Recall Newton’s Method for Systems:

Suppose $F$ is function from $\mathbb{R}^n$ to $\mathbb{R}^n$:

$$F : \mathbb{R}^n \rightarrow \mathbb{R}^n$$

and we want to solve

$$F(x) = 0 \quad (1)$$

Here

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix},$$

and

$$F(x) = \begin{bmatrix} F_1(x) \\ F_2(x) \\ \vdots \\ F_n(x) \end{bmatrix},$$

Also let

$$\nabla F = \left[ \frac{\partial F_i}{\partial x_j} \right]_{i,j=1,2,...,n}$$

is the Jacobian of $F$. 
• Newton’s Method is given by

\[ x^{(0)} \text{ given} \]

• For \( k = 0, 1, 2, \ldots \) do:
  1. Compute \( F(x^{(k)}) \) and \( A = \nabla F(x^{(k)}) \)
  2. Solve
      \[ As = -F(x^{(k)}) \]
      (The vector \( s \) is called the **Newton Step**).
  3. Let
      \[ x^{(k+1)} = x^{(k)} + s \]
  4. Repeat until
      \[ \|s\| < 0.01 \]
      (We stop when the length of the Newton step is less than 1 cm.)

• Newton’s Method is often written as

\[ x^{(0)} \text{ given } x^{(k+1)} = x^{(k)} - \left( \nabla F(x^{(k)}) \right)^{-1} F(x^{(k)}) \]

where \( k = 0, 1, 2, \ldots \). However, there is no need to compute the inverse of the Jacobian.
• Example:

\[
F_1(x, y) = x^2 + y^2 - 1 = 0 \\
F_2(x, y) = y - x^4 = 0 , \quad x_0 = y_0 = 1.
\]

• slight change in notation ...

\[
x^{(0)} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}
\]

\[
\nabla F = \begin{bmatrix} 2x & 2y \\ -4x^3 & 1 \end{bmatrix}
\]

\[
\nabla F(1, 1) = \begin{bmatrix} 2 \\ -4 \end{bmatrix}
\]

\[
F(1, 1) = \begin{bmatrix} 2 \\ 1 \end{bmatrix}
\]

\[
x^{(1)} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} - \begin{bmatrix} 2 \\ -4 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix}
\]

exercise iterate:
For the receiver program in the term project we have equations like

\[ z = \|x_V - x_S\| - c(t_V - t_S) = 0 \]

where
- \( x_V \) is the unknown position of the vehicle
- \( x_S \) is the known position of the satellite
- \( t_V \) is the unknown time at which the vehicle receives the signals, and
- \( t_S \) is the known time at which the satellite broadcasts.

Let’s say

\[
x_V = \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{bmatrix} \quad \text{and} \quad x_S = \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \end{bmatrix}.
\]

Then

\[
z = \left( \sum_{i=1}^{3} (\xi_i - \sigma_i)^2 \right)^{\frac{1}{2}} - c(t_V - t_S),
\]

and, for example,

\[
\frac{\partial z}{\partial \xi_i} = \frac{2(\xi_i - \sigma_i)}{2 \left( \sum_{i=1}^{3} (\xi_i - \sigma_i)^2 \right)^{\frac{1}{2}}} = \frac{(\xi_i - \sigma_i)}{\left( \sum_{i=1}^{3} (\xi_i - \sigma_i)^2 \right)^{\frac{1}{2}}}.
\]
We already discussed this briefly, but it is worth reiterating: $t_V$ and $x_V$ have very different accuracy requirements. We want to know $x_V$ within one centimeter and $t_V$ within one centimeter divided by the speed of light. This

- This is bound to lead to numerical trouble.

- However, $t_V$ enters our equations linearly, and can be eliminated!

- We simply take differences. For example, suppose we have data from two satellites, $S_1$ and $S_2$:

$$
\begin{align*}
    z_1 &= \|x_V - x_{S_1}\| - c (t_V - t_{S_1}) = 0 \\
    z_2 &= \|x_V - x_{S_2}\| - c (t_V - t_{S_2}) = 0
\end{align*}
$$

- Subtracting the second equation from the first gives the equation

$$
\|x_V - x_{S_1}\| - \|x_V - x_{S_2}\| + c (t_{S_1} - t_{S_2}) = 0
$$

which no longer contains the variable $t_V$.

- So, for example, if we have data from four satellites, $S_1$, $S_2$, $S_3$, and $S_4$, say, we can form 3 such equations corresponding to

$$
S_1 - S_2, \quad S_1 - S_3, \quad \text{and} \quad S_1 - S_4
$$

and solve the resulting system of 3 equations in 3 unknowns.

- But we have data from more than four satellites, and we want to (and should) use all!

- So we get an **overdetermined** system.

- We have more equations than unknowns.

- To introduce the relevant idea, **discrete nonlinear Least Squares**, consider and example:

$$
\begin{align*}
    F_1(x, y) &= x + y - 2 = 0 \\
    F_2(x, y) &= x^2 + y^2 - 2 = 0 \\
    F_3(x, y) &= xy - 2 = 0
\end{align*}
$$
• We have 3 equations in 2 unknowns, there is no solution (check it out!).

• However, if there was in fact a solution we’d have

\[ f(x, y) = \sum_{i=1}^{3} F_i^2(x, y) = F_1^2(x, y) + F_2^2(x, y) + F_3^2(x, y) = 0. \]

• Since we can’t have that we do the next best thing: Find \( x \) and \( y \) so as to minimize \( f \):

\[ f(x, y) = \sum F_i^2(x, y) = \min. \]

• This idea generalizes in an obvious way to a system of \( m \) equations in \( n \) variables, where \( m > n \):

• Suppose we have a function

\[ F : \mathbb{R}^n \rightarrow \mathbb{R}^m \]

where \( m > n \). Instead of solving the root finding problem

\[ F(x) = 0 \]

which has no solution we solve the nonlinear Least Squares problem

\[ f(x) = \|F(x)\|^2 = (F(x))^T F(x) = \min \] (2)

\[ F \] is called the \textbf{residual} in this context. Instead of making the residual zero we make it as small as possible.

• To find a solution of (2) we find a stationary point where \( \nabla f(x) = 0 \).

• We can and should apply these ideas to the term project. Suppose we have data from \( m + 1 \) satellites \( S_0, \ldots, S_m \). Then we find \( x_V \) so as to minimize

\[ f(x_V) = \sum_{i=1}^{m} (\|x_V - x_{S_0}\| - \|x_V - x_{S_i}\| + c(t_{S_0} - t_{S_i}))^2 = \min. \]
To solve this problem we need to solve the nonlinear $3 \times 3$ system

$$\nabla f = 0.$$  

This will give rise to a sequence of $3 \times 3$ positive definite linear systems.

HW 1 asks you to describe Newton’s Method. This means you need to give explicit formulas for the relevant derivatives!
• Let’s look at this from a slightly more general point of view, with an important conclusion at the end.

• Suppose

$$F : \mathbb{R}^n \rightarrow \mathbb{R}^m \quad \text{where} \quad m > n.$$ 

• We want $F(x) = 0$ but the system is overdetermined and so we settle for solving

$$f(x) = \|F(x)\|^2 = F(x)^T F(x) = \min.$$ 

• To find the minimizer we solve the nonlinear system

$$\nabla f(x) = 2 \nabla F(x)^T F(x) = 0$$

where $\nabla f$ is the gradient of $f$ and the Jacobian $\nabla F(x)$ is given by

$$B = \nabla F(x) = \left[ \frac{\partial F_i}{\partial x_j} \right]_{i=1,\ldots,m, \quad j=1,\ldots,n}.$$ 

• To solve the nonlinear system we need the Jacobian of $\nabla f$. That’s the Hessian $H(x)$ (the symmetric matrix of second order partial derivatives) of $f$.

• In our case we get

$$H(x) = \nabla (\nabla f) = \nabla^2 f = \nabla (2 \nabla F^T F)$$

$$= 2 \left( (\nabla^2 F)^T F + \nabla F^T \nabla F \right)$$

• Here, as before, $B = \nabla F$ is an $m \times n$ matrix. $\nabla^2 F$ is an $m \times n \times n$ tensor. It consists of $n \times n$ layers, each of which is the matrix of second order partial derivatives of a component of $F$.

However, the residual $F$ is very small in our project!

• So we can safely ignore the tensor term and approximate

$$H(x) \approx 2 \nabla F^T \nabla F = B^T B$$

• It’s easy to see that $B^T B$ is positive definite.