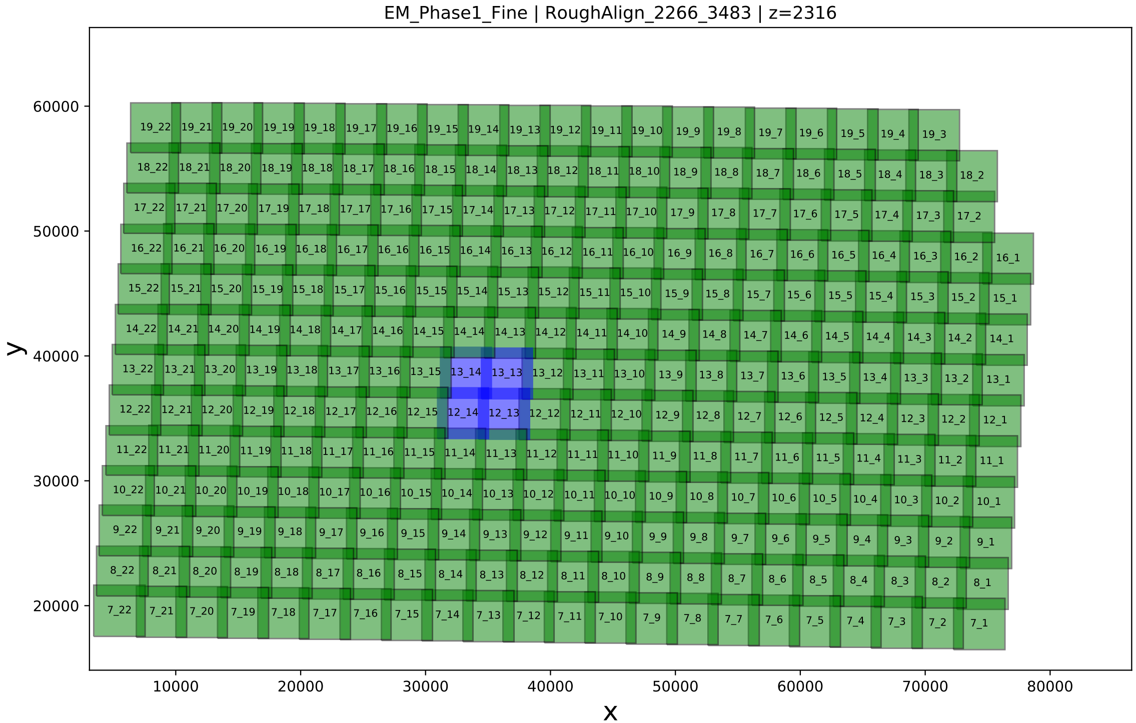
4tile example

small example

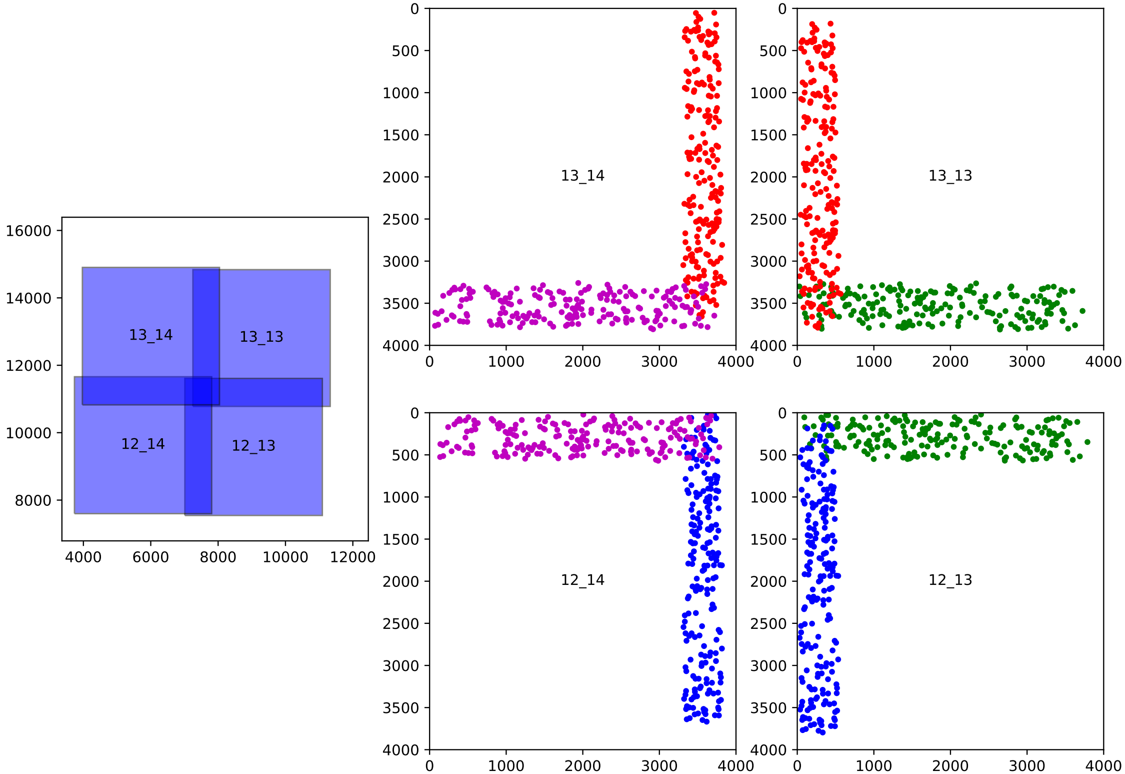
- understand matrix construction, code and linear algebra

Dan Kapner
10/2/2017

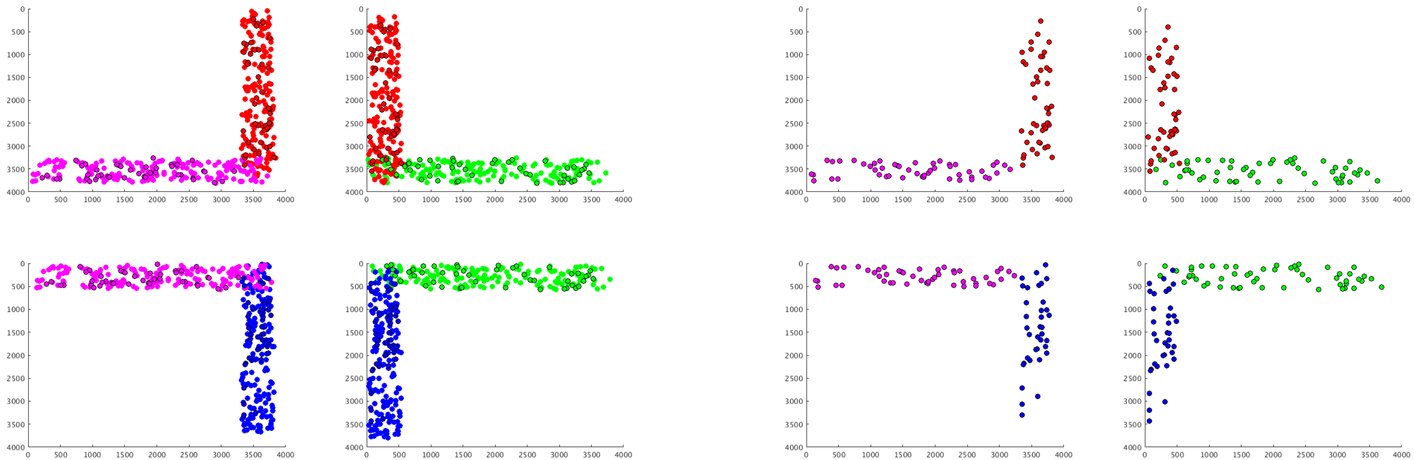
Source of Data



4 tiles and point matches



Filtering



MATLAB filters down the pointmatch collection from 200 points per tile pair to around 50

There is some random aspect to the filtering.

tile 1	tile 2
$({}^1x_1, {}^1y_1)$	$({}^2x_1, {}^2y_1)$
$({}^1x_2, {}^1y_2)$	$({}^2x_2, {}^2y_2)$
$({}^1x_3, {}^1y_3)$	$({}^2x_3, {}^2y_3)$
$({}^1x_4, {}^1y_4)$	$({}^2x_4, {}^2y_4)$
...	...

Naming and algebra

affine transformation

$$ {}^1u_1 = a_1\, {}^1x_1 + b_1\, {}^1y_1 + c_1 \tag{1} $$

$$ {}^1v_1 = d_1\, {}^1x_1 + e_1\, {}^1y_1 + f_1 \tag{2} $$

requirement for alignment

$$ {}^1u_1 = {}^2u_1 \tag{3} $$

$$ {}^{1}v_1 = {}^{2}v_1 \tag{4} $$

explicitly

$$ a_1\,{}^{1}x_1 + b_1\,{}^{1}y_1 + c_1 = a_2\,{}^{2}x_1 + b_2\,{}^{2}y_1 + c_2 \tag{5} $$

$$ d_1\,{}^{1}x_1 + e_1\,{}^{1}y_1 + f_1 = d_2\,{}^{2}x_1 + e_2\,{}^{2}y_1 + f_2 \tag{6} $$

and in matrix form, this is:

$$ \begin{bmatrix} -\,{}^{1}x_1 & -\,{}^{1}y_1 & -1 & {}^{2}x_1 & {}^{2}y_1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\,{}^{1}x_1 & -\,{}^{1}y_1 & -1 & {}^{2}x_1 & {}^{2}y_1 & 1 \end{bmatrix} \times \begin{bmatrix} a_1 \\ b_1 \\ c_1 \\ a_2 \\ b_2 \\ c_2 \\ d_1 \\ e_1 \\ f_1 \\ d_2 \\ e_2 \\ f_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} $$

and, now scaling to more than 1 point match pair:

$$\begin{bmatrix} -{}^1x_1 & -{}^1y_1 & -1 & {}^2x_1 & {}^2y_1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -{}^1x_2 & -{}^1y_2 & -1 & {}^2x_2 & {}^2y_2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -{}^1x_3 & -{}^1y_3 & -1 & {}^2x_3 & {}^2y_3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -{}^1x_4 & -{}^1y_4 & -1 & {}^2x_4 & {}^2y_4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & ... & & & & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & -{}^1x_1 & -{}^1y_1 & -1 & {}^2x_1 & {}^2y_1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -{}^1x_2 & -{}^1y_2 & -1 & {}^2x_2 & {}^2y_2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -{}^1x_3 & -{}^1y_3 & -1 & {}^2x_3 & {}^2y_3 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -{}^1x_4 & -{}^1y_4 & -1 & {}^2x_4 & {}^2y_4 & 1 \\ & & & & & & & & ... & & & \end{bmatrix} \times \begin{bmatrix} a_1 \\ b_1 \\ c_1 \\ a_2 \\ b_2 \\ c_2 \\ d_1 \\ e_1 \\ f_1 \\ d_2 \\ e_2 \\ f_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ ... \\ 0 \\ 0 \\ 0 \\ 0 \\ ... \end{bmatrix}$$

we can rearrange the columns to keep all the tile coordinates in one place:

$$
\begin{bmatrix}
-{}^1x_1 & -{}^1y_1 & -1 & 0 & 0 & 0 & {}^2x_1 & {}^2y_1 & 1 & 0 & 0 & 0 \\
-{}^1x_2 & -{}^1y_2 & -1 & 0 & 0 & 0 & {}^2x_2 & {}^2y_2 & 1 & 0 & 0 & 0 \\
-{}^1x_3 & -{}^1y_3 & -1 & 0 & 0 & 0 & {}^2x_3 & {}^2y_3 & 1 & 0 & 0 & 0 \\
-{}^1x_4 & -{}^1y_4 & -1 & 0 & 0 & 0 & {}^2x_4 & {}^2y_4 & 1 & 0 & 0 & 0 \\
 & & & \dots & & & & & & & & \\
0 & 0 & 0 & -{}^1x_1 & -{}^1y_1 & -1 & 0 & 0 & 0 & {}^2x_1 & {}^2y_1 & 1 \\
0 & 0 & 0 & -{}^1x_2 & -{}^1y_2 & -1 & 0 & 0 & 0 & {}^2x_2 & {}^2y_2 & 1 \\
0 & 0 & 0 & -{}^1x_3 & -{}^1y_3 & -1 & 0 & 0 & 0 & {}^2x_3 & {}^2y_3 & 1 \\
0 & 0 & 0 & -{}^1x_4 & -{}^1y_4 & -1 & 0 & 0 & 0 & {}^2x_4 & {}^2y_4 & 1 \\
 & & & & & & & & \dots & & &
\end{bmatrix}
\times
\begin{bmatrix} a_1 \\ b_1 \\ c_1 \\ d_1 \\ e_1 \\ f_1 \\ a_2 \\ b_2 \\ c_2 \\ d_2 \\ e_2 \\ f_2 \end{bmatrix}
=
\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \dots \\ 0 \\ 0 \\ 0 \\ 0 \\ \dots \end{bmatrix}
$$

this is starting to look like Khaled's presentation.
change all the signs (absorb into the a,b,c,d parameters)

$$
\begin{bmatrix} {}^{1,2}P & -{}^{1,2}Q \end{bmatrix} \times \begin{bmatrix} T_1 \\ T_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}
$$

where, for example

$$\left[{}^{1,2}P\right] = \begin{bmatrix} -{}^{1}x_1 & -{}^{1}y_1 & -1 & 0 & 0 & 0 \\ -{}^{1}x_2 & -{}^{1}y_2 & -1 & 0 & 0 & 0 \\ -{}^{1}x_3 & -{}^{1}y_3 & -1 & 0 & 0 & 0 \\ -{}^{1}x_4 & -{}^{1}y_4 & -1 & 0 & 0 & 0 \\ \dots & & & & & \\ 0 & 0 & 0 & -{}^{1}x_1 & -{}^{1}y_1 & -1 \\ 0 & 0 & 0 & -{}^{1}x_2 & -{}^{1}y_2 & -1 \\ 0 & 0 & 0 & -{}^{1}x_3 & -{}^{1}y_3 & -1 \\ 0 & 0 & 0 & -{}^{1}x_4 & -{}^{1}y_4 & -1 \\ & & & \dots & & \end{bmatrix}$$

and

$$T_1 = \begin{bmatrix} a_1 \\ b_1 \\ c_1 \\ d_1 \\ e_1 \\ f_1 \end{bmatrix}$$

and where the following is a column-wise concatenation

$$\begin{bmatrix} {}^{1,2}P & -{}^{1,2}Q \end{bmatrix}$$

and where the following is a row-wise concatenation

$$\begin{bmatrix} T_1 \\ T_2 \end{bmatrix}$$

this equation,again, is for 2 tiles:

$$\begin{bmatrix} {}^{1,2}P & -{}^{1,2}Q \end{bmatrix} \times \begin{bmatrix} T_1 \\ T_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

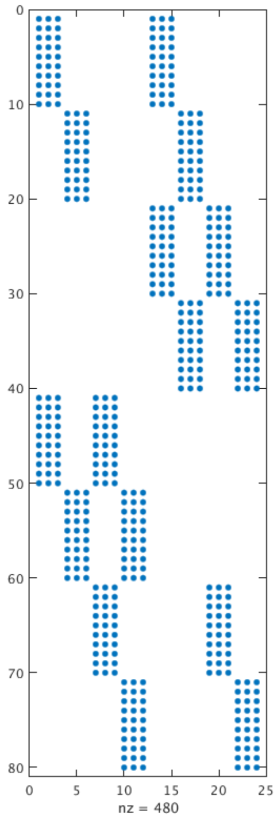
when we expand it to 3 tiles
(with overlaps 1-2, 1-3, but not 2-3)

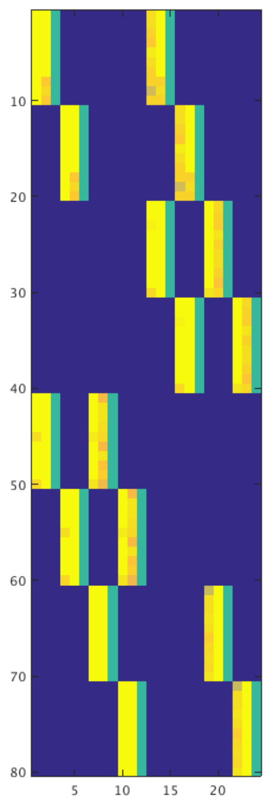
$$\begin{bmatrix} {}^{1,2}P & -\,{}^{1,2}Q & 0 \\ {}^{1,3}P & 0 & -\,{}^{1,3}Q \end{bmatrix} \times \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

and to 4 tiles
(with overlaps 1-2, 1-3, 2-4, 3-4, but not 1-4, 2-3)

$$\begin{bmatrix} {}^{1,2}P & -\,{}^{1,2}Q & 0 & 0 \\ {}^{1,3}P & 0 & -\,{}^{1,3}Q & 0 \\ 0 & {}^{2,4}P & 0 & -\,{}^{2,4}Q \\ 0 & 0 & {}^{3,4}P & -\,{}^{3,4}Q \end{bmatrix} \times \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Let's cut the point-matches per tile pair down to 10, so it is easier to see this same structure in the example. Below is structure plot and value-coded, to see the repeating structure.





based on the looks of this, the row ordering is a little different than the example above.

It's fine, because we're just re-ordering zeros on the right-hand side.
Instead, we are seeing:

$$\begin{bmatrix} {}^{1,3}P & 0 & -\,{}^{1,3}Q & 0 \\ 0 & 0 & {}^{3,4}P & -\,{}^{3,4}Q \\ {}^{1,2}P & -\,{}^{1,2}Q & 0 & 0 \\ 0 & {}^{2,4}P & 0 & -\,{}^{2,4}Q \end{bmatrix} \times \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$
$$(Ax = b) \tag{7}$$

The transforms we are solving for:

$$\begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix}$$

represent 24 unknowns.
The matrix has 80 rows.

The system is overdetermined.
We want a solution for x which minimizes the norm of the residuals:

$$S(x) = ||b - Ax||^2 \tag{8}$$

There are a few forms of this (see wikipedia: normal equations), but, I think this one makes it clear.
A is mxn

$$r_i = b_i - \sum_{j=1}^{n} A_{ij} x_j \tag{9}$$

$$S = \sum_{i=1}^{m} r_i^2 \tag{10}$$

minimize with respect to x to find the best fit:

$$\frac{\partial S}{\partial x_j} = 2 \sum_{i}^{m} r_i \frac{\partial r_i}{x_j} \tag{11}$$

$$\frac{\partial r_i}{x_j} = -A_{ij} \tag{12}$$

$$\frac{\partial S}{\partial x_j} = 2\sum_{i}^{m}\left(b_i - \sum_{k=1}^{n} A_{ik}x_k\right)(-A_{ij}) = 0 \tag{13}$$

rearranging:

$$\sum_{i=1}^{m}\sum_{k=1}^{n} A_{ij}A_{ik}x_k = \sum_{i=1}^{m} A_{ij}b_i \tag{14}$$

which is:

$$A^T Ax = A^T b \tag{15}$$

$A^T A$ is nxn, so, now we have n equations and n unknowns. The solver includes a way to weight points differently. In this example, this means including an 80x80 diagonal matrix, W:

$$A^T W Ax = A^T W b \tag{16}$$

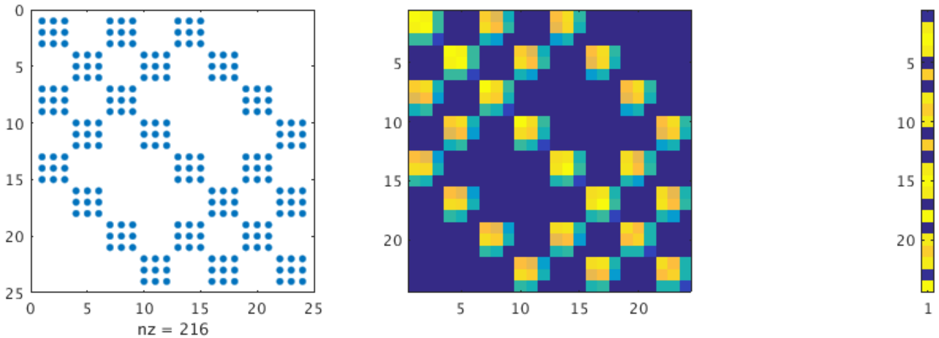
For equal weighting, $W = I$.
The solver also includes regularization:

$$A^T W A x + \lambda = A^T W b + \lambda d \tag{17}$$

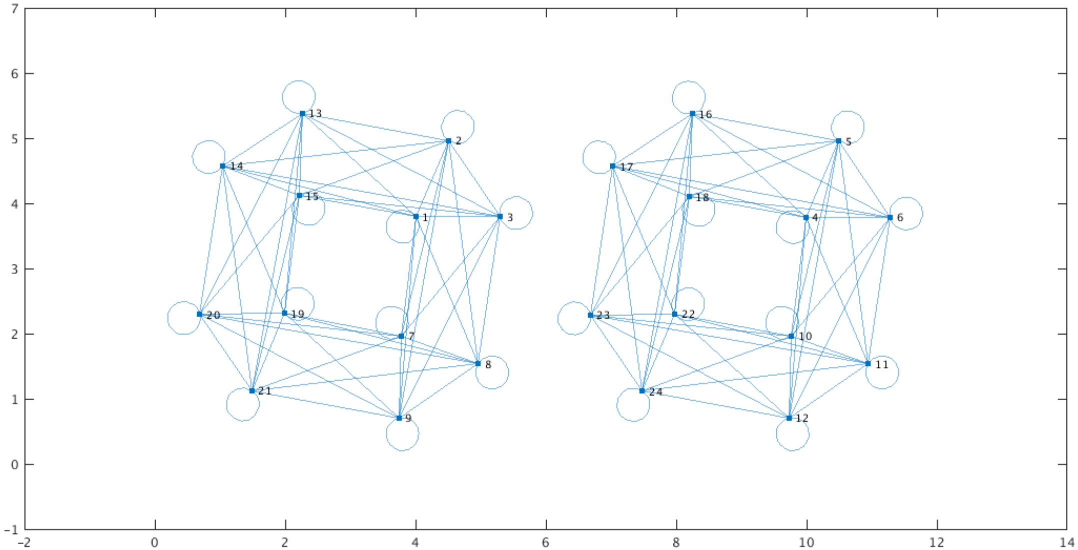
or

$$K x = L_m \tag{18}$$

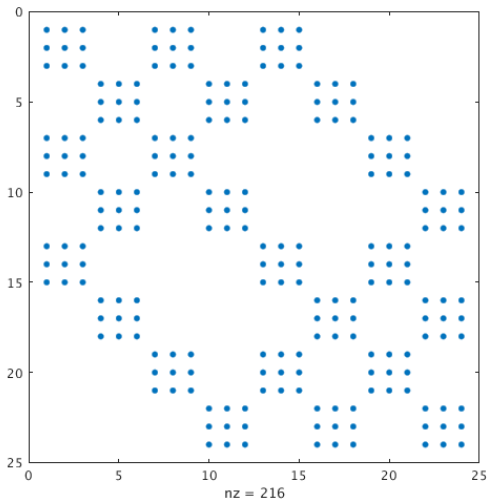
where K is square, and L_m has non-zero entries:

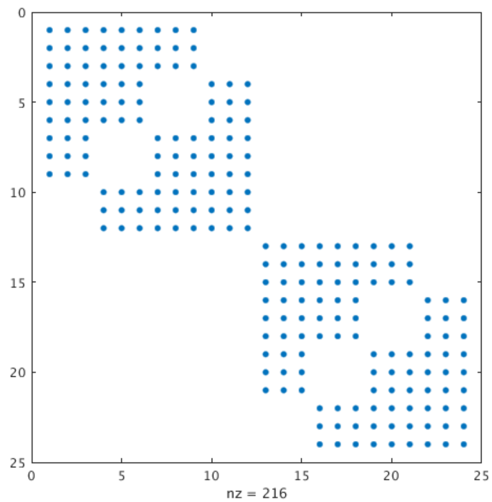


Looking at the graph of K, it shows two disconnected graphs.



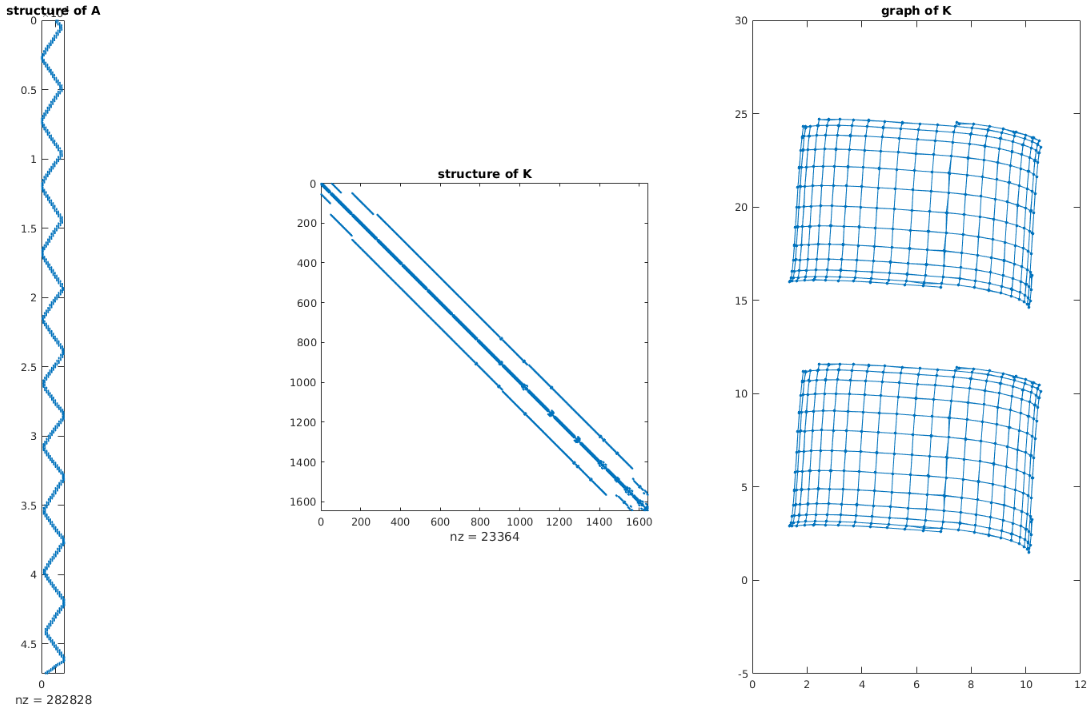
We can use this to permute the rows and columns, to get:



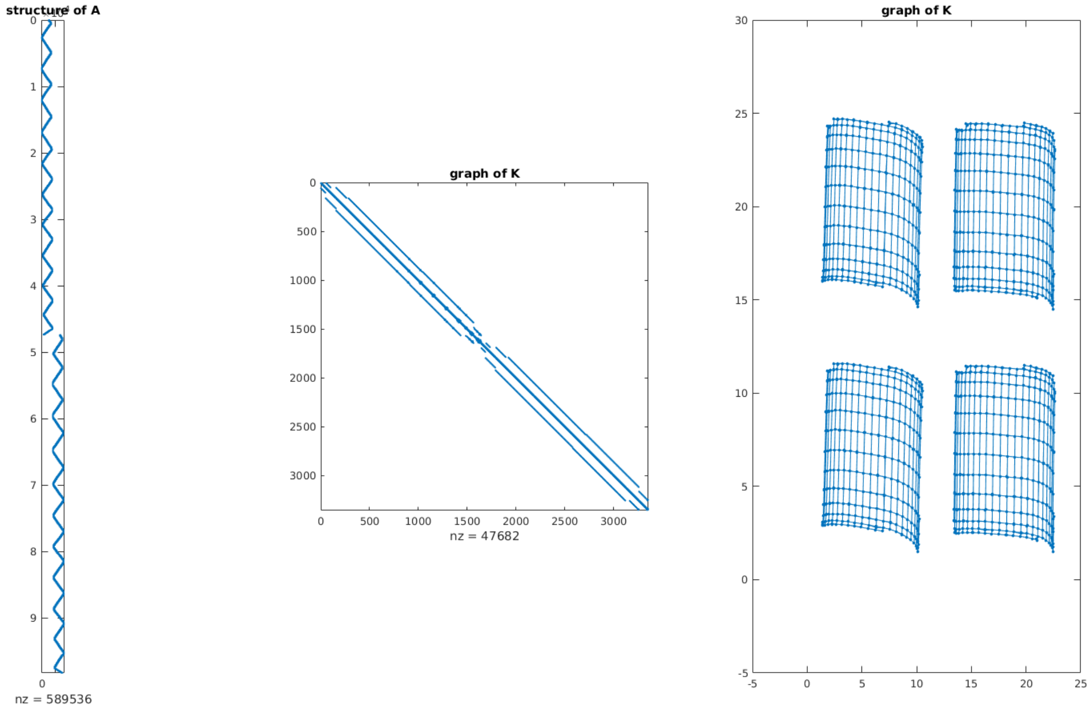


This permuted structure plot implies an obvious way to partition the matrix for distributed solving.

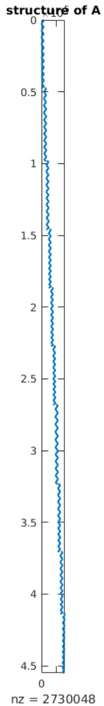
Let’s go bigger. A whole section (282 tiles):

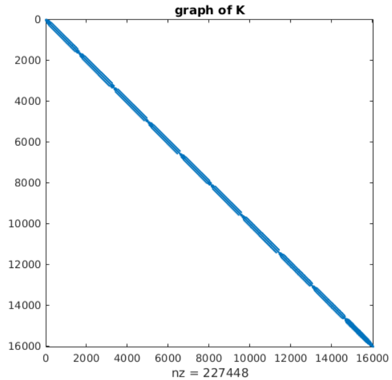


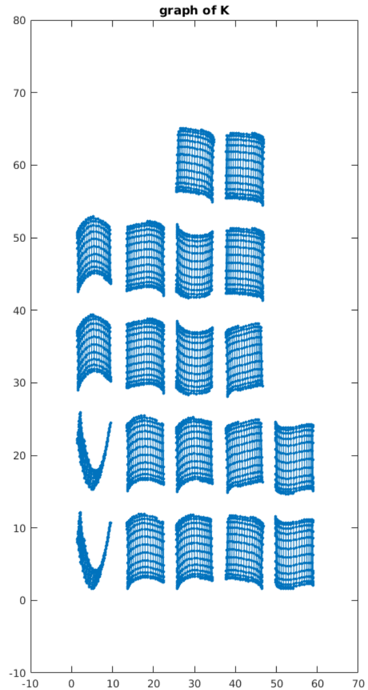
now, let’s add a second section:



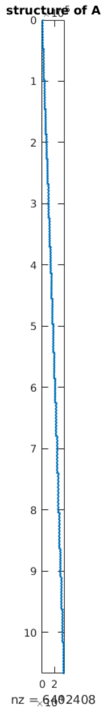
now, 10 sections:

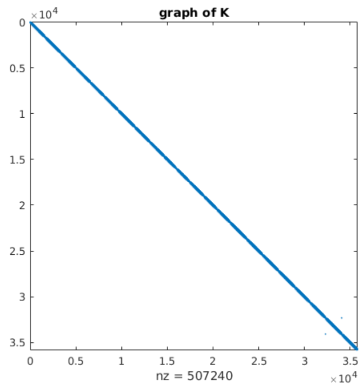


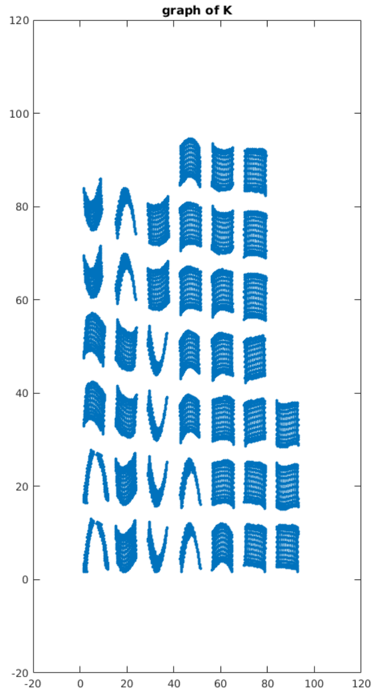




22 sections from EM_Phase1_Fine is the most I can run through right now (unknown code crash):


structure of A
nz = 6402408


graph of K
nz = 507240
×10^4


graph of K