Programming Algorithm

Anzong Zheng

January 24th, 2016

Contents

Chapter 1

Math Definitions

1.1 Vector Norm

ℓ^2-norm (also written "l^2"-norm) $|\vec{x}|$ is a **vector norm** defined for a **complex vector**

$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \tag{1.1.1}$$

by

$$|\vec{x}| = \sqrt{\sum_{k=1}^{n} |x_k|^2} \tag{1.1.2}$$

where $|x_k|$ on the right denotes the **complex modulus**. The ℓ^2-norm is the **vector norm** that is commonly encountered in vector algebra and vector operations (such as the **dot product**), where it is commonly denoted $|\vec{x}|$. However, if desired, a more explicit (but more cumbersome) notation $|\vec{x}|_2$ can be used to emphasize the distinction between the **vector norm** $|\vec{x}|$ and **complex modulus** $|z|$ together with the fact that the ℓ^2-norm is just one of several possible of norms.

For **real vectors**, the absolute value sign indicating that a complex modulus is being taken on the right of equation 1.1.2 may be dropped. So, for example, the ℓ^2-norm of the vector $\vec{x} = (x_1, x_2, x_3)$ is given by

$$|\vec{x}| = \sqrt{(x_1^2 + x_2^2 + x_3^2)} \tag{1.1.3}$$

The ℓ^2-norm is also known as Euclidean norm. However, this is terminology is not recommended since it may cause confusion with the **Frobenius norm**(a **matrix norm**) is also sometimes called the Euclidean norm.

The "L^2-norm" (denoted with an uppercase L) is reserved for application with a function $\phi(x)$,

$$|\phi|^2 \equiv \phi \cdot \phi \equiv < \phi|\phi > \equiv \int |\phi(x)|^2 dx \tag{1.1.4}$$

1.2 Matrix Norm

Given a **square complex** or **real matrix A**, a matrix norm $\|\mathbf{A}\|$ is a **nonnegative** number associated with **A**.

- the maximum absolute column sum norm of the matrix

$$\|\mathbf{A}\|_1 = \max_{1 \le j \le n} \sum_{i=1}^{m} |a_{ij}| \tag{1.2.1}$$

- the maximum absolute row sum norm of the matrix

$$\|\mathbf{A}\|_\infty = \max_{1 \le i \le m} \sum_{j=1}^{n} |a_{ij}| \tag{1.2.2}$$

- the **spectral** norm, which is the square root of the maximum eigenvalue of $\mathbf{A}^H\mathbf{A}$, is often referred as "the" matrix norm

$$\|\mathbf{A}\|_2 = \sqrt{\text{maximum eigenvalue of } \mathbf{A}^H\mathbf{A}} \tag{1.2.3}$$

- Frobenius norm or $L_{2,2}$ norm. The equality holds if and only if the matrix $\mathbf{A}$ is a rank-one matrix or a zero matrix.

$$\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2} \tag{1.2.4}$$

1.2.1 Frobenius Norm

The Frobenius norm, sometimes also called the Euclidean norm (a term unfortunately also used for the vector L^2-norm), is **matrix norm** of an $m \times n$ matrix $\mathbf{A}$ defined as the **square root** of the sum of the absolute squares of its elements,

$$\|\mathbf{A}\|_F \equiv \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2} \tag{1.2.5}$$

The Frobenius norm can also be considered as a **vector norm**.

It is also equal to the **square root** of the **matrix trace** of $\mathbf{AA}^H$, where $\mathbf{A}^H$ is the conjugate transpose, i.e.,

$$\|\mathbf{A}\|_F = \sqrt{\text{Tr}(\mathbf{AA}^H)} \tag{1.2.6}$$

1.3 Determinant

In analytical geometry, determinants express the **signed volumes of n-dimensional parallelepipeds** .

It can be proven that any matrix has a unique inverse if its determinant is nonzero.

1.3.1 2×2 **matrices**

The determinant of a 2×2 matrix is defined by:

$$\det(\mathbf{A}) = \begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc \tag{1.3.1}$$

Supposed a parallelogram defined by two vectors $\vec{v}_1 = (a, c)$ and $\vec{v}_2 = (b, d)$, so the parallelogram can be expressed with vertices at $(0, 0), (a, b), (a + c, b + d),$ and (c, d) as shown in
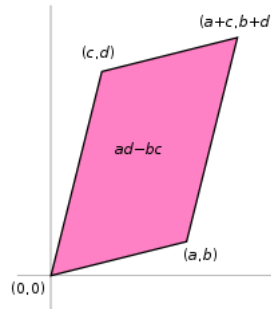

Figure 1.1: Area parallelogram

so:

$$\text{Signed Area} = |\vec{v}_1||\vec{v}_2|\sin\theta = |\vec{v}_1^{\perp}||\vec{v}_2|\cos\theta' = \begin{bmatrix} -b \\ a \end{bmatrix} \cdot \begin{bmatrix} c \\ d \end{bmatrix} = ad - bc \tag{1.3.2}$$

Thus the determinant gives the scaling factor and the orientation induced by the mapping represented by $\mathbf{A}$. When the determinant is equal to one, the linear mapping defined by the matrix **equil-area** and **orientation-preserving**.

1.3.2 3×3 **matrices**

The determinant of a 3×3 matrix (signed volume) is defined by:

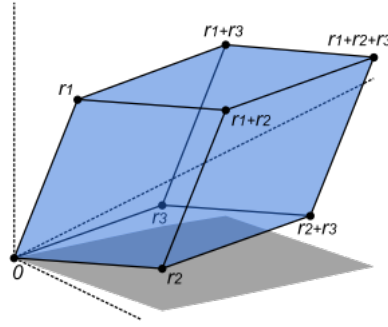

Figure 1.2

$$\det(\mathbf{A}) = \begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} = a \begin{vmatrix} e & f \\ h & i \end{vmatrix} - b \begin{vmatrix} d & f \\ g & i \end{vmatrix} + c \begin{vmatrix} d & e \\ g & h \end{vmatrix}$$
$$= a(ei - fh) - b(di - fg) + c(dh - eg)$$
$$= aei + bfg + cdh - ceg - bdi - afh$$

(1.3.3)

1.4 Operators

1.4.1 Kronecker product

In mathematics, the **Kronecker product**, denoted by $\otimes$, is an operation on two matrices of arbitrary size resulting in a block matrix. It is generalization of the outer product (which is denoted by the same symbol) from vectors to matrices, and gives the matrix of the tensor product with respect to a standard choice of basis. The Kronecker product should not be confused with the usual matrix multiplication, which is an entirely different operation.

If $\mathbf{A}$ is an $m \times n$ matrix and $\mathbf{B}$ is a $p \times q$ matrix, then the Kronecker product $\mathbf{A} \otimes \mathbf{B}$ is the $mp \times nq$ block matrix:

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & \cdots & a_{mn}\mathbf{B} \end{bmatrix}$$

(1.4.1)

more explicitly:

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}b_{11} & a_{11}b_{12} & \cdots & a_{11}b_{1q} & \cdots & \cdots & a_{1n}b_{11} & a_{1n}b_{12} & \cdots & a_{1n}b_{1q} \\ a_{11}b_{21} & a_{11}b_{22} & \cdots & a_{11}b_{2q} & \cdots & \cdots & a_{1n}b_{21} & a_{1n}b_{22} & \cdots & a_{1n}b_{2q} \\ \vdots & \vdots & \ddots & \vdots & & & \vdots & \vdots & \ddots & \vdots \\ a_{11}b_{p1} & a_{11}b_{p2} & \cdots & a_{11}b_{pq} & \cdots & \cdots & a_{1n}b_{p1} & a_{1n}b_{p2} & \cdots & a_{1n}b_{pq} \\ \vdots & \vdots & & \vdots & \ddots & & \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & & \vdots & & \ddots & \vdots & \vdots & \ddots & \vdots \\ a_{m1}b_{11} & a_{m1}b_{12} & \cdots & a_{m1}b_{1q} & \cdots & \cdots & a_{mn}b_{11} & a_{mn}b_{12} & \cdots & a_{mn}b_{1q} \\ a_{m1}b_{21} & a_{m1}b_{22} & \cdots & a_{m1}b_{2q} & \cdots & \cdots & a_{mn}b_{21} & a_{mn}b_{22} & \cdots & a_{mn}b_{2q} \\ \vdots & \vdots & \ddots & \vdots & & & \vdots & \vdots & \ddots & \vdots \\ a_{m1}b_{p1} & a_{m1}b_{p2} & \cdots & a_{m1}b_{pq} & \cdots & \cdots & a_{mn}b_{p1} & a_{mn}b_{p2} & \cdots & a_{mn}b_{pq} \end{bmatrix}$$

(1.4.2)

If $\mathbf{A}$ and $\mathbf{B}$ represent linear transformations $\mathbf{v}_1 \to \mathbf{w}_1$ and $\mathbf{v}_2 \to \mathbf{w}_2$, respectively, then $\mathbf{A} \otimes \mathbf{B}$ represents the tensor product of the two maps, $\mathbf{v}_1 \otimes \mathbf{v}_2 \to \mathbf{w}_1 \otimes \mathbf{w}_2$

Example

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \otimes \begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix} = \begin{bmatrix} 1 \cdot 0 & 1 \cdot 5 & 2 \cdot 0 & 2 \cdot 5 \\ 1 \cdot 6 & 1 \cdot 7 & 2 \cdot 6 & 2 \cdot 7 \\ 3 \cdot 0 & 3 \cdot 5 & 4 \cdot 0 & 4 \cdot 5 \\ 3 \cdot 6 & 3 \cdot 7 & 4 \cdot 6 & 4 \cdot 7 \end{bmatrix} = \begin{bmatrix} 0 & 5 & 0 & 10 \\ 6 & 7 & 12 & 14 \\ 0 & 15 & 0 & 20 \\ 18 & 21 & 24 & 28 \end{bmatrix} \tag{1.4.3}$$

1.5   Spline section

1.5.1   B-spline

In the mathematics subfield of numerical analysis, a **B-spline**, or **basis spline**, is a spline function that has minimal support with respect to a given degree, smoothness, and domain partition. Any spline function of given degree can be expressed as a linear combination of B-spline of that degree.

A spline function is a piecewise polynomial function of degree $< k$ in a variable x. The places where the pieces meet are known as knots. The number of internal knots must be equal to, or greater than $k - 1$. The key property of spline functions is that they are continuous at the knots. Some derivatives of the spline function may also be continuous, depending on whether the knots are distinct or not.

Definition

A B-spline is a piecewise polynomial function of degree $< n$ in a variable x. It is defined over a domain $t_0 \le x \le t_m$, $m = n$. The points where $x = t_j$ are known as knots or break-points. The number of internal knots is equal to the degree of the polynomial if there are no knot multiplicities. The knots must be in ascending order. The number of knots is the minimum for the degree of the B-spline, which has a non-zero vale only in the range between the first and last knot. Each piece of the function is a polynomial of degree $< n$ between and including adjacent knots. A B-spline is a continuous function at the knots. When all internal knots are distinct its derivatives are also continuous up to the derivative of degree $n - 1$. If internal knots are coincident at a given value of x, the continuity of derivative order is reduced by 1 for each additional knot.

For any given set of knots, the B-spline is unique, hence the name, B being short for Basis. The usefulness of B-splines lies in the fact that any spline function of order n on a given set of knots can be expressed as a linear combination of B-splines.

$$S_{n,t}(x) = \sum_i \alpha_i B_{i,n}(x) \tag{1.5.1}$$

This follows from the fact that all pieces have the same continuity properties, within their individual range of support, at the knots.

Expressions for the polynomial pieces can be derived by means of a recursion formula

$$B_{i,1}(x) := \begin{cases} 1 & \text{if } t_i \le x < t_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{1.5.2}$$

$$B_{i,k}(x) := \frac{x - t_i}{t_{i+k-1} - t_i} B_{i,k-1}(x) + \frac{t_{i+k} - x}{t_{i+k} - t_{i+1} B_{i+1,k-1}}(x) \tag{1.5.3}$$

That is, $B_{j,1}(x)$ is piecewise constant one or zero indicating which knot span x is in (zero if knot span j is repeated). The recursion equation is in two parts:

$$\frac{x - t_i}{t_{i+k-1} - t_i} \tag{1.5.4}$$

ramps from zero to one as x goes from t_i to t_{i+k-1} and

$$\frac{t_{i+k} - x}{t_{i+k} - t_{i+1}} \tag{1.5.5}$$

1.5.2   Quartic & biquartic function

In mathematics, a **quartic function**, is a function of the form

$$f(x) = ax^4 + bx^3 + cx^2 + dx + e \tag{1.5.6}$$

where a is nonzero, which is defined by a polynomial of degree four, called **quartic polynomial**.

Sometimes the term **biquartic** is used instead of *quartic*, but, **biquartic function** refers to a **quadratic function** of a aquare (or, equivalently, to the function defined by a quartic polynomial without terms of odd degree), having the form

$$f(x) = ax^4 + cx^2 + e \tag{1.5.7}$$

A **quartic equation**, or equation of the fourth degree, is an equation that equates a quartic polynomial to zero, of the form

$$ax^4 + bx^3 + cx^2 + dx + e = 0 \tag{1.5.8}$$

where $a \neq 0$

1.5.3   Cubic & bicubic function

In mathematics, a **cubic function** is a function of the form

$$f(x) = ax^3 + bx^2 + cx + d \tag{1.5.9}$$

where a is nonzero. In other words, a cubic function is defined by a polynomial of degree three.
Setting $f(x) = 0$ produces a **cubic equation** of the form:

$$ax^3 + bx^2 + cx + d = 0 \tag{1.5.10}$$

Cubic splines can be extended to functions of two or more parameters, in several ways. Bicubic splines are often used to interpolate data on a regular grid, such as **pixel** values in a digital image. **Bicubic surface patches**, defined by three bicubic splines, are an essential tool in computer graphics.

1.6   Rotation In two dimensions

In two dimensions, every rotation matrix has the following form,

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{1.6.1}$$

1.6.1   Common rotations

Particularly useful are the matrices for 90°, 180°, and 270° counterclockwise rotation:

$$R(90°) = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

$$R(180°) = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$R(270°) = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

1.7   Rotation In three dimensions

A basic rotation (also called elemental rotation) is a rotation about one of the axes of a coordinate system. The following three basic rotation matrices rotate vectors by an angle θ about the x, y or z axis, in three dimensions, using the **right hand rule**–which codifies their alternating signs.

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$$

$$R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

$$R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

For **column vectors**, each of these basic vector rotations appears counter-clockwise when the axis about which they occur points toward the observer, the coordinate system is right-handed, and the angle θ is positive. R_z, for instance, would rotate toward the y-axis a vector aligned with the x-axis, as can easily be checked by operating with R_z on vector $(1, 0, 0)$:

$$R_z(90°) \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \cos 90° & -\sin 90° & 0 \\ \sin 90° & \cos 90° & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \tag{1.7.1}$$

Chapter 2

Basic geometries

2.1 Triangle Surface Area

If your 3 points are $\vec{A}, \vec{B}, \vec{C}$ then you may use directly (half) cross-product formula:

$$\vec{u} = \vec{AB} \tag{2.1.1}$$

$$\vec{v} = \vec{AC} \tag{2.1.2}$$

$$S = \frac{|\vec{u} \times \vec{v}|}{2} = \frac{|\vec{u}||\vec{v}||sin(\theta)|}{2} \tag{2.1.3}$$

that is (see the Wikipedia link to get the cross-product in $\mathbb{R}^3$):

$$S = \frac{1}{2} \sqrt{(v_1 w_2 - w_1 v_2)^2 + (w_1 u_2 - u_1 w_2)^2 + (u_1 v_2 - v_1 u_2)^2}$$

2.2 Superellipse
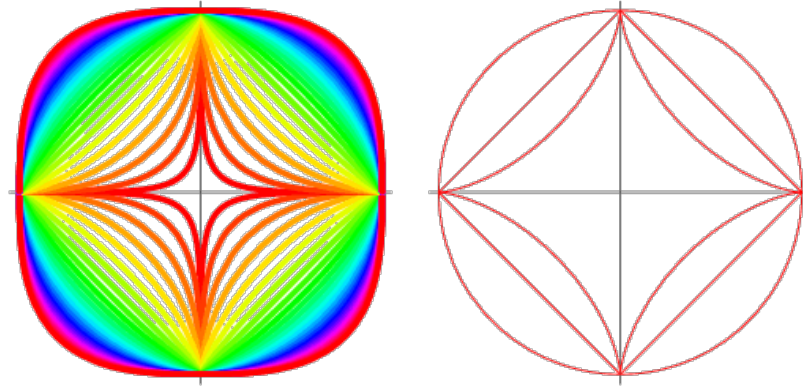
Figure 2.1

A superellipse is a curve with Cartesian equation

$$\left(\frac{x}{a}\right)^r + \left(\frac{y}{b}\right)^r = 1, \tag{2.2.1}$$

first discussed in 1818 by Lamé. A superellipse may be described parametrically by

$$x = a \cos^{\frac{2}{r}} t$$

$$y = b \sin^{\frac{2}{r}} t$$

The restriction to $r > 2$ is sometimes made.

Superellipses with $a = b$ are also know Lamè curve or Lamè ovals, and the case $a = b$ with $r = 4$ is sometimes known as the squircle. By analogy, the superellipse with $a! = b$ and $r = 4$ might be termed the rectellipse.

A range of superellipses are shown above, with special cases $r = 2/3$, 1, and 2 illustrated right above. The following table summarizes a few special cases. Piet Hein used r=5/2 with a number of different a/b ratios for various of his projects. For example, he used $a/b = 6/5$ for Sergels Torg (Sergel's Square) in Stockholm, Sweden (Vestergaard), and $a/b = 3/2$ for his table.

r	curve
$\frac{2}{3}$	(squashed) astroid
1	(squashed) diamond
2	ellipse
$\frac{5}{2}$	Piet Hein's "superellipse"
4	rectellipse

If r is a rational, then a superellipse is algebraic. However, for irrational r, it is transcendental. For even integers $r = n$, the curve becomes closer to a rectangle as n increases.

The area of a superellipse is given by

$$\frac{4^{1-\frac{1}{r}} ab \sqrt{\pi} \text{Gamma}\left[1 + \frac{1}{r}\right]}{\text{Gamma}\left[\frac{1}{2} + \frac{1}{r}\right]} \tag{2.2.2}$$
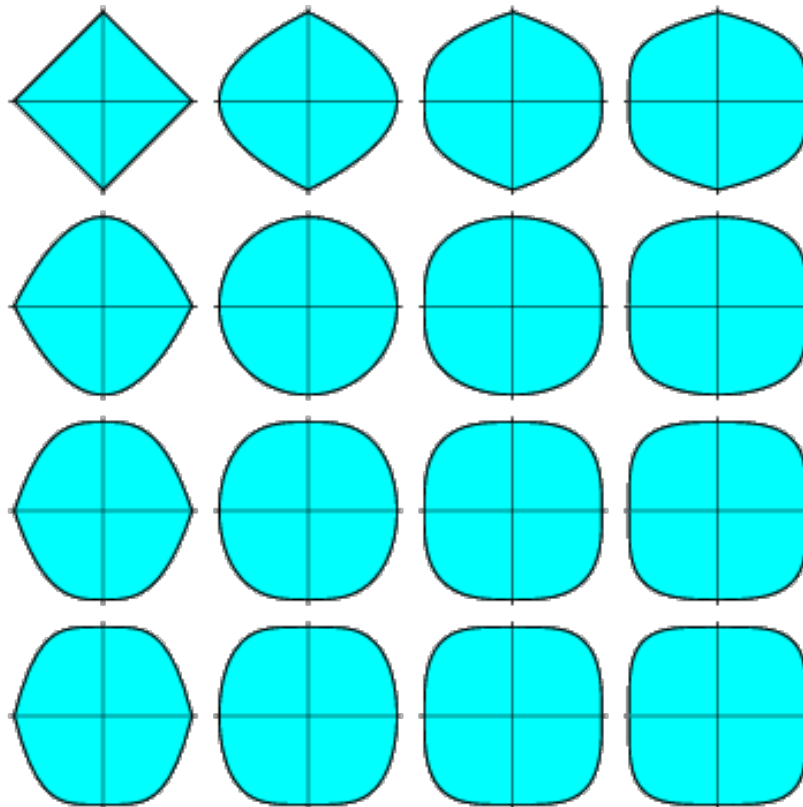

Figure 2.2

2.3 Frustum volume

If the shape parameters at left side (square) of ribs is $b1, h1$, at right side (square) of rib is $b2, h2$, then the volume of the frustum is:

$$\begin{aligned} \text{vol} &= \frac{S_1 + S_2 + \sqrt{(S_1 S_2)}}{3} l \\ &= \frac{b_1 h_1 + b_2 h_2 + \sqrt{(b_1 b_2 h_1 h_2)}}{3} l \end{aligned} \tag{2.3.1}$$

the gradient of the volume is:

$$\begin{aligned} \frac{\partial \text{vol}}{\partial b_1} &= \frac{h_1}{6} l + \frac{1}{2} \sqrt{\frac{h_1 b_2 h_2}{b_1}} l \\ \frac{\partial \text{vol}}{\partial h_1} &= \frac{b_1}{6} l + \frac{1}{2} \sqrt{\frac{b_1 b_2 h_2}{h_1}} l \\ \frac{\partial \text{vol}}{\partial b_2} &= \frac{h_2}{6} l + \frac{1}{2} \sqrt{\frac{b_1 h_1 h_2}{b_2}} l \\ \frac{\partial \text{vol}}{\partial h_2} &= \frac{b_2}{6} l + \frac{1}{2} \sqrt{\frac{b_1 b_2 h_1}{h_2}} l \end{aligned} \tag{2.3.2}$$

2.4 Combined H beam volume

let shape parameters at left side (square) of ribs is $b_{1i}, h_{1i}, b_{2i}, h_{2i}$, at right side (square) of rib is $b_{1j}, h_{1j}, b_{2j}, h_{2j}$, then the volume of the frustum is:

$$\begin{aligned} \text{vol} &= \frac{S_1 + S_2 + \sqrt{(S_1 S_2)}}{3} l \\ &= \frac{b_{1i} h_{1i} + b_{1j} h_{1j} + \sqrt{(b_{1i} b_{1j} h_{1i} h_{1j})}}{3} l + \frac{b_{2i}(h_{2i} - h_{1i}) + b_{2j}(h_{2j} - h_{1j}) + \sqrt{(b_{2i} b_{2j}(h_{2i} - h_{1i})(h_{2j} - h_{1j}))}}{3} l \end{aligned} \tag{2.4.1}$$

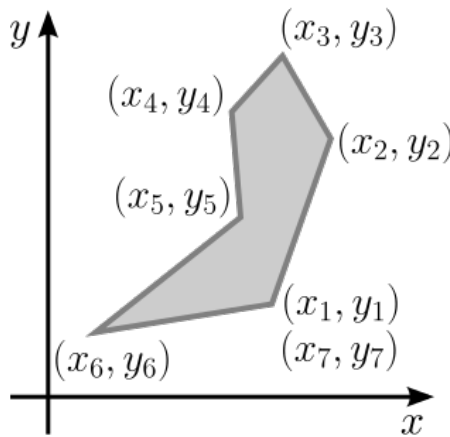## 2.5 Area and second moments of any polygon

Figure 2.3: Numbering of a polygon

The second moment of area for any simple polygon on the XY-plane can be computed in general by summing contributions from each segment of the polygon. A polygon is assumed to have n vertices, numbered in counterclockwise fashion. If polygon vertices are numbered clockwise, returned values will be negative, but absolute values will be correct.

$$A = \frac{1}{2}\sum_{i=1}^{n}(x_i y_{i+1} - x_{i+1} y_i)$$

$$I_x = \frac{1}{12}\sum_{i=1}^{n}(y_i^2 + y_i y_{i+1} + y_{i+1}^2)(x_i y_{i+1} - x_{i+1} y_i)$$

$$I_y = \frac{1}{12}\sum_{i=1}^{n}(x_i^2 + x_i x_{i+1} + x_{i+1}^2)(x_i y_{i+1} - x_{i+1} y_i)$$

where x_i, y_i are the coordinates of the i-th polygon vertex, for $1 \le i \le n$. Also, x_{n+1}, y_{n+1} are assumed to be equal to the coordinates of the first vertex, i.e., $x_{n+1} = x_1$ and $y_{n+1} = y_1$.
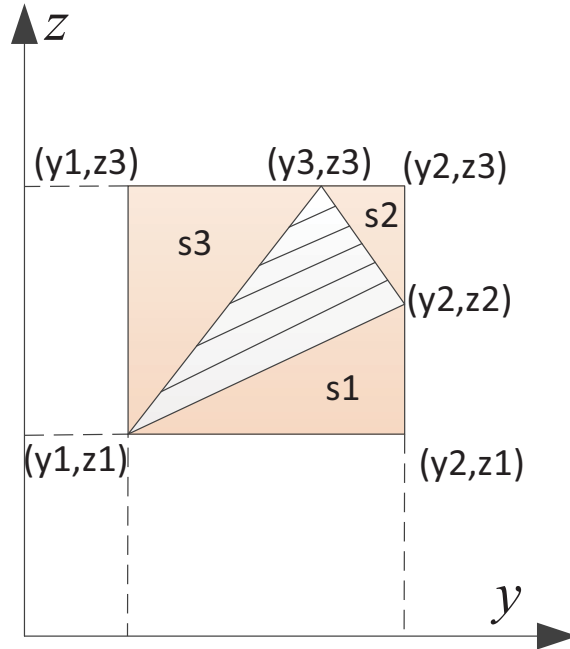
2.5.1   equations for a triangle

Figure 2.4

$$\delta_{y_1} = (y_2 - y_1) - \frac{y_2 - y_1}{z_2 - z_1}(z - z_1) \qquad \delta_{y_2} = \frac{y_2 - y_3}{z_3 - z_2}(z - z_2) \qquad \delta_{y_3} = \frac{y_3 - y_1}{z_3 - z_1}(z - z_1)$$

$$\delta_{z_1} = \frac{z_2 - z_1}{y_2 - y_1}(y - y_1) \qquad \delta_{z_2} = \frac{z_3 - z_2}{y_2 - y_3}(y - y_3) \qquad \delta_{z_3} = (z_3 - z_1) - \frac{z_3 - z_1}{y_3 - y_1}(y - y_1)$$

$$A_1 = \int_{z_1}^{z_2}\delta_{y_1}dz \qquad A_2 = \int_{z_2}^{z_3}\delta_{y_2}dz \qquad A_3 = \int_{z_1}^{z_3}\delta_{y_3}dz$$

$$I_{y_1} = \int_{z_1}^{z_2}\delta_{y_1}z^2dz \qquad I_{y_2} = \int_{z_2}^{z_3}\delta_{y_2}z^2dz \qquad I_{y_3} = \int_{z_1}^{z_3}\delta_{y_3}z^2dz$$

$$I_{z_1} = \int_{y_1}^{y_2}\delta_{z_1}y^2dz \qquad I_{z_2} = \int_{y_3}^{y_2}\delta_{z_2}y^2dz \qquad I_{z_3} = \int_{y_1}^{y_3}\delta_{z_3}y^2dz$$

$$A = (y_2 - y_1)(z_3 - z_1) - A_1 - A_2 - A_3$$

$$I_y = \int_{z_1}^{z_3} (y_2 - y_1)z^2 dz - I_{y_1} - I_{y_2} - I_{y_3}$$

$$I_z = \int_{y_1}^{y_3} (z_3 - z_1)y^2 dz - I_{z_1} - I_{z_2} - I_{z_3}$$

The results obtained by this exact equations is euqal to the results from above section. see example in Run_anyPolygon.m

Chapter 3

Mesh Processing

3.1 Recsconstruct quad rib mesh

See `Run_reconstruct_quadRibMesh.cpp` and `generalRibs3.cpp`

3.2 slice quad rib mesh

See `Run_slice_quadRibMesh.cpp` and `generalRibs3.cpp`

3.3 Laplacian Smoothing

For each vertex in a mesh, a new position is chosed based on local information (such as the position of neighbors) and the veertex is moved there. In the case that a mesh is topologically a rectangular grid (that is, each internal vertex is connected to four neighbrs) then this operation produces the Laplacian of the mesh.

The smoothing operation may be described per-vertex as:

$$\bar{x}_i = \frac{1}{N} \sum_{j=1}^{N} \bar{x}_j \tag{3.3.1}$$

Where N is the number of adjacent vertices to node i, $\bar{x}_j$ is the position of the j-thadjacent vertex and $\bar{x}_i$ is the new position for node i

3.4 Surface Parameterization

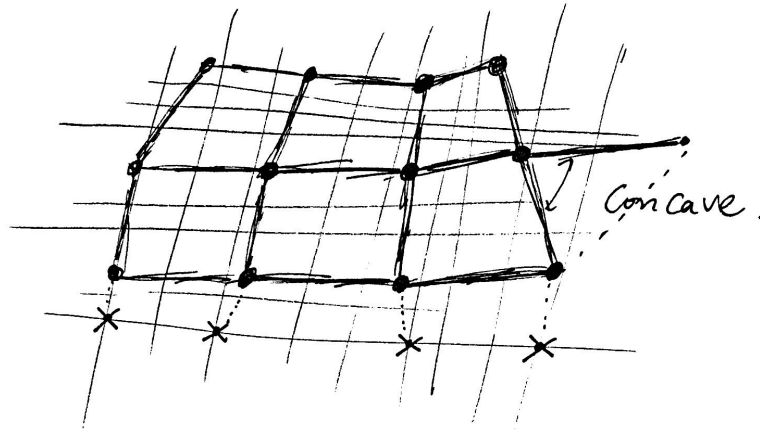A surface in 3-space can be parameterized by two variabls (or coordinates) u and v such that:

$$x = x(u, v) \tag{3.4.1}$$
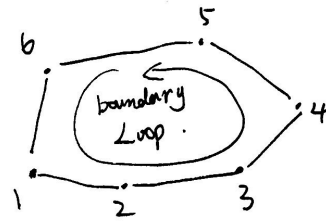$$y = y(u, v) \tag{3.4.2}$$
$$z = z(u, v) \tag{3.4.3}$$

If a surface is parameterized as above, then the tangent vectors

$$\mathbf{T}_u = \frac{\partial x}{\partial u}\hat{\mathbf{x}} + \frac{\partial y}{\partial u}\hat{\mathbf{y}} + \frac{\partial z}{\partial u}\hat{\mathbf{z}}$$
$$\mathbf{T}_v = \frac{\partial x}{\partial v}\hat{\mathbf{x}} + \frac{\partial y}{\partial v}\hat{\mathbf{y}} + \frac{\partial z}{\partial v}\hat{\mathbf{z}}$$
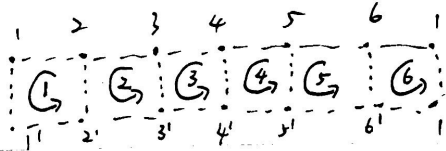
Figure 3.1: reconstruct quad rib mesh

3.5   Trace edges

3.6   Polyline to polygon

Given a set of points arranged in anti-clockwise direction with thickness at each point, a polygon mesh is built based on these points.

The vertices enclosing the polygon are actually intersection vertices of two lines:

As can be seen in figure 3.3, p_a and p_b are two points above polylines with offse $\delta_{i,1}$ and $\delta_{i,2}$ separately. Two red lines a and b through p_a and p_b are built respectively.

The coordinates of p_a and p_b are:

$$p_a = p_i + \frac{\delta_{i,1}}{2}\vec{e}_a$$

$$p_b = p_i + \frac{\delta_{i,2}}{2}\vec{e}_b$$

where $\vec{e}_a = \mathrm{cross}([1;0;0], p_i - p_{i-1})$, $\vec{e}_b = \mathrm{cross}([1;0;0], p_{i+1} - p_i)$
and the equations for red lines are:

$$\frac{z - z_a}{y - y_a} = \frac{z_i - z_{i-1}}{y_i - y_{i-1}}$$

$$\frac{z - z_b}{y - y_b} = \frac{z_{i+1} - z_i}{y_{i+1} - y_i}$$

The intersected point of two lines can be found by soling symbolic equations:

```matlab
function SolveIntersectionPoint()
% syms y1 z1 y2 z2 y3 z3
syms t1 t2 t3 t4
syms ya za yb zb
syms y z

%   z-za        z2-z1       t1
% ------ = ------- = -----
%   y-ya        y2-y1       t2

%   z-zb        z3-z2       t3
% ------ = ------- = -----
%   y-yb        y3-y2       t4

% [y, z] = solve( (z-za)/(y-ya) == (z2-z1)/(y2-y1), (z-zb)/(y-yb) == (z3-z2)/(y3-↩
    y2), y, z );
[y, z] = solve( (z-za)/(y-ya) == t1/t2, (z-zb)/(y-yb) == t3/t4, y, z )

% results.
% y = (t1*t4*ya - t2*t3*yb - t2*t4*za + t2*t4*zb)/(t1*t4 - t2*t3)
% z = (t1*t3*ya - t1*t3*yb - t2*t3*za + t1*t4*zb)/(t1*t4 - t2*t3)

end % end of function.
```

After having all vertices for the polygon, the mesh is obtained in figure 3.4:
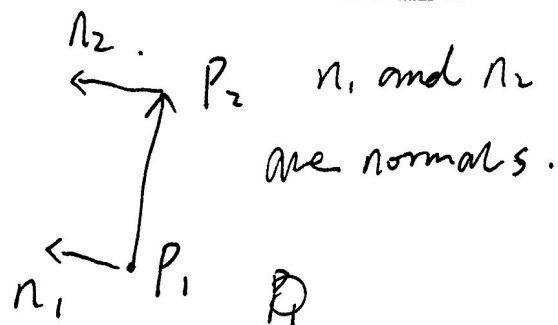Note the red lines are the input polyline.

t_1

t_2

P_2 direction:

$P_1 \longrightarrow P_2$

t_2'
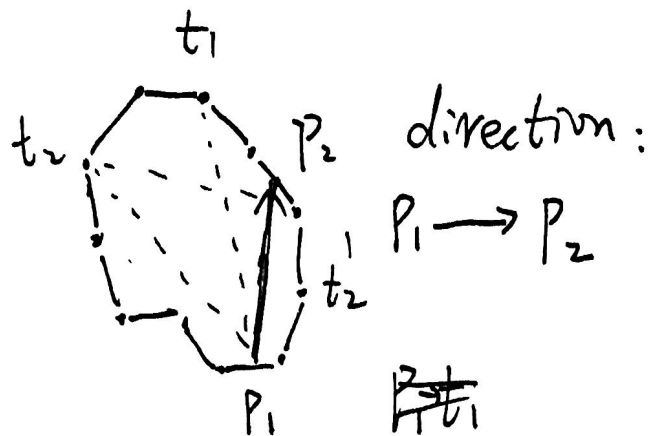
P_1 $\boxed{P_1 \atop t_1}$

$P_1 \to t_1$ and $P_1 \to P_2$

① $dot(P_1 \to t_1, P_1 \to P_2) > 0$

   outside . $P_1 \to P_2$

② $dot(P_1 \to t_1, P_1 \to t_2) < 0$

   between $P_1 \to P_2$

③ t_2 and t_2' are

   inside .

n_2 .

P_2 n_1 and n_2

   are normals .

n_1 P_1 P_2

$Cross(P_1 \to t_2', P_1 \to P_2)$

$Cross(P_1 \to t_2, P_1 \to P_2)$

$facenrm = Cross(P_1 \to \emptyset, n_1)$

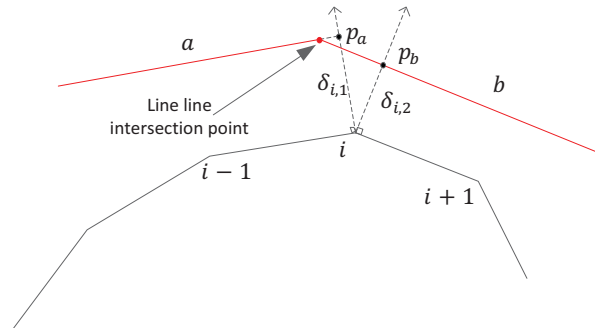Figure 3.2: slice quad rib mesh

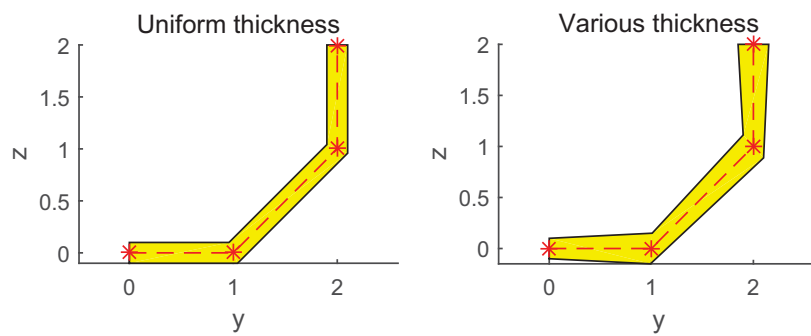Figure 3.3: Intersection points

Figure 3.4: polygon formed from polyline

Chapter 4

Optimization

4.1 fmincon

4.1.1 options

```
options = optimset('GradObj', 'on', 'Algorithm', 'interior-point', ...
'Display', 'iter', 'MaxIter', 500000, 'MaxFunEvals',10000000,  'Diagnostics', 'on↩
    ',...
'TolFun', 1.0e-6, 'TolX', 1.0e-6, 'TolCon', 1.0e-6);
```

4.1.2 test volume function

Set all constraints except lower bounds and upper bounds to [], and run **fmincon** to test whether all variables converge to lower bounds.

```
[x,fval,exitflag,output] = fmincon(@(x_br)Rib_Vol_m(x_br), x0, [], [], [], [], lx↩
    , ux, [], options);
```