We compute the vertices (poses) using the given path:

$$x_{k+1} = x_k + \Delta x \cos\theta_k - \Delta y \sin\theta_k$$

$$y_{k+1} = y_k + \Delta y \cos\theta_k + \Delta x \sin\theta_k$$

We aim to minimise:

$$F(x) = f(x)^T \Omega f(x) \quad [\text{objective function}]$$

$$= \arg\min_x \left\{ \sum_i \| f(x_i, u_i) - x_{i+1} \|^2_{\Sigma_i} \right. \text{ — i}$$

gaussian parameter

$$\text{ii} \text{ — } + \left. \sum_{ij} \| f(x_i, u_{ij}) - x_j \|^2_{\Lambda_{ij}} \right\} \rightarrow \text{gaussian parameter}$$

$f(x) \rightarrow$ vector of stacked residuals.

i $\rightarrow$ odometry constraints   ii $\rightarrow$ loop closure constraints.

$$f(x) = \begin{bmatrix} f(x_0, u_1) - x_1 \\ f(x_1, u_2) - x_2 \\ \vdots \\ f(x_i, u_{ij}) - x_j \\ \vdots \end{bmatrix} \begin{matrix} \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} \text{odometry residuals} \\ \left.\begin{matrix} \\ \end{matrix}\right\} \text{loop closure residuals.} \end{matrix}$$

$f(x_i, u_i)$ is defined by the given motion model.

Ω is the information matrix dictating how much one should trust a reading. The higher the value, the better.

$$\Omega = \begin{bmatrix} \Sigma_{11} & 0 & \cdots & & 0 \\ 0 & \Sigma_{22} & & & \vdots \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \Sigma_{kk} & \Lambda_{ij} \\ & & \cdots & & 0 \end{bmatrix}^{-1}$$

where the diagonal elements are the covariances of the readings.

In 2.1: The loss is optimised with respect to the parameters x_i, y_i, θ_i for each and every point.

Assuming no loop closure, the $f(x)$ vector looks like:

$$f(x) = \begin{bmatrix} f(x_0, u_1) - x_1 \\ f(y_1, u_2) - y_2 \\ f(\theta_1, u_3) - \theta_2 \end{bmatrix} = \begin{bmatrix} x_1 + \Delta x \cos\theta_1 - \Delta y \sin\theta_1 - x_2 \\ y_1 + \Delta x \sin\theta_1 + \Delta y \cos\theta_1 - y_2 \\ \theta_1 + \Delta\theta - \theta_2 \end{bmatrix}$$

The LM ~~based~~ descent based algorithm also requires the jacobian of the parameters as Δx is found from $(J^T\Omega J + \lambda I)\Delta x = -J^T\Omega f(x)$.

The Jacobian is constructed from the partial derivatives with respect to the parameters, that is the coordinates.

$$\text{Jacobian}(J) = \begin{bmatrix} \frac{\partial f(x)}{\partial x_1} & \frac{\partial f(x)}{\partial y_1} & \frac{\partial f(x)}{\partial \theta_1} & \frac{\partial f(x)}{\partial x_2} & \frac{\partial f(x)}{\partial y_2} & \frac{\partial f(x)}{\partial \theta_2} \end{bmatrix}$$

$$J = \begin{bmatrix} 1 & 0 & -\Delta x \sin\theta - \Delta y \cos\theta & -1 & 0 & 0 \\ 0 & 1 & -\Delta y \sin\theta + \Delta x \cos\theta & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \end{bmatrix}$$

Note: ~~If there~~ for loop closure constraints, the jacobians rows' would increase in number. The "anchor" constraint adds a row as it states the starting points of the trajectory. If there are m poses and n loop closures, the resulting dimension of the jacobian will be:

$$J_{((m+n)*3) \times (m*3)}$$

The anchor constraint is given the highest confidence value (lowest variance) as we know that this is where the trajectory starts. This point shouldn't change.

The initial guesses (given in edges.txt) are optimised by our chosen optimisation method with the help of the jacobian and residual vector.