Getting back to GN method, why do we take steps
\[ \mathbf{s}_k = -D\mathbf{R}_k + \mathbf{R}_k \]  
(*)
when \( D\mathbf{R}_k \) is not full rank?

First order necessary optimality condition:
\[ \mathbf{x}^* \text{ minimizer of } \frac{1}{2} \| \mathbf{R}(\mathbf{x}) \|_2^2 = f(\mathbf{x}) \Rightarrow Df(\mathbf{x}^*) = D\mathbf{R}(\mathbf{x}^*) R(\mathbf{x}^*) = 0 \]

Thus if our current iterate happens to be \( \mathbf{x}^* \),
\[ \mathbf{s}^* = -D\mathbf{R}^* + \mathbf{R}^* = - (D\mathbf{R} + D\mathbf{R}^*)^T D\mathbf{R}^* \]

so we will not move from optimum if we find it.
(regardless of having small/large residual \( \mathbf{R}^* \))

Yet another way of seeing this:
\[ \mathbf{s}^* \text{ is the minimal norm sol to } \min_{\mathbf{s}} \frac{1}{2} \| \mathbf{R}^* + D\mathbf{R} \|_2^2 \]
\[ \Rightarrow D\mathbf{R}^* D\mathbf{R} s^* = -D\mathbf{R}^* \mathbf{R}^* s^* = 0 \]
And the smallest such \( s \) so clearly \( s = 0 \).

Large residual problem

Gauss-Newton relies on approximating the Hessian by parts involving
only first derivative information:
\[ f(\mathbf{x}) = \frac{1}{2} \| \mathbf{R}(\mathbf{x}) \|_2^2 \text{ then } D^2 f(\mathbf{x}) = D\mathbf{R}(\mathbf{x}) D\mathbf{R}(\mathbf{x}) + \sum_{i=1}^m R_i(\mathbf{x})^T R_i(\mathbf{x}), \]
\[ \approx D\mathbf{R}(\mathbf{x}) D\mathbf{R}(\mathbf{x}). \]

This approximation is not very good if \( \| \mathbf{R}(\mathbf{x}) \|_2 \) is large.

There are second type methods (e.g. BF65) that capture this missing
second order information. (pp 262-265)
§ 1.1 Non-linear Equations

F: \( \mathbb{R}^n \rightarrow \mathbb{R}^n \), want to find \( x^* \) s.t.

\[ F(x^*) = 0 \]

\( x^* \) is a root or zero of \( F \).

**Newton's method:**

\[ F(x_{k+1}) = F(x_k) + DF(x_k) s \quad \text{(lm model)} \]

where \( DF(x_k) = \text{Jacobian of } F \text{ at } x_k \)

\[ DF(x_k) = \begin{bmatrix}
    \frac{\partial F_1(x_k)}{\partial x_1} \\
    \frac{\partial F_2(x_k)}{\partial x_2} \\
    \vdots \\
    \frac{\partial F_n(x_k)}{\partial x_n}
\end{bmatrix} \]

Compute \( s \) for which \( m(x_{k+1}) = 0 \) (we want a root!)

\[ \Rightarrow DF_k s = -F_k \quad (2) \]

**Newton's method for NL eqs:** \( x_{k+1} = x_k + s_k \), where \( s_k \) solves (2)

**Note that:**

\[ T(x) = F(x_k) + F'(x_k)(x - x_k) \] is tangent to \( F(x) \) at \( x_k \),

so by solving (2) we are finding root of tangent: for 1D function,
Connection with unconstrained optimization

necessary optimality condition:

\[
\frac{\nabla f(x)}{F(x)} = \nabla^2 f(x) \quad \text{(guaranteed to be symmetric, but in general } \nabla^2 f(x) \text{ needs not be symmetric)}
\]

New Linearizes as an optimization problem

\[F(x) = 0 \quad \Rightarrow \quad \text{find } x \text{ that solves } \min_{x} \frac{1}{2} \| F(x) \|^2\]

If we solve this problem using Gauss–Newton we find step as solution to:

\[
\min_{x} \frac{1}{2} \| Df_k + F_k \|^2 \quad \text{then assuming } Df_k \in \mathbb{R}^{n \times n}, \text{ non-singular}
\]

\[
\Rightarrow \quad \Delta x = -(Df_k^T Df_k)^{-1} Df_k^T F_k
\]

\[
\quad = -Df_k^{-1} F_k \quad \text{which is equivalent to what we derived before.}
\]

Convergence result for Newton's method for non-linear equations:

Let \( F: \mathbb{R}^n \rightarrow \mathbb{R}^n \) be continuously differentiable with \( Df(x) \) Lipschitz continuous.

Let \( x^* \) be a point at which \( Df(x^*) \) is invertible.

Then for \( x_0 \) close enough to \( x^* \),

\[
\| x_{k+1} - x^* \| \leq C \| x_k - x^* \|^2 \quad \text{(Q-quadratic convergence)}
\]

Proof: Banach fixed pt theorem + Banach lemma

It is possible to come up with quasi-Newton methods that do not require the computation of \( Df(x) \). Their derivation is similar to that of DFP, BFGS, and the most popular of such methods is “Broyden's method” (p. 278–283) which can achieve superlinear convergence.
Broyden's method:

choose initial point $x_0$.
choose initial approx $B_0$ to $DR(x_0)$ (usually $I$)

$B_0$ non-singular

for $k = 0, 1, \ldots$

Solve $B_k p_k = -R(x_k)$
choose step size $\lambda_k$ (line search)

$x_{k+1} = x_k + \lambda_k p_k$

$s_k = x_{k+1} - x_k$

$y_k = R(x_{k+1}) - R(x_k)$

$B_{k+1} = B_k + \frac{(y_k - B_k s_k) s_k^T}{s_k^T s_k}$

Can also do update to approx to $DR(x_k)^{-1}$ (as BFGS does)

Update so it satisfies second equation:

$y_k = B_{k+1} s_k$

and $B_{k+1}$ solves

$$\min_{B \ s.t. \ y_k = B s_k} \| B - B_k \|$$

Proof:

$$\| B_{k+1} - B_k \| = \left\| \frac{(y_k - B_k s_k) s_k^T}{s_k^T s_k} \right\|$$

$$= \left\| \frac{(B - B_k) s_k s_k^T}{s_k^T s_k} \right\| \leq \| B - B_k \| \frac{\| s_k \|}{s_k^T s_k} = \| B - B_k \|$$