Homework 3: Solvers

Christopher Klammer
cklammer@andrew.cmu.edu

April 1, 2023

1 Linear SLAM

1.1 Measurement Function

1.1.1 Odometry Measurement

The odometry measurement should just be the vector from the pose at time t to the pose at time $t+1$

$$h_o(\mathbf{r^t}, \mathbf{r^{t+1}}) = \begin{bmatrix} r_x^{t+1} - r_x^t & r_y^{t+1} - r_y^t \end{bmatrix}^T$$

$$H_o = \begin{bmatrix} \frac{\partial r_x^{t+1} - r_x^t}{\partial r_x^t} & \frac{r_x^{t+1} - r_x^t}{\partial r_y^t} & \frac{\partial r_x^{t+1} - r_x^t}{\partial r_x^{t+1}} & \frac{\partial r_x^{t+1} - r_x^t}{\partial r_y^{t+1}} \\ \frac{\partial r_y^{t+1} - r_y^t}{\partial r_x^t} & \frac{\partial r_y^{t+1} - r_y^t}{\partial r_y^t} & \frac{\partial r_y^{t+1} - r_y^t}{\partial r_x^{t+1}} & \frac{\partial r_y^{t+1} - r_y^t}{\partial r_y^{t+1}} \end{bmatrix}$$

$$H_o = \begin{bmatrix} -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix}$$

1.1.2 Landmark Measurement

This will be extremely similar, just find the vector from the robot's position to the landmark

$$h_l(\mathbf{r^t}, \mathbf{l^k}) = \begin{bmatrix} l_x^k - r_x^t & l_y^k - r_y^t \end{bmatrix}^T$$

$$H_l = \begin{bmatrix} \frac{\partial l_x^k - r_x^t}{\partial r_x^t} & \frac{\partial l_x^k - r_x^t}{\partial r_y^t} & \frac{\partial l_x^k - r_x^t}{\partial l_x^k} & \frac{\partial l_x^k - r_x^t}{\partial l_y^k} \\ \frac{\partial l_y^k - r_y^t}{\partial r_x^t} & \frac{\partial l_y^k - r_y^t}{\partial r_y^t} & \frac{\partial l_y^k - r_y^t}{\partial l_x^k} & \frac{\partial l_y^k - r_y^t}{\partial l_y^k} \end{bmatrix}$$

$$H_l = \begin{bmatrix} -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & -1 \end{bmatrix}$$

1.2 Solvers and Sparsity

1.2.1 2D Linear Data

Timing:

PInv	LU	LU COLAMD	QR	QR COLAMD
4.900 sec	0.073 sec	0.186 sec	1.074 sec	1.113 sec

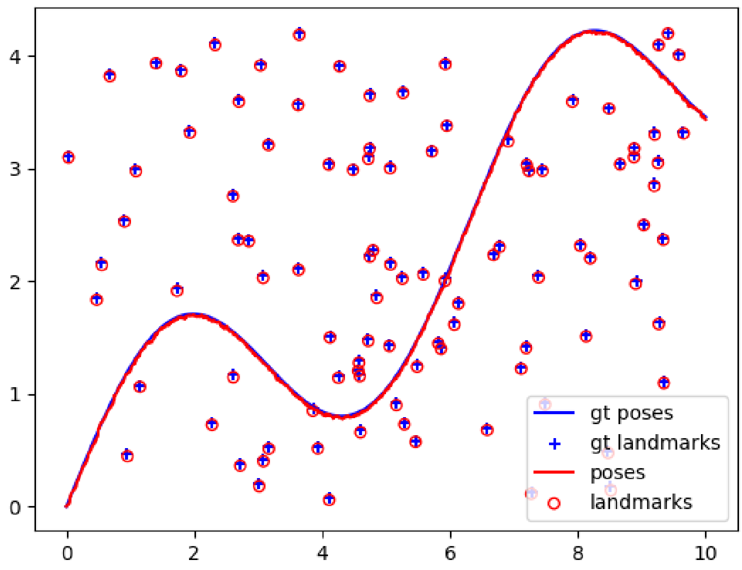

Figure 1: Trajectory for PInv Solver

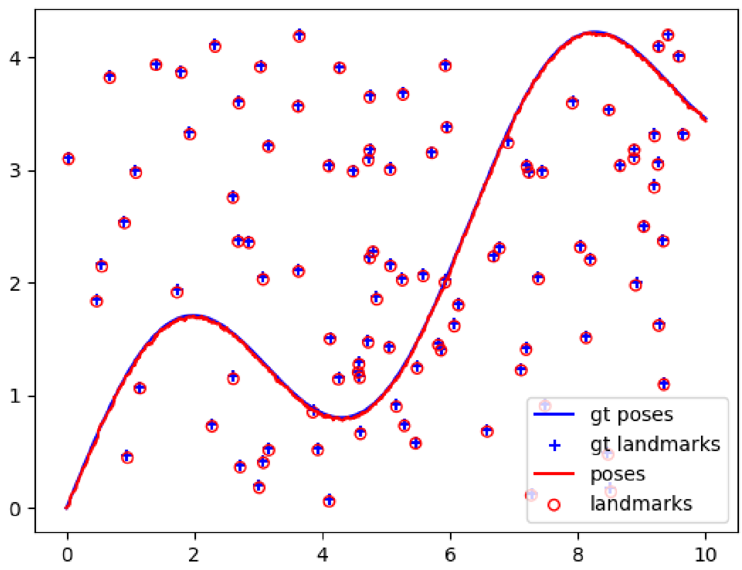

Figure 2: Trajectory for LU Solver

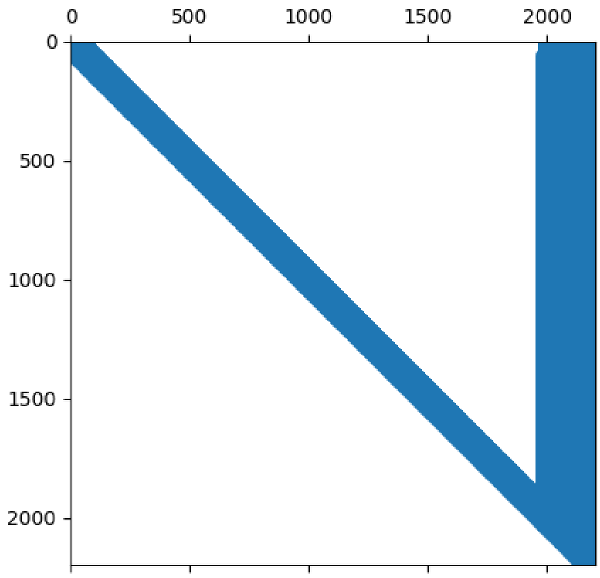

Figure 3: Sparsity for LU Solver

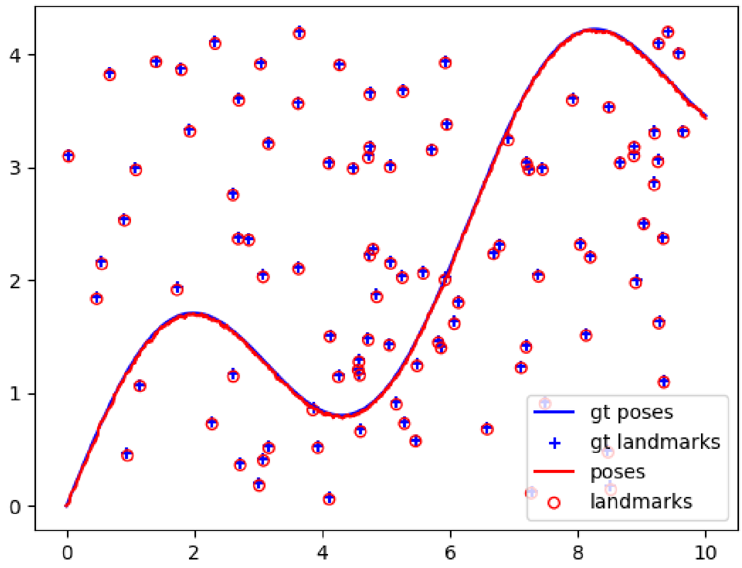

Figure 4: Trajectory for LU COLAMD Solver

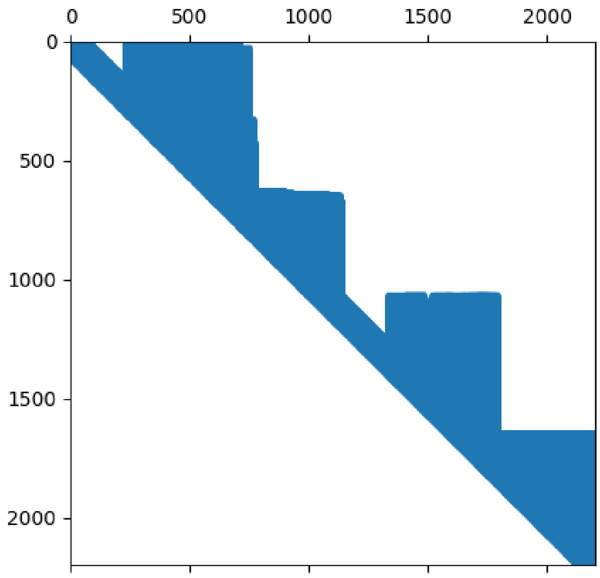

Figure 5: Sparsity for LU COLAMD Solver

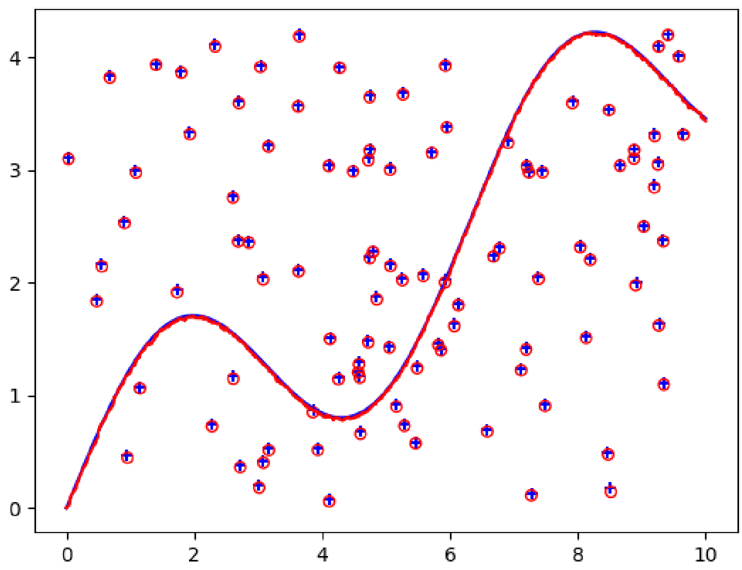

Figure 6: Trajectory for QR Solver

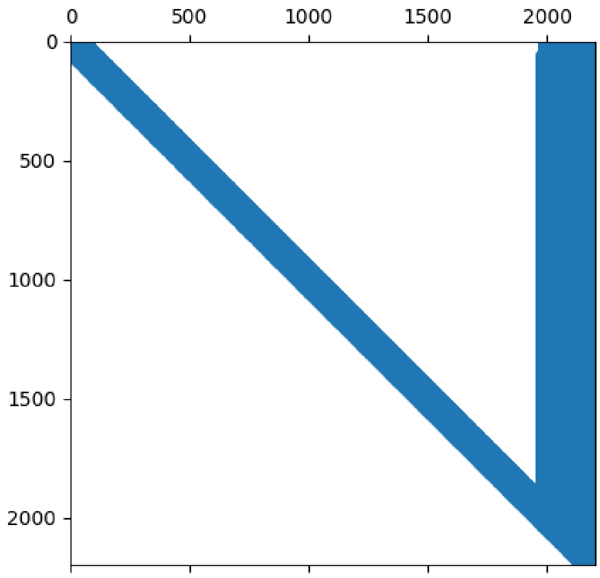

Figure 7: Sparsity for QR Solver

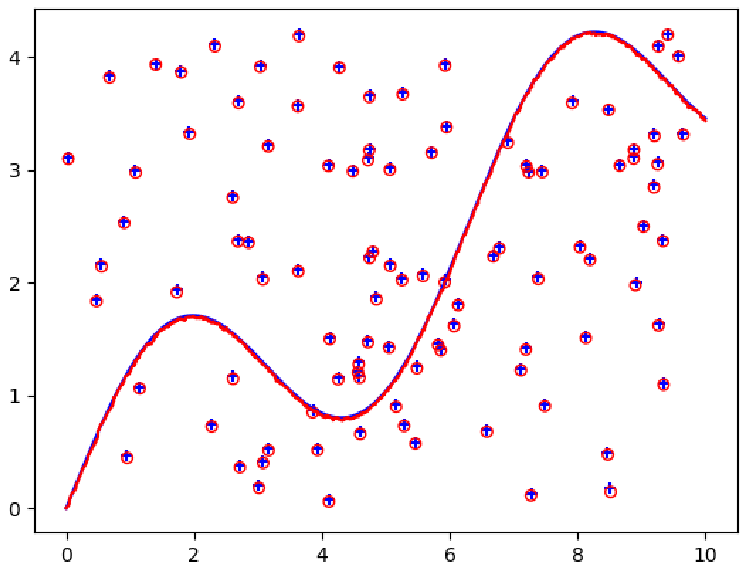

Figure 8: Trajectory for QR COLMAD Solver

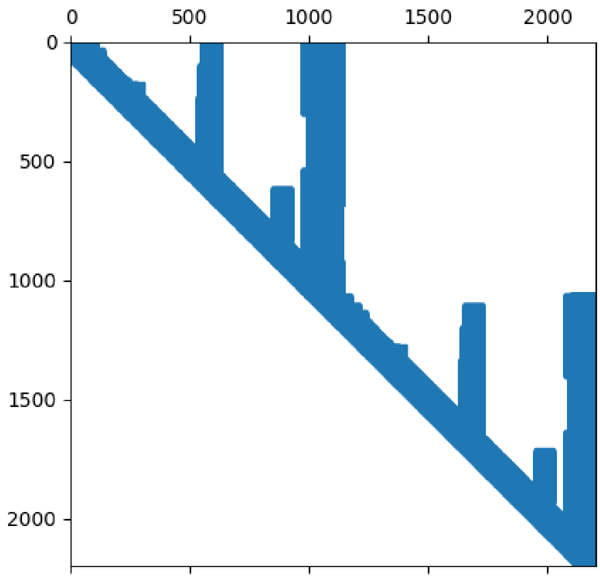

Figure 9: Sparsity for QR COLMAD Solver

1.2.2 2D Linear Discussion

First off, we see here that the psuedo-inverse is the slowest, the LU solver is the fastest, and the QR solver is in the middle. We also see that the COLAMD methods do not cause any speedup.

In the case of a linear trajectory, we do not have any redundant observations. Each landmark that we see is relatively new and, since the trajectory is fairly linear, we see that our factorized matricies are sparse and close to diagonal. Since this is the case, we do not see performance increase by using the COLAMD ordering. In my opinion, it seems that no further sparsification can be gained with COLAMD so the reordering acts as an overhead making the performance worse.

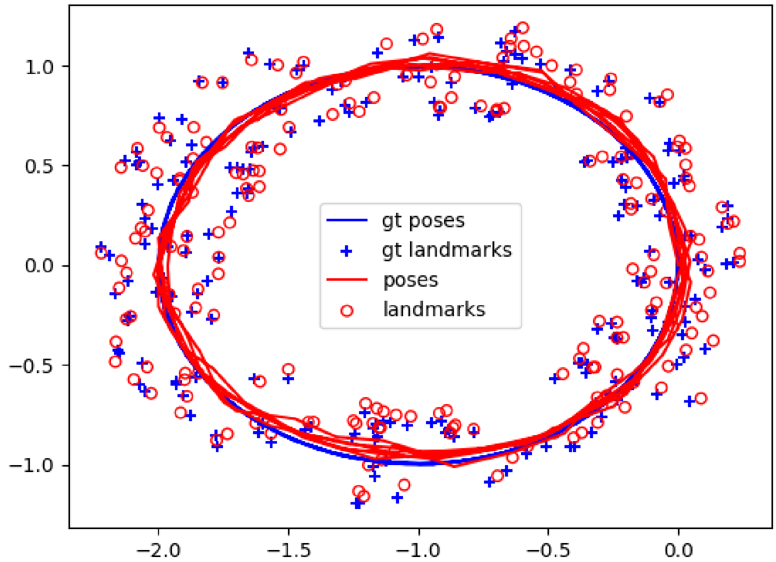

Figure 10: Trajectory for PInv Solver

1.2.3 2D Linear Loop Data

Timing:

PInv	LU	LU COLAMD	QR	QR COLAMD
0.451 sec	0.056 sec	0.009 sec	0.616 sec	0.063 sec

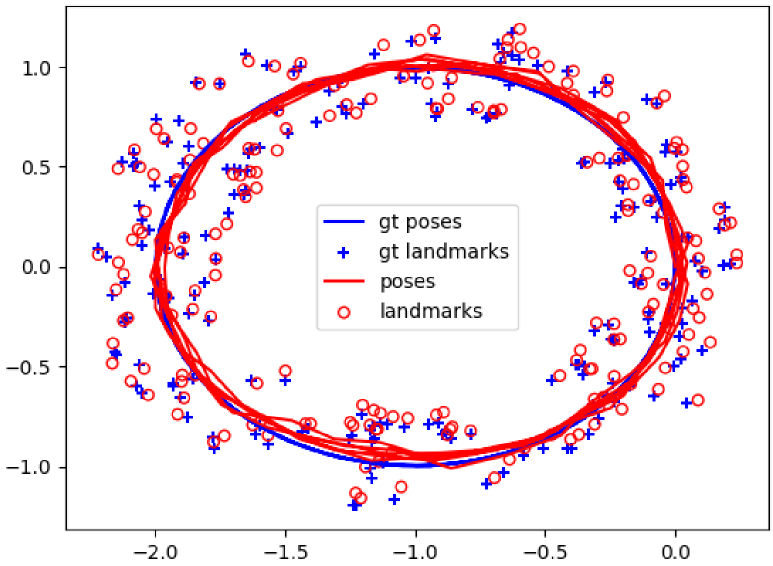

Figure 11: Trajectory for LU Solver

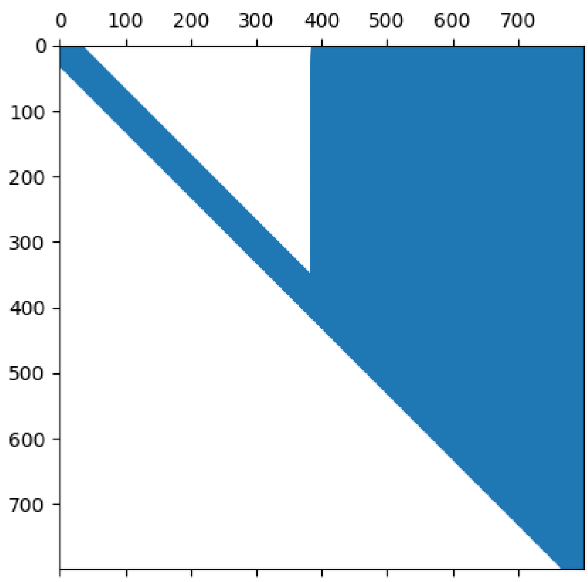

Figure 12: Sparsity for LU Solver

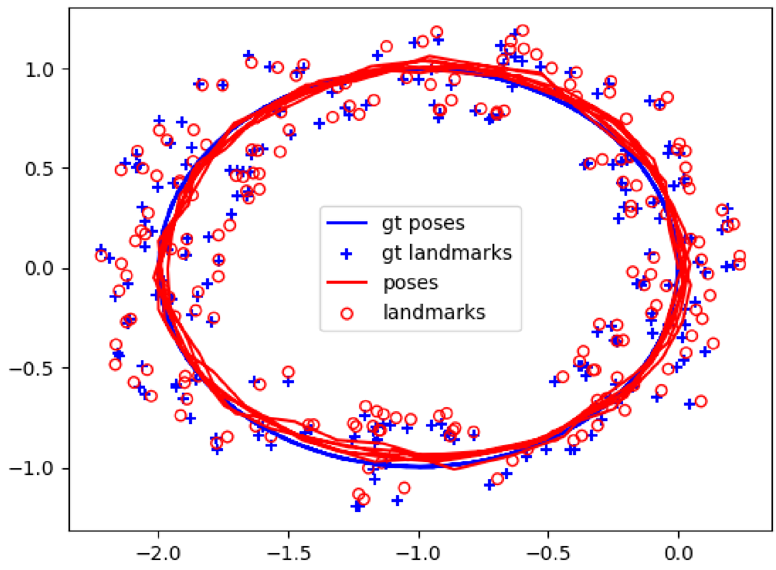

Figure 13: Trajectory for LU COLAMD Solver

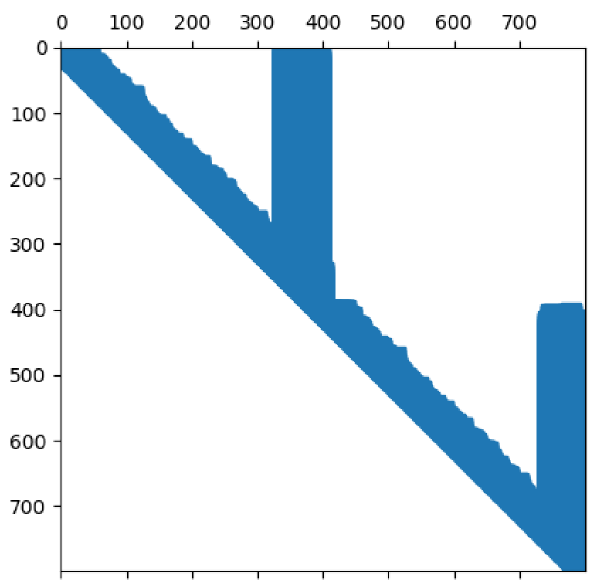

Figure 14: Sparsity for LU COLAMD Solver

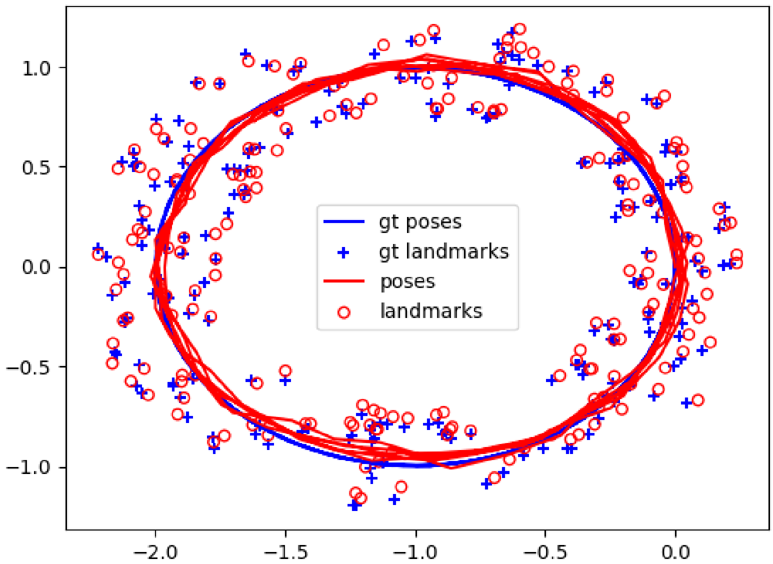

Figure 15: Trajectory for QR Solver

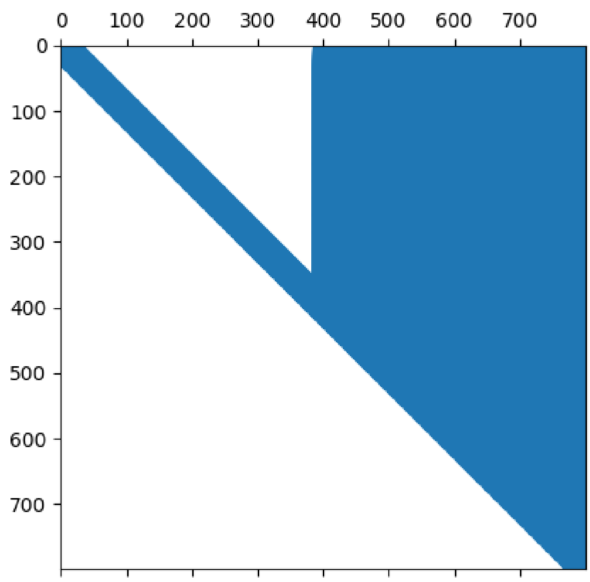

Figure 16: Sparsity for QR Solver

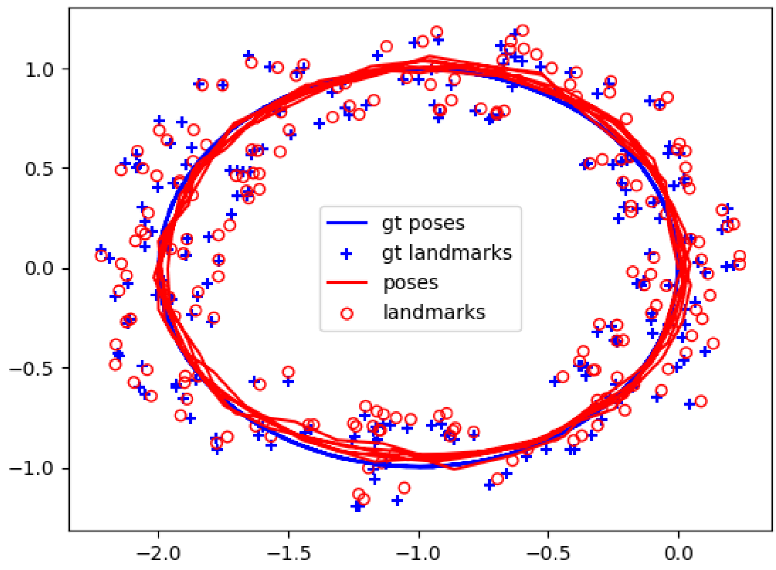

Figure 17: Trajectory for QR COLMAD Solver

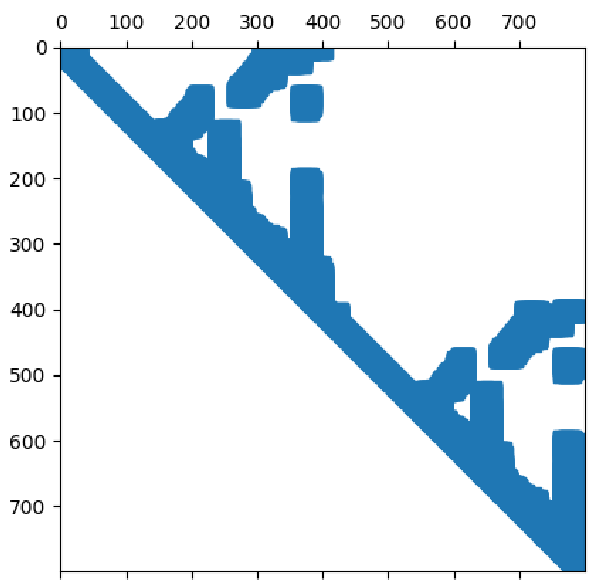

Figure 18: Sparsity for QR COLMAD Solver

1.2.4 2D Linear Loop Discussion

In general, we see the same relative ordering in that the psuedo-inverse is the slowest, the LU solver is the quickest, and the QR solver is in the middle. However, in this case, we see that COLAMD offers massive speedups.

In this case, we appear to move in a loop where we visit the same landmarks multiple times and we also see similar odometry measurements as well because we are moving in the same loop. This creates a much more dense matrix, especially compared to the linear case.

Here, we see large speedups in performance when using the COLAMD ordering which helps sparsify the matrix by removing more redudant measurements. For example, seeing the same landmark from the same pose is probably not useful, we can essentially remove or generally combine these redundant rows in order to create a more sparse representation.

2 Non-Linear SLAM

2.1 Measurement Function

2.1.1 Landmark Measurement

Notation:

$$dx = l_x^k - r_x^t$$
$$dy = l_y^k - r_y^t$$

Calculation:

$$H_l = \begin{bmatrix} \frac{\partial\theta}{\partial r_x^t} & \frac{\partial\theta}{\partial r_y^t} & \frac{\partial\theta}{\partial l_x^k} & \frac{\partial\theta}{\partial l_y^k} \\ \frac{\partial d}{\partial r_x^t} & \frac{\partial d}{\partial r_y^t} & \frac{\partial d}{\partial l_x^k} & \frac{\partial d}{\partial l_y^k} \end{bmatrix}$$

$$\frac{\partial\theta}{\partial r_x^t} = \frac{dx^2}{dx^2 + dy^2} * \frac{dy}{dx^2} = \frac{dy}{dx^2 + dy^2}$$

$$\frac{\partial\theta}{\partial r_y^t} = \frac{dx^2}{dx^2 + dy^2} * -\frac{dx}{dx^2} = -\frac{dx}{dx^2 + dy^2}$$

$$\frac{\partial\theta}{\partial l_x^k} = \frac{dx^2}{dx^2 + dy^2} * \frac{-dy}{dx^2} = -\frac{dy}{dx^2 + dy^2}$$

$$\frac{\partial\theta}{\partial l_y^k} = \frac{dx^2}{dx^2 + dy^2} * \frac{dx}{dx^2} = \frac{dx}{dx^2 + dy^2}$$

$$\frac{\partial d}{\partial r_x^t} = \frac{1}{2} * (dx^2 + dy^2)^{-\frac{1}{2}} * -2 * dx = \frac{-dx}{(dx^2 + dy^2)^{1/2}}$$

$$\frac{\partial d}{\partial r_y^t} = \frac{1}{2} * (dx^2 + dy^2)^{-\frac{1}{2}} * -2 * dy = \frac{-dy}{(dx^2 + dy^2)^{1/2}}$$

$$\frac{\partial d}{\partial l_x^k} = \frac{1}{2} * (dx^2 + dy^2)^{-\frac{1}{2}} * 2 * dx = \frac{dx}{(dx^2 + dy^2)^{1/2}}$$

$$\frac{\partial d}{\partial l_y^k} = \frac{1}{2} * (dx^2 + dy^2)^{-\frac{1}{2}} * 2dy = \frac{dy}{(dx^2 + dy^2)^{1/2}}$$

$$H_l = \begin{bmatrix} \frac{dy}{dx^2+dy^2} & \frac{-dx}{dx^2+dy^2} & \frac{-dy}{dx^2+dy^2} & \frac{dx}{dx^2+dy^2} \\ \frac{-dx}{(dx^2+dy^2)^{1/2}} & \frac{-dy}{(dx^2+dy^2)^{1/2}} & \frac{dx}{(dx^2+dy^2)^{1/2}} & \frac{dy}{(dx^2+dy^2)^{1/2}} \end{bmatrix}$$

2.2 Solver

2.2.1 Linear vs. Non-Linear SLAM

The difference between the linear and non-linear SLAM problem lies in a few key aspects here. I will first discuss linear SLAM and why it is simpler that non-linear SLAM. Then I will discuss the complexities of non-linear SLAM and why the optimization must be formulated differently.

Linear least squares is a fairly simple formulation, where $h_i(x) = Hx + h_0$ and we assume that there is no rotations that would make the formulation non-linear. At this point we can solve for x that minimizes the least squares solution $||Ax - b||^2$ where A is the jacobian of our measurement function, x is the state and b is $z - h_0$ or the true measurement with some constant from the measurement function.

The first being that with non-linear SLAM, we are inherently working with a non-convex problem with a non-direct solution. Thus, we must perform an approximation for the measurement function. This approximation is carried out in the form of a first order taylor series expansion about a particular linearization point x^0 of the form:

$$z_{est} = h_i(x) = h(x^0 + \delta) = h(x^0) + \frac{\partial h(x)}{x} * (x - x^0)$$

Here, this linearization point needs to be dictated by essentially a current guess or estimate and will be iteratively improved to move the estimated and actual measurements closer together. This methodology is inherently dependent on a good initialization and its covergence (or at least its speed of convergence) is highly sensitive to the initial estimate of the state x^0. Knowing this formulation we can find the residual of our measurement and estimated approximation in order to iteratively solve:

$$||z_{est} = z||^2 = ||h(x^0) + \frac{\partial h(x)}{\partial x}|_{x=x_0} * (x - x^0) - z||^2 = ||A\delta - b||^2$$

where:

$$A = h(x^0) + \frac{\partial h(x)}{\partial x}|_{x=x_0}$$

$$b = z$$

$$\delta = x - x_0$$

2.2.2 Visualization

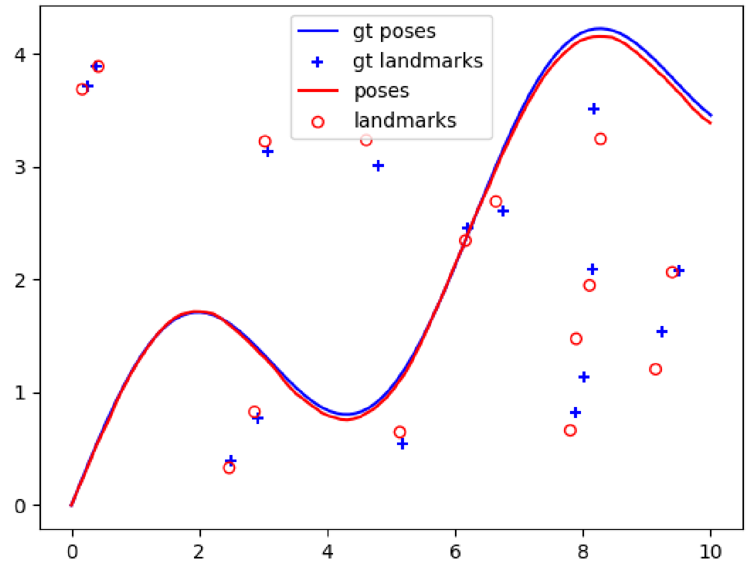

Figure 19: QR COLAMD Before Optimization

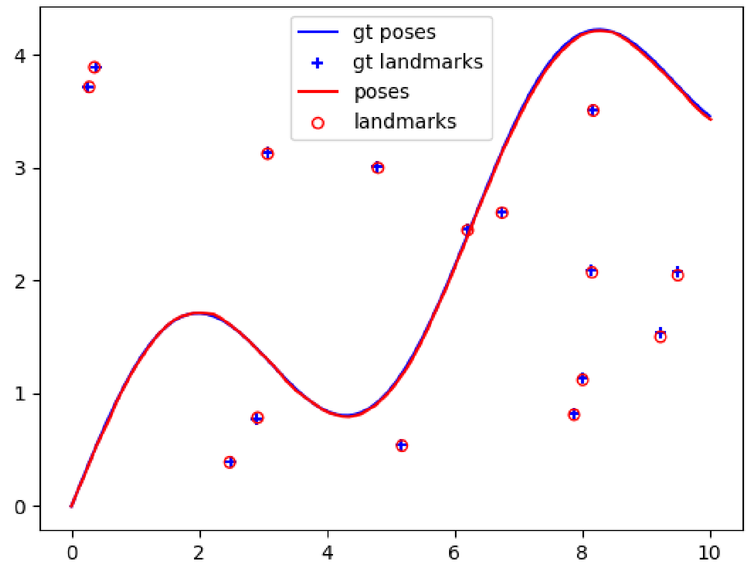

Figure 20: QR COLAMD After Optimization