I study the linear least squares problem here to understand `do_lfit()` of the mesh optimization.

## Least-Squares Problems

- The Least-Squares problem is to find $\mathbf{x}$ to make $A\mathbf{x}$ as close as possible to $\mathbf{b}$ in $A\mathbf{x} = \mathbf{b}$ problem.
- We can convert the above problem into finding $\mathbf{x}$ such that $||\mathbf{b} - A\mathbf{x}||$ is as smallest as possible.
- If $A \in \mathbb{R}^{m \times n}$, $\mathbf{b} \in \mathbb{R}^m$, a least-squares solution of $A\mathbf{x} = \mathbf{b}$ is an $\hat{\mathbf{x}} \in \mathbb{R}^n$ such that $||\mathbf{b} - A\hat{\mathbf{x}}|| \leq ||\mathbf{b} - A\mathbf{x}||$ for all $\mathbf{x}$ in $\mathbb{R}^n$.
- According to the Best Approximation Theorem, $\hat{\mathbf{b}} = \text{proj}_{Col A}\mathbf{b}$ is the closest point in $\text{Col } A$ to $\mathbf{b}$, so the solution of the least squares problem is to find $\hat{\mathbf{x}}$ such that $A\hat{\mathbf{x}} = \hat{\mathbf{b}}$. According to the Orthogonal Decomposition Theorem, $\mathbf{b} - \hat{\mathbf{b}} = \mathbf{b} - A\hat{\mathbf{x}}$ is the orthogonal to $\text{Col } A$, which means $A^T(\mathbf{b} - A\hat{\mathbf{x}}) = \mathbf{0}$.
- Therefore, the solution is solving this equation $A^T A\hat{\mathbf{x}} = A^T\mathbf{b}$, and finally $\hat{\mathbf{x}} = (A^T A)^{-1}A^T\mathbf{b}$.

## Optimization Equation

$$\begin{aligned}
E(K,V,B) &= E_{dist}(K,V) + E_{spring}(K,V) \\
&= \sum_{i=1}^{n} ||\mathbf{x}_i - \phi_V(\mathbf{b}_i)||^2 + E_{spring}(K,V) \\
&= \sum_{i=1}^{n} ||\mathbf{x}_i - \phi_V(\mathbf{b}_i)||^2 + \sum_{\{j,k\} \in K} \kappa||\mathbf{v}_j - \mathbf{v}_k||^2
\end{aligned}$$

- The $K$ and $B$ are fixed. You need to solve $\arg\min_V E(K,V,B)$
- $\mathbf{x}_i$ is the randomly sampled points from a mesh. $\mathbf{b}_i$ is the barycentric coordinates for the newly fitted point $V$ and other two points to form a triangle. $\phi_V$ calculates the closest point to the triangle from the data point $\mathbf{x}_i$.

## Application to the Least-Squares Problem for $E_{dist}$

- As for the $E_{dist}(K,V)$, we want to fit a vertex $V$ to the data point $\mathbf{x}$, so we should minimize $||\mathbf{x} - \phi_V(\mathbf{b})||^2$. However, it's not easy to directly represent the equation in the linear least squares problem, because of the $\phi_V(\mathbf{b})$.
- To put it simply, we have three vertices of a triangle such as $V$, $P_1$, $P_2$, where the $V$ is the one to change to minimize the energy. When we project the data point $\mathbf{x}$ to the triangle $VP_1P_2$, we will get one point $\mathbf{y}$ on the triangle which is also closest to the data point $\mathbf{x}$, and the $\mathbf{y}$ is the $\phi_V(\mathbf{b})$. so we want to the change $V$ so that the distance between $\mathbf{x}$ and $\mathbf{y}$ should be minimized.
- If we represent the above statement as the equation,

$$\min_V ||(uV + vP_1 + wP_2) - \mathbf{x}|| \tag{1}$$

,where $u, v, w$ are the barycentric coordinates evaluated by projecting $\mathbf{x}$ to the triangle $VP_1P_2$. If we rewrite some part of the above equation in the matrix form,

$$uV + vP_1 + wP_2 - \mathbf{x} = [V \quad P_1 \quad P_2] \begin{bmatrix} u \\ v \\ w \end{bmatrix} - \mathbf{x}. \tag{2}$$

However, this matrix equation can't be used for the least-squares problem, because the vector $u, v, w$ will be the one to evaluated in the linear least-squares problem (For example, if the linear least-squares problem is $C\mathbf{r} = \mathbf{d}$, then $\mathbf{r}$ will be the barycentric coordinate vectors.) So, $V$ should be on the place of the barycentric coordinate vector to have an equation like this:

$$[? \quad ? \quad ?]\begin{bmatrix}V_x\\V_y\\V_z\end{bmatrix} - @, \tag{3}$$

so that we can change $V$ to minimize the energy as a linear least-squares problem.

- Because we already know all of the terms in the equation (2), we can complete the equation (3). If we specify the equation (2) in more details,

$$[V \quad P_1 \quad P_2]\begin{bmatrix}u\\v\\w\end{bmatrix} - \mathbf{x} = \begin{bmatrix}uV_x + vP_{1,x} + wP_{2,x} - \mathbf{x}_x\\uV_y + vP_{1,y} + wP_{2,y} - \mathbf{x}_y\\uV_z + vP_{1,z} + wP_{2,z} - \mathbf{x}_z\end{bmatrix}. \tag{4}$$

  We can rewrite this equation in the form of the equation (3),

$$\begin{bmatrix}u & 0 & 0\\0 & u & 0\\0 & 0 & u\end{bmatrix}\begin{bmatrix}V_x\\V_y\\V_z\end{bmatrix} - (\mathbf{x} - vP_1 - wP_2) \tag{5}$$

  Now, it's also in the form of the linear least-squares problem, where you can evaluate the $V$.

- In the linear least-squares form $C\mathbf{r} = \mathbf{d}$, the solution is $\mathbf{r} = (C^TC)^{-1}C^T\mathbf{d}$, so we can easily know what $C^TC$ and $C^T\mathbf{d}$ are.

$$C^TC = uI^TuI = u^2I$$
$$C^T\mathbf{d} = u(\mathbf{x} - vP_1 - wP_2)$$

  Because $C^TC$ is $u^2I$ and $(C^TC)^{-1}$ is $\frac{1}{u^2}I$, the $\mathbf{r}$ will be finally solved like this

$$\mathbf{r} = (C^TC)^{-1}C^T\mathbf{d} = \frac{1}{u}(\mathbf{x} - vP_1 - wP_2).$$

## Application to the Least-Squares Problem for $E_{spring}$

- As for the $E_{spring}$, we also want to fit a vertex $V_j$ to another vertex $V_k$ to minimize $\kappa||V_j - V_k||^2$. To do this in the least-squares problem, we need to move the $\kappa$ inside the length operator like this $\kappa||V_j - V_k||^2 = ||\sqrt{\kappa}(V_j - V_k)||^2$ .
- We can minimize the above equation in the linear least-squares form like this

$$\begin{bmatrix}\sqrt{\kappa} & 0 & 0\\0 & \sqrt{\kappa} & 0\\0 & 0 & \sqrt{\kappa}\end{bmatrix}\begin{bmatrix}V_{j,x}\\V_{j,y}\\V_{j,z}\end{bmatrix} - \sqrt{\kappa}\begin{bmatrix}V_{k,x}\\V_{k,y}\\V_{k,z}\end{bmatrix},$$

  so that you can solve the linear least-squares problem.

- In the linear least-squares form $C\mathbf{r} = \mathbf{d}$,

$$C^TC = \sqrt{\kappa}I^T\sqrt{\kappa}I = \kappa I$$
$$C^T\mathbf{d} = \sqrt{\kappa}I^T\sqrt{\kappa}V_k = \kappa V_k$$

  the $\mathbf{r}$ will be finally solved like this

$$\mathbf{r} = \frac{1}{\kappa}I\kappa V_k = V_k$$

- Here the $\mathbf{r}$ is just the $V_k$, but you need to consider that the $\mathbf{r}$ will be cumulated with other vertices.

## Residual Sum of Squares (RSS) for the linear regression.

According to the comment in the mesh optimization code, he says that the RSS $||C\mathbf{r} - \mathbf{d}||^2$ can be converted like this :

$$RSS = ||\mathbf{d}||^2 - ||C\mathbf{r}||^2 = ||\mathbf{d}||^2 - \mathbf{r}^T\mathbf{r}||C||^2.$$

He says that he got this from Werner Stuetzle, the last paper author.  This term is used later for optimizing simplicial complex when using edge operations. I don't know exactly what his RSS is based on. I could use the normal RSS later, and compare the result.

## Solving in the global context.

Now I study `global_fit` of the mesh optimization. It's similar to the `local_fit`, but you consider all of the vertices in the linear least-squares problem format $||A\mathbf{x} - \mathbf{b}||^2$.

For example, If we have $d$ randomly sampled points and $e$ edges of a mesh, the number of the row of $A$ in the linear least squares problem is $m = d + e$. The number of the column of $A$ is the number of vertices $n$. So, the dimension of $A$ is $m \times n$. $\mathbf{x}$ is the vertices of the mesh to be optimized. So its dimension is $n \times 3$. $\mathbf{b}$ is the randomly sampled points, so its dimension is $m \times 3$ with the row for edges begin zero. To visualize the encoding of the data into the matrix (you need to forget the symbols in the linear least squares problem here),

$$\underbrace{\begin{bmatrix} 0 & \cdots & 0 & b_{0,u} & 0 & \cdots & b_{0,v} & 0 & \cdots & b_{0,w} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & b_{d-1,u} & \cdots & 0 & \cdots & \cdots & b_{d-1,v} & \cdots & b_{d-1,w} & \cdots & 0 \\ 0 & \cdots & 0 & \cdots & \sqrt{\kappa} & \cdots & \cdots & 0 & \cdots & -\sqrt{\kappa} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \sqrt{\kappa} & \cdots & \cdots & \cdots & 0 & -\sqrt{\kappa} & \cdots & 0 & \cdots & \cdots & 0 \end{bmatrix}}_{\mathbb{R}^{m \times n}} \underbrace{\begin{bmatrix} \mathbf{v}_{0,x} & \mathbf{v}_{0,y} & \mathbf{v}_{0,z} \\ \vdots & \vdots & \vdots \\ \mathbf{v}_{n-1,x} & \mathbf{v}_{n-1,y} & \mathbf{v}_{n-1,z} \end{bmatrix}}_{\mathbb{R}^{n \times 3}} - \underbrace{\begin{bmatrix} \mathbf{x}_{0,x} & \mathbf{x}_{0,y} & \mathbf{x}_{0,z} \\ \vdots & \vdots & \vdots \\ \mathbf{x}_{d-1,x} & \mathbf{x}_{d-1,y} & \mathbf{x}_{d-1,z} \\ 0 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \end{bmatrix}}_{\mathbb{R}^{m \times 3}}$$

$b_{0,u/v/w}$ is the barycentric coordinates evaluated by projecting the randomly sampled point $\mathbf{x}_0$ to a closest face of the mesh. The barycentric coordinates is encoded in the location of the indices of the vertices from the closest face of a triangle. $\mathbf{v}$ is the vertices of the mesh. The spring energy function $E_{spring}(K, V)$ is encoded from the $d + 1$ row with $\sqrt{\kappa}$ and $-\sqrt{\kappa}$ in the indices of the two vertices of each edge. Because the matrix is sparse, the authors solved this with their own Sparse Conjugate Gradient approach.