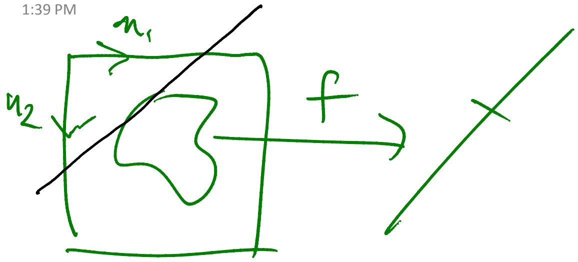
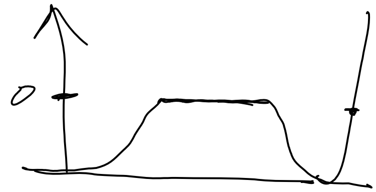
Implicit Curves

Friday, February 01, 2008
1:39 PM



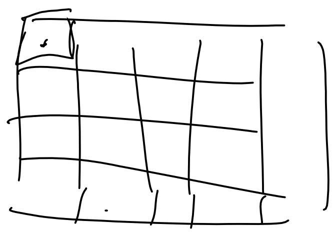


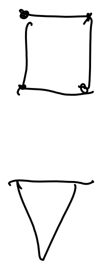
$$C = \{(u_1, u_2) : f(u_1, u_2) = 0\}$$

Set of zeros of a continuous function

If the function is zero on an open set, the "curve" is no curve-like

$$f(u_1, u_2) = (u_1 - 1)^2 + (u_2 - 1)^2 - 1$$





Gradient & Normal

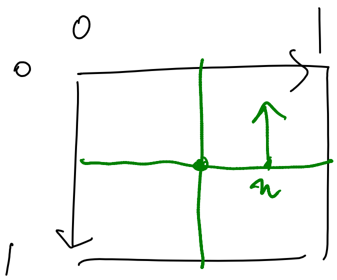
Friday, February 01, 2008
2:11 PM

Function cannot be constant on an open set

$$f(u_1, u_2) = (u_1 - \tfrac{1}{2})(u_2 - \tfrac{1}{2})$$

$$C_L = \{u : f(u) = L\}$$

If function is differentiable, a sufficient condition for the curve to be "manifold" is that its gradient be non-zero on every point of the curve



$$\nabla f(u) = \begin{pmatrix} \dfrac{\partial f}{\partial u_1} \\[2ex] \dfrac{\partial f}{\partial u_2} \end{pmatrix}$$

Points of the implicit curve where the gradient is zero are "singular" points

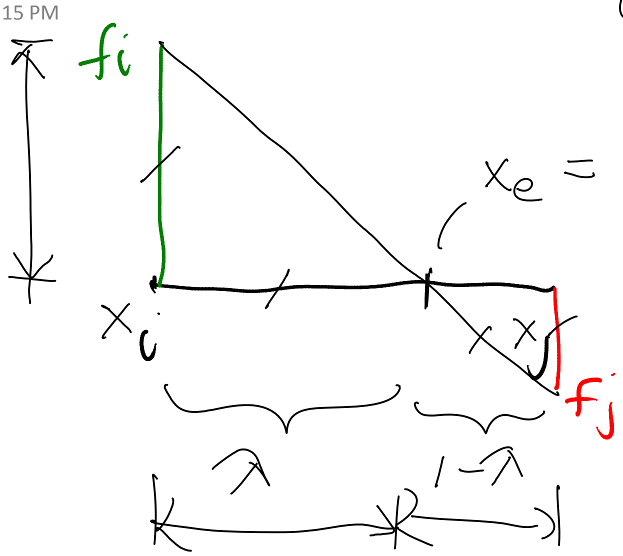
$$\nabla f(u) \perp C$$

The gradient vector is perpendicular to the curve

Linear Interpolation Along Edge

Friday, February 01, 2008
2:15 PM

$0 < \lambda < 1$



$$x_e = \lambda x_j + (1-\lambda) x_i$$

$$\lambda = \frac{\|x_e - x_i\|}{\|x_j - x_i\|}$$

$$\Downarrow$$

$$\|x_e - x_i\| = \lambda \|x_j - x_i\|$$

$$\|x_e - x_j\| = (1-\lambda) \|x_j - x_i\|$$

$$\frac{f_i}{\|x_e - x_i\|} = \frac{-f_j}{\|x_e - x_j\|}$$

$$\Downarrow$$

$$\frac{f_i}{\lambda} = \frac{-f_j}{1-\lambda}$$

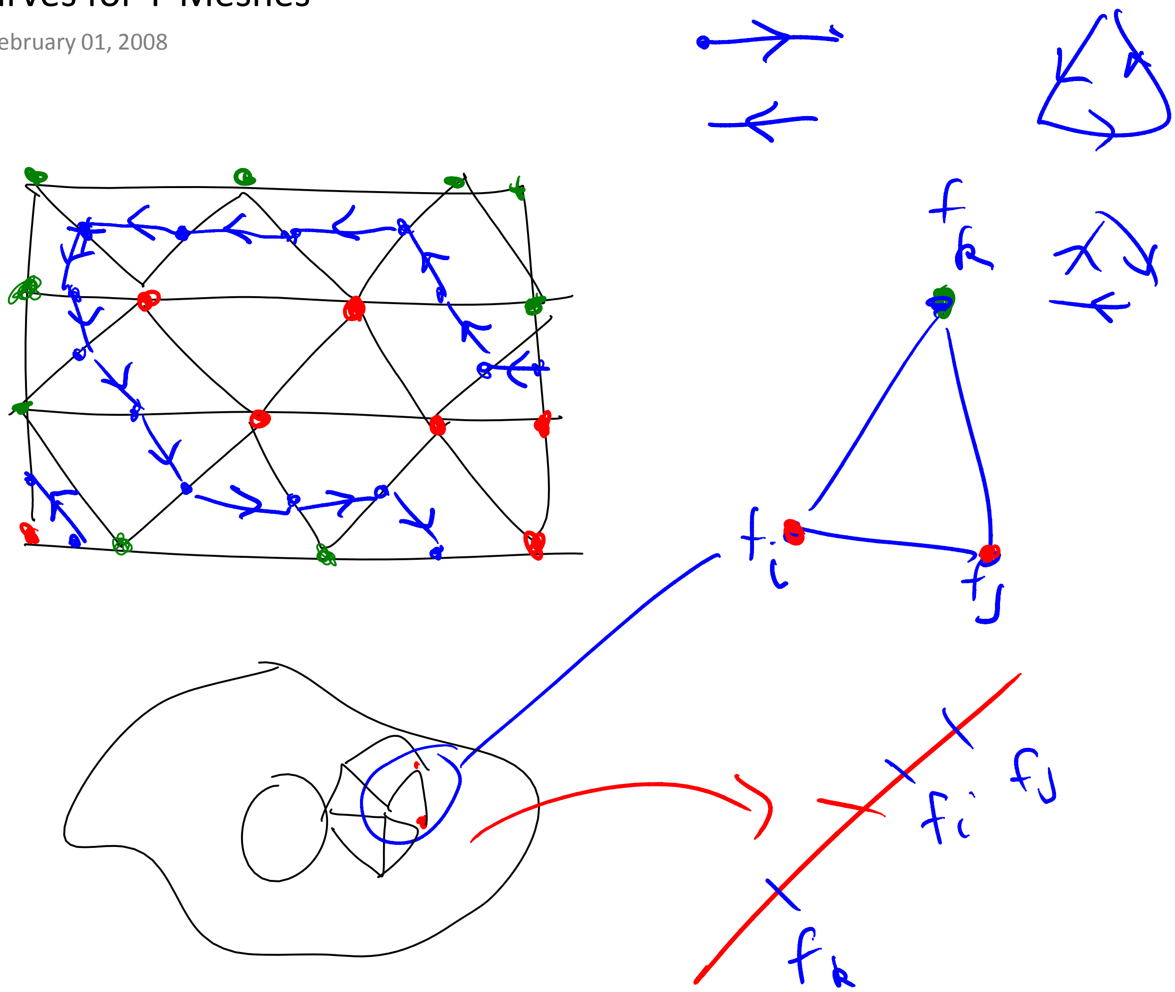
$$(1-\lambda) f_i = -\lambda f_j$$

$$f_i = \lambda (f_i - f_j)$$

$$\lambda = \frac{f_i}{f_i - f_j} \qquad 1 - \lambda = \frac{f_j}{f_j - f_i}$$

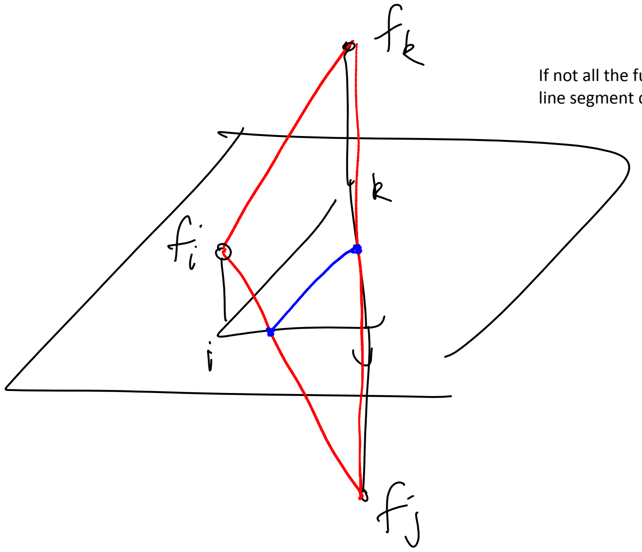
IsoCurves for T-Meshes

Friday, February 01, 2008
2:17 PM



IsoCurve within one triangle

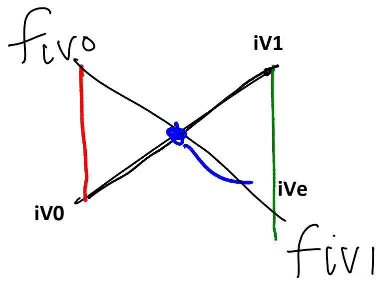
Friday, February 01, 2008
2:39 PM

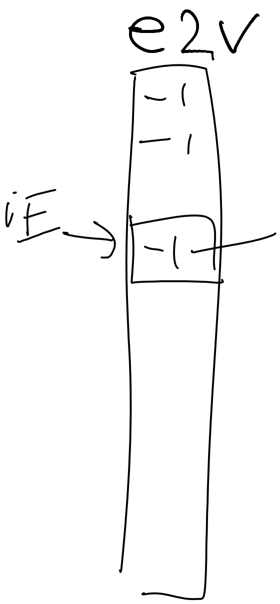


If not all the function values at the vertices have the same sign, then the isocurve within the triangle is a line segment connecting two points on the triangle edges

IsoCurve Vertices

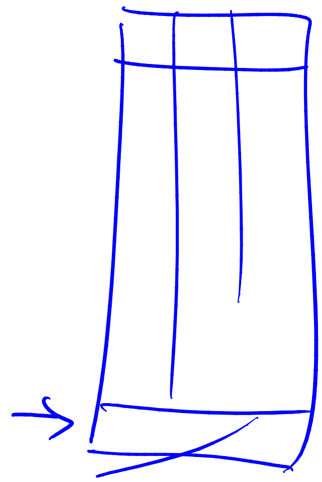
Friday, February 01, 2008
2:43 PM





iVe

iVe

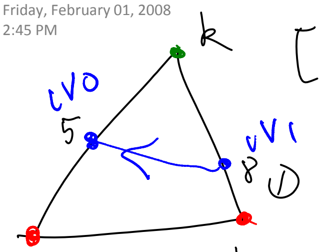


The first step of the algorithm is to create the isocurve vertices and to store pointers to those vertices on the corresponding supporting vertices

1) Create an array **e2v** of length **nE** to store the indices of the newly created vertices
2) Initialize the array to all -1's which indicates that no vertex is associated with any edge
3) For each edge **e=(iV0,iV1)** of the mesh
    a. if isocurve cuts the edge
    b. Allocate space in **coord** array to store new vertex coordinates and store new vertex index **iVe** in **e2v**
    c. Compute new vertex coordinates by linear interpolation of vertex function values

IsoCurve edges

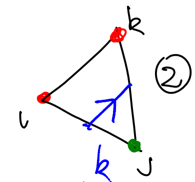
Friday, February 01, 2008
2:45 PM



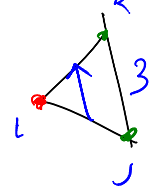
$[i, j, k, -1] \longrightarrow [iv_1, iv_0, -1]$

$2^3 = 8$

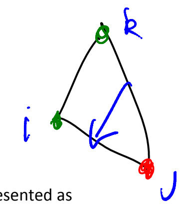
	i f_i	f_j	f_k	V_{ij} (0)	V_{jk} (1)	V_{ki} (2)
0	+	+	+	X	X	X
1	+	+	−	X		
2	+	−	+			X
3	+	−	−		X	
4	−	+	+			
5	−	+	−			X
6	−	−	+	X		
7	−	−	−	X	X	X



The second step is to determine how the isocurve vertices computed in step 1 should be connected forming isocurve edges



Look-up table approach: precompute connectivity



0	[ ]
1	[1, 2]
2	[0, 1]
3	[0, 2]
4	[2, 0]
5	[1, 0]
6	[2, 1]
7	[ ]

Table contains connectivity of isocurve represented as a list of half edge indices (within the triangle)

$\sigma_i \quad \sigma_j \quad \sigma_k \in \{0, 1\}$

Index into the table computed from signature of function at corners

$$indx = 4\sigma_i + 2\sigma_j + \sigma_k$$

$$\begin{array}{c} 0 \\ 1 \\ 2 \end{array}\begin{bmatrix} -1 \\ 5 \\ 8 \end{bmatrix} \Longrightarrow [5, 8, -1]$$

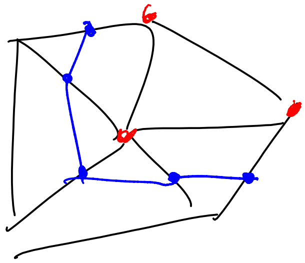
Half edge indices are replaced by corresponding isocurve vertex indices computed in step 1

The output curve can be represented as an **IndexedLineSet**

IsoCurve Surgery

Friday, February 01, 2008
3:01 PM

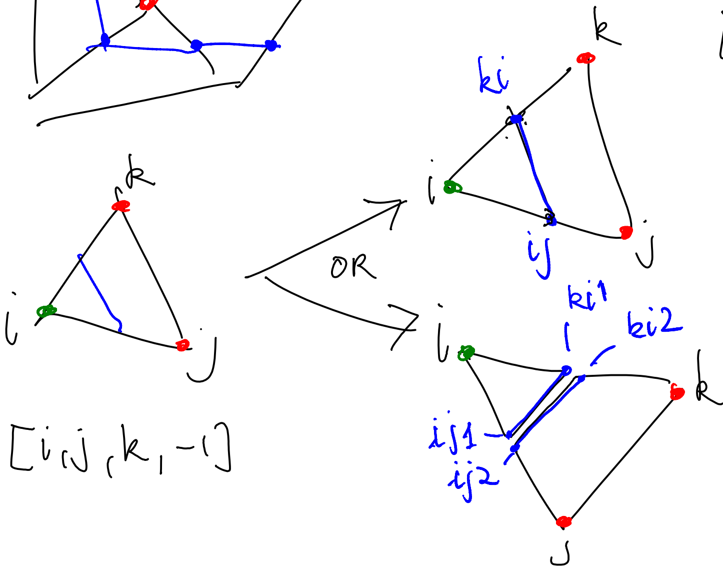
An alternative is to use an implicit function defined on the vertices of a triangle mesh to cut the faces along the isocurve



If **nV** is the number of vertices of the mesh before the operation, the isocurve vertices should be given vertex indices **nV,nV+1,nV+2,...**

The mesh can be kept connected

[i, ij, ki, -1
j, k, ki, ij, -1]



Or with the same effort can be cut through the isocurve edges

[i, ij1, ki1, -1
j, k, ki2, ij2, -1]