHAPE)):
[ > print(`Gauss Curvature is `,factor(KK));
[ >
[ Note that the zeros of the Mean and Gauss curvatures do not depend upon the scaling coefficient. The
[ abnormality Torsion does not effect flatness or minimal surfaces.

```

## CARTAN TORSION AND CURVATURE

Next compute the induced Cartan Torsion 2-forms and the Cartan Curvature 2-forms on the 2D subspace

```
[ > Sigma1:=simplify(omega^gamma1);Sigma2:=simplify(omega^gamma2);
[ > Phi1:=simplify(Omega^h1);Phi2:=simplify(Omega^h2);
[ >
[ > CURV2FORMS:=evalm(array([[simplify(h1^gamma1),simplify(h1^gamma2)],[factor(simplify(h2^gamma1)),simplify(h2^gamma2)]]));
[ >
[ >
```

\*\*\*\*\*

## General Case Parametric Surfaces

```
[ > restart:with(linalg):with(plots):with(diffforms):with(liesymm): setup(u,v):
[ > deform(u=0,v=0,U=0,V=0,W=0);
[ >
[ >
[ Define a position vector in R3 as three arbitrary functions U,V,W parametrized by two variables u,v
[ > R:=[U(u,v),V(u,v),W(u,v)];dR:=[diff(U(u,v),u)&^d(u)+diff(U(u,v),v)&^d(v),diff(V(u,v),u)&^d(u)+diff(V(u,v),v)&^d(v),diff(W(u,v),u)&^d(u)+diff(W(u,v),v)&^d(v)];
```

$$R := [U(u, v), V(u, v), W(u, v)]$$

$$dR := \left[ \left( \frac{\partial}{\partial u} U(u, v) \right) d(u) + \left( \frac{\partial}{\partial v} U(u, v) \right) d(v), \left( \frac{\partial}{\partial u} V(u, v) \right) d(u) + \left( \frac{\partial}{\partial v} V(u, v) \right) d(v), \left( \frac{\partial}{\partial u} W(u, v) \right) d(u) + \left( \frac{\partial}{\partial v} W(u, v) \right) d(v) \right]$$

Insert specific examples here:

```
[ >
[ Define two tangent vectors, Yu and Yv
```

```
[ > Yu:=diff(R,u);
```

$$Y_u := \left[ \frac{\partial}{\partial u} U(u, v), \frac{\partial}{\partial u} V(u, v), \frac{\partial}{\partial u} W(u, v) \right]$$

```
[ > Yv:=diff(R,v);
```

$$Y_v := \left[ \frac{\partial}{\partial v} U(u, v), \frac{\partial}{\partial v} V(u, v), \frac{\partial}{\partial v} W(u, v) \right]$$

For a given choice of R make a 3D plot.

```
[ > #plot3d(R,u=-1.3..1.3,v=-1.3..1.3,orientation=[134,86],numpoints=5000,style=PATC H);
```

Use the cross product to define a third independent basis vector for the Cartan frame.

```
[ >
[ > NNU:=crossprod(Yu,Yv):
```

Scale the (adjoint) normal field here by rho (the following computation uses the Gauss norm)

```
[ > rho(u,v):=innerprod(NNU,NNU)^(q/2):
```

The choice of the scaling factor rho will determine the abnormality torsion coefficients of the system. As the position vector is parametrically defined, the 1-form, small omega, vanishes identically, and therefore there is no affine (translational dislocation shear) torsion for the parametric surfaces. However, the rotational shear disclination torsion coefficients exist unless the normal field is scaled by the quadratic

factor given above (the Gauss map). Note the scaling could have been chosen to leave the Frame matrix unimodular or even involutory!

```
[ >
[ > NN:=evalm([factor(NNU[1]),factor(NNU[2]),simplify(factor(NNU[3]))]):
[ Construct the Cartan Frame matrix from the two "tangent" vectors and the adjoint normal:
[ > FF:=array([[Yu[1],Yv[1],NN[1]/rho(u,v)],[Yu[2],Yv[2],NN[2]/rho(u,v)],[Yu[3],Yv[3]
[   ],NN[3]/rho(u,v)  ] ]):
```

[ The Repere Mobile or FRAME MATRIX, FF. note that the frame matrix is not orthonormal!!

```
[ > detFF:=simplify(det(FF)):
[ For the case at hand, the determinant is non-zero globally, hence an inverse always exists.
[ > FFINVD:=evalm(FF^(-1)):
[ The 1-form components of the differential position vector with respect to the Basis Frame, F.
[ > dRF:=innerprod(FFINVD,[dR[1],dR[2],dR[3]]);
[                                     dRF := [d(u), d(v), 0]
[ > sigma1:=(wcollect(dRF[1]));
[                                     sigma1 := d(u)
[ > sigma2:=(wcollect(dRF[2]));
[                                     sigma2 := d(v)
[ Note that sigma1 is du and sigma2 is dv for a parametric Monge surfaces!!
[ > omega:=(wcollect(dRF[3]));
[                                     omega := 0
```

**Note that this term vanishes for a parametric surface !!  
, hence parametric surfaces exhibit no TORSION!! of the Affine type  
( that is there is no translational shear defects!)**

[ Compute the Cartan Matrix of connection forms from C=[F(inverse)] times d[F]

```
[ >
[ >
[ > dFF:=array([[innerprod(grad(FF[1,1],[u,v]),[d(u),d(v)]),innerprod(grad(FF[1,2],[u,v]),[d(u),d(v)]),innerprod(grad(FF[1,3],[u,v]),[d(u),d(v)]),[innerprod(grad(FF[2,1],[u,v]),[d(u),d(v)]),innerprod(grad(FF[2,2],[u,v]),[d(u),d(v)]),innerprod(grad(FF[2,3],[u,v]),[d(u),d(v)]),[innerprod(grad(FF[3,1],[u,v]),[d(u),d(v)]),innerprod(grad(FF[3,2],[u,v]),[d(u),d(v)]),innerprod(grad(FF[3,3],[u,v]),[d(u),d(v)]))] ]]):
[ > cartan:=evalm(FFINVD*dFF):
```

[ The interior connection coefficients (can be Christoffel symbols on the parameter space

```
[ > Gamma11:=(wcollect(cartan[1,1])):
[ > Gamma12:=(wcollect(cartan[1,2])):
[ > Gamma21:=(wcollect(cartan[2,1])):
[ > Gamma22:=(wcollect(cartan[2,2])):
```

[ The second fundamental form or shape matrix comes from the third row of the Cartan matrix

```
[ > h1:=simplify(cartan[3,1]):
[ > gamma1:=(wcollect(cartan[1,3])):
[ > h2:=(wcollect(cartan[3,2])):
[ > gamma2:=(wcollect(cartan[2,3])):
```

[ Next compute the twist (BigOmega):

The abnormality for the parametric surface will show up as a non-zero entry in the [3,3] slot of the Cartan Matrix. Big Omega is always an exact differential for parametric and Monge surfaces. Big Omega is not zero unless the euclidean scaling is subsumed.

Therefore implicit Monge surfaces will admit twist defects (Torsion of the second kind due to rotations if  $q < 1$ )

> **Omega := ((factor(simplify(wcollect(cartan[3,3])))));**

$$\begin{aligned}
\Omega := & - \left( (q-1) \left( d(v) \left( \frac{\partial^2}{\partial v \partial u} U(u, v) \right) \left( \frac{\partial}{\partial v} W(u, v) \right)^2 \left( \frac{\partial}{\partial u} U(u, v) \right) \right. \right. \\
& + \left( \frac{\partial}{\partial u} W(u, v) \right) \left( \frac{\partial}{\partial v} V(u, v) \right)^2 d(v) \left( \frac{\partial^2}{\partial v \partial u} W(u, v) \right) + \left( \frac{\partial}{\partial v} U(u, v) \right)^2 \left( \frac{\partial}{\partial u} V(u, v) \right) d(u) \left( \frac{\partial^2}{\partial u^2} V(u, v) \right) \\
& + \left( \frac{\partial}{\partial v} U(u, v) \right) \left( \frac{\partial}{\partial u} W(u, v) \right)^2 d(v) \left( \frac{\partial^2}{\partial v^2} U(u, v) \right) + \left( \frac{\partial}{\partial u} V(u, v) \right)^2 \left( \frac{\partial}{\partial v} W(u, v) \right) d(v) \left( \frac{\partial^2}{\partial v^2} W(u, v) \right) \\
& + \left( \frac{\partial}{\partial v} U(u, v) \right)^2 \left( \frac{\partial}{\partial u} V(u, v) \right) d(v) \left( \frac{\partial^2}{\partial v \partial u} V(u, v) \right) + \left( \frac{\partial}{\partial v} U(u, v) \right)^2 \left( \frac{\partial}{\partial u} W(u, v) \right) d(v) \left( \frac{\partial^2}{\partial v \partial u} W(u, v) \right) \\
& + d(u) \left( \frac{\partial}{\partial u} U(u, v) \right)^2 \left( \frac{\partial^2}{\partial v \partial u} W(u, v) \right) \left( \frac{\partial}{\partial v} W(u, v) \right) + \left( \frac{\partial}{\partial v} U(u, v) \right)^2 \left( \frac{\partial}{\partial u} W(u, v) \right) d(u) \left( \frac{\partial^2}{\partial u^2} W(u, v) \right) \\
& + d(u) \left( \frac{\partial^2}{\partial u^2} U(u, v) \right) \left( \frac{\partial}{\partial v} W(u, v) \right)^2 \left( \frac{\partial}{\partial u} U(u, v) \right) + \left( \frac{\partial}{\partial v} U(u, v) \right) d(u) \left( \frac{\partial^2}{\partial v \partial u} U(u, v) \right) \left( \frac{\partial}{\partial u} V(u, v) \right)^2 \\
& + \left( \frac{\partial}{\partial u} U(u, v) \right) \left( \frac{\partial}{\partial v} V(u, v) \right)^2 d(v) \left( \frac{\partial^2}{\partial v \partial u} U(u, v) \right) + \left( \frac{\partial}{\partial u} U(u, v) \right)^2 \left( \frac{\partial}{\partial v} V(u, v) \right) d(v) \left( \frac{\partial^2}{\partial v^2} V(u, v) \right) \\
& + \left( \frac{\partial}{\partial u} V(u, v) \right) \left( \frac{\partial}{\partial v} W(u, v) \right)^2 d(v) \left( \frac{\partial^2}{\partial v \partial u} V(u, v) \right) + d(v) \left( \frac{\partial}{\partial u} U(u, v) \right)^2 \left( \frac{\partial^2}{\partial v^2} W(u, v) \right) \left( \frac{\partial}{\partial v} W(u, v) \right) \\
& + \left( \frac{\partial}{\partial u} W(u, v) \right) \left( \frac{\partial}{\partial v} V(u, v) \right)^2 d(u) \left( \frac{\partial^2}{\partial u^2} W(u, v) \right) + \left( \frac{\partial}{\partial u} W(u, v) \right)^2 \left( \frac{\partial}{\partial v} V(u, v) \right) d(u) \left( \frac{\partial^2}{\partial v \partial u} V(u, v) \right) \\
& + \left( \frac{\partial}{\partial u} W(u, v) \right)^2 \left( \frac{\partial}{\partial v} V(u, v) \right) d(v) \left( \frac{\partial^2}{\partial v^2} V(u, v) \right) + \left( \frac{\partial}{\partial v} U(u, v) \right) d(v) \left( \frac{\partial^2}{\partial v^2} U(u, v) \right) \left( \frac{\partial}{\partial u} V(u, v) \right)^2 \\
& + \left( \frac{\partial}{\partial v} U(u, v) \right) \left( \frac{\partial}{\partial u} W(u, v) \right)^2 d(u) \left( \frac{\partial^2}{\partial v \partial u} U(u, v) \right) + \left( \frac{\partial}{\partial u} V(u, v) \right)^2 \left( \frac{\partial}{\partial v} W(u, v) \right) d(u) \left( \frac{\partial^2}{\partial v \partial u} W(u, v) \right) \\
& + \left( \frac{\partial}{\partial u} V(u, v) \right) \left( \frac{\partial}{\partial v} W(u, v) \right)^2 d(u) \left( \frac{\partial^2}{\partial u^2} V(u, v) \right) + \left( \frac{\partial}{\partial u} U(u, v) \right)^2 \left( \frac{\partial}{\partial v} V(u, v) \right) d(u) \left( \frac{\partial^2}{\partial v \partial u} V(u, v) \right) \\
& + \left( \frac{\partial}{\partial u} U(u, v) \right) \left( \frac{\partial}{\partial v} V(u, v) \right)^2 d(u) \left( \frac{\partial^2}{\partial u^2} U(u, v) \right) - \left( \frac{\partial}{\partial v} U(u, v) \right) \left( \frac{\partial}{\partial u} W(u, v) \right) d(u) \left( \frac{\partial}{\partial u} U(u, v) \right) \left( \frac{\partial^2}{\partial v \partial u} W(u, v) \right) \\
& - \left( \frac{\partial}{\partial u} V(u, v) \right) \left( \frac{\partial}{\partial v} W(u, v) \right) d(v) \left( \frac{\partial^2}{\partial v \partial u} W(u, v) \right) \left( \frac{\partial}{\partial v} V(u, v) \right) \\
& - d(u) \left( \frac{\partial}{\partial u} U(u, v) \right) \left( \frac{\partial}{\partial v} V(u, v) \right) \left( \frac{\partial}{\partial u} V(u, v) \right) \left( \frac{\partial^2}{\partial v \partial u} U(u, v) \right) \\
& - \left( \frac{\partial}{\partial u} V(u, v) \right) \left( \frac{\partial}{\partial v} W(u, v) \right) d(u) \left( \frac{\partial^2}{\partial u^2} W(u, v) \right) \left( \frac{\partial}{\partial v} V(u, v) \right) \\
& - d(v) \left( \frac{\partial}{\partial u} U(u, v) \right) \left( \frac{\partial}{\partial v} V(u, v) \right) \left( \frac{\partial}{\partial u} V(u, v) \right) \left( \frac{\partial^2}{\partial v^2} U(u, v) \right) \\
& - \left( \frac{\partial}{\partial v} U(u, v) \right) \left( \frac{\partial}{\partial u} U(u, v) \right) \left( \frac{\partial}{\partial v} V(u, v) \right) d(u) \left( \frac{\partial^2}{\partial u^2} V(u, v) \right) \\
& - \left( \frac{\partial}{\partial u} W(u, v) \right) d(v) \left( \frac{\partial^2}{\partial v^2} U(u, v) \right) \left( \frac{\partial}{\partial u} U(u, v) \right) \left( \frac{\partial}{\partial v} W(u, v) \right)
\end{aligned}$$





(Omega vanishes for the Gaussian type of normalization. q=1,p=2

```
[ > shape11:=-factor(gamma1&^d(v)/d(u)&^d(v)):
[ > shape12:=-factor(gamma1&^d(u)/d(v)&^d(u)):
[ > shape21:=-factor(gamma2&^d(v)/d(u)&^d(v)):
[ > shape22:=-factor(gamma2&^d(u)/d(v)&^d(u)):
[ >
[ > SHAPE:=array([[shape11,shape12],[shape21,shape22]]):
[ > HH:=factor(simplify(trace(SHAPE)/2)):
```

(The general formula could be factored better. The massive formulas below are somewhat useless and have not been printed out)

```
[ > #print(`Mean Curvature is `,factor(HH)):
[ > KK:=simplify(det(SHAPE)):
```

(The general formula could be factored better. The massive formulas below are somewhat useless and have not been printed out)

```
[ > #print(`Gauss Curvature is `,factor(KK)):
[ >
[ >
```

## CARTAN TORSION AND CURVATURE

Next compute the induced Cartan Torsion 2-forms and the Cartan Curvature 2-forms on the 2D subspace.

**Note that in that which follows the Particle Affine torsion 2-forms go to zero as little omega vanishes. The Wave Affine torsion 2-forms go to zero due to the Gauss normalization. (No expansion)**

```
> Sigma1:=simplify(omega&^gamma1);Sigma2:=simplify(omega&^gamma2);
          Sigma1 := 0
          Sigma2 := 0
> Phi1:=simplify(Omega&^h1);Phi2:=simplify(Omega&^h2);
```

$$\begin{aligned} \Phi 1 := & (q-1) \left( \left( \frac{\partial}{\partial u} V(u, v) \right)^2 \left( \frac{\partial}{\partial v} W(u, v) \right)^2 - 2 \left( \frac{\partial}{\partial u} V(u, v) \right) \left( \frac{\partial}{\partial v} W(u, v) \right) \left( \frac{\partial}{\partial u} W(u, v) \right) \left( \frac{\partial}{\partial v} V(u, v) \right) \right. \\ & + \left( \frac{\partial}{\partial u} W(u, v) \right)^2 \left( \frac{\partial}{\partial v} V(u, v) \right)^2 + \left( \frac{\partial}{\partial u} W(u, v) \right)^2 \left( \frac{\partial}{\partial v} U(u, v) \right)^2 \\ & - 2 \left( \frac{\partial}{\partial u} W(u, v) \right) \left( \frac{\partial}{\partial v} U(u, v) \right) \left( \frac{\partial}{\partial u} U(u, v) \right) \left( \frac{\partial}{\partial v} W(u, v) \right) + \left( \frac{\partial}{\partial u} U(u, v) \right)^2 \left( \frac{\partial}{\partial v} W(u, v) \right)^2 \\ & + \left( \frac{\partial}{\partial u} U(u, v) \right)^2 \left( \frac{\partial}{\partial v} V(u, v) \right)^2 - 2 \left( \frac{\partial}{\partial u} U(u, v) \right) \left( \frac{\partial}{\partial v} V(u, v) \right) \left( \frac{\partial}{\partial u} V(u, v) \right) \left( \frac{\partial}{\partial v} U(u, v) \right) \\ & + \left( \frac{\partial}{\partial u} V(u, v) \right)^2 \left( \frac{\partial}{\partial v} U(u, v) \right)^2 \left. \right)^{-2+1/2q} \left( \right. \\ & - 2 \left( \frac{\partial}{\partial u} W(u, v) \right)^2 \left( \frac{\partial}{\partial v} V(u, v) \right) \left( \frac{\partial^2}{\partial v^2} V(u, v) \right) \left( \frac{\partial}{\partial u} V(u, v) \right) \left( \frac{\partial}{\partial v} W(u, v) \right) \left( \frac{\partial^2}{\partial u^2} U(u, v) \right) \\ & + \left( \frac{\partial}{\partial v} U(u, v) \right)^2 \left( \frac{\partial}{\partial u} W(u, v) \right) \left( \frac{\partial}{\partial u} U(u, v) \right) \left( \frac{\partial^2}{\partial v \partial u} W(u, v) \right) \left( \frac{\partial}{\partial u} V(u, v) \right) \\ & \left. + \left( \frac{\partial}{\partial u} V(u, v) \right)^2 \left( \frac{\partial}{\partial v} W(u, v) \right)^2 \left( \frac{\partial^2}{\partial u^2} V(u, v) \right) \left( \frac{\partial^2}{\partial v \partial u} U(u, v) \right) \right) \end{aligned}$$







































$$\begin{aligned}
& - \left( \frac{\partial^2}{\partial v \partial u} U(u, v) \right) \left( \frac{\partial}{\partial v} W(u, v) \right)^2 \left( \frac{\partial}{\partial u} U(u, v) \right)^2 \left( \frac{\partial}{\partial v} V(u, v) \right) \left( \frac{\partial^2}{\partial v \partial u} W(u, v) \right) \\
& + \left( \frac{\partial}{\partial v} U(u, v) \right)^2 \left( \frac{\partial}{\partial u} W(u, v) \right)^3 \left( \frac{\partial^2}{\partial v \partial u} U(u, v) \right) \left( \frac{\partial^2}{\partial v^2} V(u, v) \right) \left( d(v) \wedge d(u) \right)
\end{aligned}$$

[ **The Wave Affine torsion 2-forms vanish if  $q = 1$  (no expansions or contractions)**

[ The massive formulas below have not been printed out.

[ > **CURV2FORMS:=evalm(array([[simplify(h1&^gamma1),simplify(h1&^gamma2)],[factor(simplify(h2&^gamma1)),simplify(h2&^gamma2)]])):**

[ THE END

[ copyright CSDC Inc. 1999

[ >