Non-linear problems in n variable

Lectures for PHD course on
Non-linear equations and numerical optimization

Enrico Bertolazzi

DIMS – Università di Trento

March 2005

Outline

The problem to solve

Problem

Given $\mathbf{F} : D \subseteq \mathbb{R}^n \mapsto \mathbb{R}^n$
Find $\boldsymbol{x}_\star \in D$ *for which* $\mathbf{F}(\boldsymbol{x}_\star) = 0$.

Example

Let

$$\mathbf{F}(\boldsymbol{x}) = \begin{pmatrix} x_1^2 + x_2^3 + 7 \\ x_1 + x_2 + 1 \end{pmatrix}$$

which has $\mathbf{F}(\boldsymbol{x}_\star) = \mathbf{0}$ for $\boldsymbol{x}_\star = (1, -2)^T$.

Outline

The Newton procedure (1/3)

- Consider the following map

$$\mathbf{F}(\boldsymbol{x}) = \begin{pmatrix} x_1^2 + x_2^3 + 7 \\ x_1 + x_2 + 1 \end{pmatrix}$$

we known an approximation of a root $\boldsymbol{x}_0 \approx (1.1, -1.9)^T$.

- Setting $\boldsymbol{x}_1 = \boldsymbol{x}_0 + \boldsymbol{p}$ we obtain [1]

$$\mathbf{F}(\boldsymbol{x}_0 + \boldsymbol{p}) = \begin{pmatrix} 1.351 \\ 0.2 \end{pmatrix} + \begin{pmatrix} 2.2 & 10.83 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} + \vec{\mathcal{O}}(\|\boldsymbol{p}\|^2)$$

if $\boldsymbol{x}_0$ is a good approximation of a root of $\mathbf{F}(\boldsymbol{x})$ then $\vec{\mathcal{O}}(\|\boldsymbol{p}\|^2)$ is a small vector.

[1] Here $\vec{\mathcal{O}}(x)$ means $(\mathcal{O}(x), \ldots, \mathcal{O}(x))^T$

The Newton procedure (2/3)

- Neglecting $\vec{\mathcal{O}}(\|\boldsymbol{p}\|^2)$ and solving

$$\begin{pmatrix} 1.351 \\ 0.2 \end{pmatrix} + \begin{pmatrix} 2.2 & 10.83 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} = \mathbf{0}$$

we obtain $\boldsymbol{p} = (-0.094438, -0.105562)^T$.

- Now we set

$$\boldsymbol{x}_1 = \boldsymbol{x}_0 + \boldsymbol{p} = \begin{pmatrix} 1.005562 \\ -2.0055612 \end{pmatrix}$$

The Newton procedure (3/3)

- Considering

$$\mathbf{F}(\boldsymbol{x}_1 + \boldsymbol{q}) = \begin{pmatrix} -0.05576 \\ 8\,10^{-7} \end{pmatrix} + \begin{pmatrix} 2.0111 & 12.0668 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} + \vec{\mathcal{O}}(\|\boldsymbol{q}\|^2)$$

- Neglecting $\vec{\mathcal{O}}(\|\boldsymbol{q}\|^2)$ and solving

$$\begin{pmatrix} -0.05576 \\ 8\,10^{-7} \end{pmatrix} + \begin{pmatrix} 2.0111 & 12.0668 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} = \mathbf{0}$$

  we obtain $\boldsymbol{q} = (-0.0055466, 0.0055458)^T$.
- Now we set $\boldsymbol{x}_2 = \boldsymbol{x}_1 + \boldsymbol{q} = (1.000015, -2.000015)^T$

The Newton procedure: a modern point of view (1/2)

The previous procedure can be resumed as follows:

1. Consider the following function $\mathbf{F}(\boldsymbol{x})$. We known an approximation of a root $\boldsymbol{x}_0$.
2. Expand by Taylor series

$$\mathbf{F}(\boldsymbol{x}) = \mathbf{F}(\boldsymbol{x}_0) + \nabla\mathbf{F}(\boldsymbol{x}_0)(\boldsymbol{x} - \boldsymbol{x}_0) + \vec{\mathcal{O}}(\|\boldsymbol{x} - \boldsymbol{x}_0\|^2)$$

3. Drop the term $\vec{\mathcal{O}}(\|x - \boldsymbol{x}_0\|^2)$ and solve

$$\mathbf{0} = \mathbf{F}(\boldsymbol{x}_0) + \nabla\mathbf{F}(\boldsymbol{x}_0)(\boldsymbol{x} - \boldsymbol{x}_0)$$

   Call $\boldsymbol{x}_1$ this solution.
4. Repeat $1 - 3$ with $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, $\boldsymbol{x}_3$, ...

The Newton procedure: a modern point of view (2/2)

Algorithm (Newton iterative scheme)

Let $\boldsymbol{x}_0$ assigned, then for $k = 0, 1, 2, \ldots$

1. *Solve for $\boldsymbol{p}_k$:*

$$\nabla \mathbf{F}(\boldsymbol{x}_k)\boldsymbol{p}_k + \mathbf{F}(\boldsymbol{x}_k) = \mathbf{0}$$

2. *Update*

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{p}_k$$

Standard Assumptions

In the study of convergence of numerical scheme, some standard regularity assumption are assumed for the function $\mathbf{F}(\boldsymbol{x})$.

Assumption (Standard Assumptions)

The function $\mathbf{F} : D \subset \mathbb{R}^n \mapsto \mathbb{R}^n$ is continuous, differentiable with Lipschitz derivative $\nabla \mathbf{F}(\boldsymbol{x})$. i.e.

$$\|\nabla \mathbf{F}(\boldsymbol{x}) - \nabla \mathbf{F}(\boldsymbol{y})\| \leq \gamma \|\boldsymbol{x} - \boldsymbol{y}\| \qquad \forall \boldsymbol{x}, \boldsymbol{y} \in D \subset \mathbb{R}^n$$

Lemma (Taylor like expansion)

Let $\mathbf{F}(\boldsymbol{x})$ satisfy the standard assumptions, then

$$\|\mathbf{F}(\boldsymbol{y}) - \mathbf{F}(\boldsymbol{x}) - \nabla \mathbf{F}(\boldsymbol{x})(\boldsymbol{y} - \boldsymbol{x})\| \leq \frac{\gamma}{2} \|\boldsymbol{x} - \boldsymbol{y}\|^2 \quad \forall \boldsymbol{x}, \boldsymbol{y} \in D \subset \mathbb{R}^n$$

Proof.

From basic Calculus:

$$\mathbf{F}(\boldsymbol{y}) - \mathbf{F}(\boldsymbol{x}) = \int_0^1 \nabla\mathbf{F}(\boldsymbol{x} + t(\boldsymbol{y} - \boldsymbol{x}))(\boldsymbol{y} - \boldsymbol{x})\, dt$$

subtracting on both side $\nabla\mathbf{F}(\boldsymbol{x})(\boldsymbol{y} - \boldsymbol{x})$ we have

$$\mathbf{F}(\boldsymbol{y}) - \mathbf{F}(\boldsymbol{x}) - \nabla\mathbf{F}(\boldsymbol{x})(\boldsymbol{y} - \boldsymbol{x}) = \int_0^1 \left[\nabla\mathbf{F}(\boldsymbol{x} + t(\boldsymbol{y} - \boldsymbol{x})) - \nabla\mathbf{F}(\boldsymbol{x})\right](\boldsymbol{y} - \boldsymbol{x})\, dt$$

and taking the norm

$$\left\|\mathbf{F}(\boldsymbol{y}) - \mathbf{F}(\boldsymbol{x}) - \nabla\mathbf{F}(\boldsymbol{x})(\boldsymbol{y} - \boldsymbol{x})\right\| \leq \int_0^1 \gamma t \left\|\boldsymbol{y} - \boldsymbol{x}\right\|^2 \, dt$$

□

Lemma (Jacobian norm control)

Let $\mathbf{F}(\boldsymbol{x})$ *satisfying standard assumptions, and* $\nabla\mathbf{F}(\boldsymbol{x}_\star)$ *non singular. Then there exists* $\delta > 0$ *such that for all* $\left\|\boldsymbol{x} - \boldsymbol{x}_\star\right\| \leq \delta$ *we have*

$$2^{-1}\left\|\nabla\mathbf{F}(\boldsymbol{x})\right\| \leq \left\|\nabla\mathbf{F}(\boldsymbol{x}_\star)\right\| \leq 2\left\|\nabla\mathbf{F}(\boldsymbol{x})\right\|$$

and

$$2^{-1}\left\|\nabla\mathbf{F}(\boldsymbol{x})^{-1}\right\| \leq \left\|\nabla\mathbf{F}(\boldsymbol{x}_\star)^{-1}\right\| \leq 2\left\|\nabla\mathbf{F}(\boldsymbol{x})^{-1}\right\|$$

Proof. (1/3).

From standard assumptions choosing $\gamma\delta \leq 2^{-1}\|\nabla \mathbf{F}(\boldsymbol{x}_\star)\|$

$$\begin{aligned}\|\nabla \mathbf{F}(\boldsymbol{x})\| &\leq \|\nabla \mathbf{F}(\boldsymbol{x}) - \nabla \mathbf{F}(\boldsymbol{x}_\star)\| + \|\nabla \mathbf{F}(\boldsymbol{x}_\star)\| \\ &\leq \gamma\|\boldsymbol{x} - \boldsymbol{x}_\star\| + \|\nabla \mathbf{F}(\boldsymbol{x}_\star)\| \\ &\leq (3/2)\|\nabla \mathbf{F}(\boldsymbol{x}_\star)\| \leq 2\|\nabla \mathbf{F}(\boldsymbol{x}_\star)\|\end{aligned}$$

again choosing $\gamma\delta \leq 2^{-1}\|\nabla \mathbf{F}(\boldsymbol{x}_\star)\|$

$$\begin{aligned}\|\nabla \mathbf{F}(\boldsymbol{x}_\star)\| &\leq \|\nabla \mathbf{F}(\boldsymbol{x}_\star) - \nabla \mathbf{F}(\boldsymbol{x})\| + \|\nabla \mathbf{F}(\boldsymbol{x})\| \\ &\leq \gamma\|\boldsymbol{x} - \boldsymbol{x}_\star\| + \|\nabla \mathbf{F}(\boldsymbol{x})\| \\ &\leq 2^{-1}\|\nabla \mathbf{F}(\boldsymbol{x}_\star)\| + \|\nabla \mathbf{F}(\boldsymbol{x})\|\end{aligned}$$

so that $2^{-1}\|\nabla \mathbf{F}(\boldsymbol{x}_\star)\| \leq \|\nabla \mathbf{F}(\boldsymbol{x})\|$.

The Newton Raphson — Standard Assumptions

Proof. (2/3).

From the continuity of the determinant there exists a neighbor with $\nabla \mathbf{F}(\boldsymbol{x})$ non singular for all $\|\boldsymbol{x} - \boldsymbol{x}_\star\| \leq \delta$.

$$\begin{aligned}&\left\|\nabla \mathbf{F}(\boldsymbol{x})^{-1} - \nabla \mathbf{F}(\boldsymbol{x}_\star)^{-1}\right\| \\ &\qquad \leq \left\|\nabla \mathbf{F}(\boldsymbol{x})^{-1}\right\|\|\nabla \mathbf{F}(\boldsymbol{x}_\star) - \nabla \mathbf{F}(\boldsymbol{x})\|\left\|\nabla \mathbf{F}(\boldsymbol{x}_\star)^{-1}\right\| \\ &\qquad \leq \gamma\|\boldsymbol{x} - \boldsymbol{x}_\star\|\left\|\nabla \mathbf{F}(\boldsymbol{x})^{-1}\right\|\left\|\nabla \mathbf{F}(\boldsymbol{x}_\star)^{-1}\right\|\end{aligned}$$

and choosing δ such that $\gamma\delta\left\|\nabla \mathbf{F}(\boldsymbol{x}_\star)^{-1}\right\| \leq 2^{-1}$ we have

$$\left\|\nabla \mathbf{F}(\boldsymbol{x})^{-1} - \nabla \mathbf{F}(\boldsymbol{x}_\star)^{-1}\right\| \leq 2^{-1}\left\|\nabla \mathbf{F}(\boldsymbol{x})^{-1}\right\|$$

and using this last inequality

$$\begin{aligned}\left\|\nabla \mathbf{F}(\boldsymbol{x}_\star)^{-1}\right\| &\leq \left\|\nabla \mathbf{F}(\boldsymbol{x}_\star)^{-1} - \nabla \mathbf{F}(\boldsymbol{x})^{-1}\right\| + \left\|\nabla \mathbf{F}(\boldsymbol{x})^{-1}\right\| \\ &\leq (3/2)\left\|\nabla \mathbf{F}(\boldsymbol{x})^{-1}\right\| \leq 2\left\|\nabla \mathbf{F}(\boldsymbol{x})^{-1}\right\|\end{aligned}$$

Proof. (3/3).

Using last inequality again

$$\left\|\nabla \mathbf{F}(\boldsymbol{x})^{-1}\right\| \leq \left\|\nabla \mathbf{F}(\boldsymbol{x})^{-1} - \nabla \mathbf{F}(\boldsymbol{x}_\star)^{-1}\right\| + \left\|\nabla \mathbf{F}(\boldsymbol{x}_\star)^{-1}\right\|$$
$$\leq 2^{-1}\left\|\nabla \mathbf{F}(\boldsymbol{x})^{-1}\right\| + \left\|\nabla \mathbf{F}(\boldsymbol{x}_\star)^{-1}\right\|$$

so that

$$2^{-1}\left\|\nabla \mathbf{F}(\boldsymbol{x})^{-1}\right\| \leq \left\|\nabla \mathbf{F}(\boldsymbol{x}_\star)^{-1}\right\|$$

choosing δ such that for all $\|\boldsymbol{x} - \boldsymbol{x}_\star\| \leq \delta$ we have $\nabla \mathbf{F}(\boldsymbol{x})$ non singular and $\gamma\delta \leq 2^{-1}\|\nabla \mathbf{F}(\boldsymbol{x}_\star)\|$ and $\gamma\delta\left\|\nabla \mathbf{F}(\boldsymbol{x}_\star)^{-1}\right\| \leq 2^{-1}$ then the inequality of the lemma are true. □

Theorem (Local Convergence of Newton method)

Let $\mathbf{F}(\boldsymbol{x})$ satisfying standard assumptions, and $\boldsymbol{x}_\star$ a simple root (i.e. $\nabla \mathbf{F}(\boldsymbol{x}_\star)$ non singular). Then, if $\|\boldsymbol{x}_0 - \boldsymbol{x}_\star\| \leq \delta$ with $C\delta \leq 1$ where

$$C = \gamma\left\|\nabla \mathbf{F}(\boldsymbol{x}_\star)^{-1}\right\|$$

then, the sequence generated by Newton method satisfies:

1. $\|\boldsymbol{x}_k - \boldsymbol{x}_\star\| \leq \delta$ *for* $k = 0, 1, 2, 3, \ldots$
2. $\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_\star\| \leq C\|\boldsymbol{x}_k - \boldsymbol{x}_\star\|^2$ *for* $k = 0, 1, 2, 3, \ldots$
3. $\lim_{k\mapsto\infty} \boldsymbol{x}_k = \boldsymbol{x}_\star$.

- The point 2 of the theorem is the second q-order of convergence of Newton method.

Proof.

Consider a Newton step with $\|\boldsymbol{x}_k - \boldsymbol{x}_\star\| \leq \delta$ and

$$\begin{aligned}\boldsymbol{x}_{k+1} - \boldsymbol{x}_\star &= \boldsymbol{x}_k - \boldsymbol{x}_\star - \nabla \mathbf{F}(\boldsymbol{x}_k)^{-1}\big[\mathbf{F}(\boldsymbol{x}_k) - \mathbf{F}(\boldsymbol{x}_\star)\big] \\ &= \nabla \mathbf{F}(\boldsymbol{x}_k)^{-1}\big[\nabla \mathbf{F}(\boldsymbol{x}_k)(\boldsymbol{x}_k - \boldsymbol{x}_\star) - \mathbf{F}(\boldsymbol{x}_k) + \mathbf{F}(\boldsymbol{x}_\star)\big]\end{aligned}$$

taking the norm and using Taylor like lemma

$$\|\boldsymbol{x}_{k+1} - \alpha\| \leq 2^{-1}\gamma \|\boldsymbol{x}_k - \alpha\|^2 \left\|\nabla \mathbf{F}(\boldsymbol{x}_k)^{-1}\right\|$$

from Jacobian norm control lemma there exist a δ such that $2\left\|\nabla \mathbf{F}(\boldsymbol{x}_k)^{-1}\right\| \geq \left\|\nabla \mathbf{F}(\boldsymbol{x}_\star)^{-1}\right\|$ for all $\|\boldsymbol{x}_k - \boldsymbol{x}_\star\| \leq \delta$. Reducing eventually δ such that $\gamma\delta\left\|\nabla \mathbf{F}(\boldsymbol{x}_\star)^{-1}\right\| \leq 1$ we have

$$\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_\star\| \leq \gamma \left\|\nabla \mathbf{F}(\boldsymbol{x}_\star)^{-1}\right\| \delta \|\boldsymbol{x}_k - \boldsymbol{x}_\star\|^2 \leq \|\boldsymbol{x}_k - \boldsymbol{x}_\star\|,$$

So that by induction we prove point 1. Point 2 and 3 follows trivially. □

- The problem of Newton method is that it converge normally only when $\boldsymbol{x}_0$ is near $\boldsymbol{x}_\star$ a root of the nonlinear system.
- A way to make a more robust non linear solver is to use the techniques developed for minimization to make a globally convergent nonlinear solver.
- In particular if we consider the merit function

$$\mathsf{f}(\boldsymbol{x}) = \frac{1}{2}\|\mathbf{F}(\boldsymbol{x})\|^2$$

  we have that $\mathsf{f}(\boldsymbol{x}) \geq 0$ and if $\boldsymbol{x}_\star$ is such that $\mathsf{f}(\boldsymbol{x}_\star) = 0$ than we have that

  1. $\boldsymbol{x}_\star$ is a global minimum of $\mathsf{f}(\boldsymbol{x})$;
  2. $\mathbf{F}(\boldsymbol{x}_\star) = \mathbf{0}$, i.e. is a solution of the nonlinear system $\mathbf{F}(\boldsymbol{x})$.
- So that finding a global minimum of the merit function $\mathsf{f}(\boldsymbol{x})$ is the same of finding a solution of the nonlinear system $\mathbf{F}(\boldsymbol{x})$.

- We can apply for example the gradient method to the merit function $\mathsf{f}(\boldsymbol{x})$. This produce a slow method.
- Instead, we can use the Newton method to produce a search direction. The resulting method is the following
  1. Compute the search direction by solving $\nabla \mathbf{F}(\boldsymbol{x}_k)\boldsymbol{d}_k + \mathbf{F}(\boldsymbol{x}_k) = \mathbf{0}$;
  2. Find an approximate solution of the problem $\alpha_k = \arg\min_{\alpha \geq 0} \|\mathbf{F}(\boldsymbol{x}_k + \alpha \boldsymbol{d}_k)\|^2$;
  3. Update the solution $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k$.
- The previous algorithm work if the direction $\boldsymbol{d}_k$ is a descent direction.

Is $\boldsymbol{d}_k$ a descent direction? (1/2)

Consider the gradient of $\mathsf{f}(\boldsymbol{x}) = (1/2)\|\mathbf{F}(\boldsymbol{x})\|^2$:

$$\frac{\partial}{\partial x_k}\mathsf{f}(\boldsymbol{x}) = \frac{1}{2}\frac{\partial}{\partial x_k}\|\mathbf{F}(\boldsymbol{x})\|^2 = \frac{1}{2}\frac{\partial}{\partial x_k}\sum_{i=1}^{n} F_i(\boldsymbol{x})^2$$

$$= \sum_{i=1}^{n} \frac{\partial F_i(\boldsymbol{x})}{\partial x_k} F_i(\boldsymbol{x})$$

this can be written as

$$\nabla \mathsf{f}(\boldsymbol{x}) = \mathbf{F}(\boldsymbol{x})^T \nabla \mathbf{F}(\boldsymbol{x})$$

Is $\boldsymbol{d}_k$ a descent direction? (2/2)

Now we check $\nabla \mathsf{f}(\boldsymbol{x}_k)\boldsymbol{d}_k$:

$$\begin{aligned}\nabla \mathsf{f}(\boldsymbol{x}_k)\boldsymbol{d}_k &= \mathbf{F}(\boldsymbol{x}_k)^T \nabla \mathbf{F}(\boldsymbol{x}_k)\boldsymbol{d}_k \\ &= -\mathbf{F}(\boldsymbol{x}_k)^T \nabla \mathbf{F}(\boldsymbol{x}_k)\nabla \mathbf{F}(\boldsymbol{x}_k)^{-1}\mathbf{F}(\boldsymbol{x}_k) \\ &= -\mathbf{F}(\boldsymbol{x}_k)^T \mathbf{F}(\boldsymbol{x}_k) \\ &= -\left\|\mathbf{F}(\boldsymbol{x}_k)\right\|^2 < 0\end{aligned}$$

so that Newton direction is a descent direction.

Is the angle from $\boldsymbol{d}_k$ and $\nabla \mathsf{f}(\boldsymbol{x}_k)$ bounded from $\pi/2$? (2/2)

Let θ_k the angle form $\nabla \mathsf{f}(\boldsymbol{x}_k)$ and $\boldsymbol{d}_k$, then we have

$$\begin{aligned}\cos\theta_k &= -\frac{\nabla \mathsf{f}(\boldsymbol{x}_k)\boldsymbol{d}_k}{\left\|\mathbf{F}(\boldsymbol{x}_k)\right\| \left\|\nabla \mathbf{F}(\boldsymbol{x}_k)^{-1}\mathbf{F}(\boldsymbol{x}_k)\right\|} \\ &= \frac{\left\|\mathbf{F}(\boldsymbol{x}_k)\right\|}{\left\|\nabla \mathbf{F}(\boldsymbol{x}_k)^{-1}\mathbf{F}(\boldsymbol{x}_k)\right\|} \\ &\geq \frac{\left\|\mathbf{F}(\boldsymbol{x}_k)\right\|}{\left\|\nabla \mathbf{F}(\boldsymbol{x}_k)^{-1}\right\| \left\|\mathbf{F}(\boldsymbol{x}_k)\right\|} \\ &\geq \left\|\nabla \mathbf{F}(\boldsymbol{x}_k)^{-1}\right\|^{-1}\end{aligned}$$

so that, if for example $\left\|\nabla \mathbf{F}(\boldsymbol{x})^{-1}\right\|$ is bounded from below then the angle θ_k is strictly less then $\pi/2$ radiants. By the Zoutendijk theorem then the globalized Newton scheme is globally convergent.

Algorithm (The globalized Newton method)

$k \leftarrow 0$; $\boldsymbol{x}$ *assigned*;
$\boldsymbol{f} \leftarrow \mathbf{F}(\boldsymbol{x})$;
while $\|\boldsymbol{f}_k\| > \epsilon$ **do**
  — *Evaluate search direction*
  Solve $\nabla\mathbf{F}(\boldsymbol{x})\boldsymbol{d} = \mathbf{F}(\boldsymbol{x})$;
  — *Evaluate dumping factor* λ
  Approximate $\lambda = \arg\min_{\alpha>0} \|\mathbf{F}(\boldsymbol{x} - \alpha\boldsymbol{d}_k)\|^2$ *by line-search*;
  — *perform step*
  $\boldsymbol{x} \leftarrow \boldsymbol{x} - \lambda\boldsymbol{d}$;
  $\boldsymbol{f} \leftarrow \mathbf{F}(\boldsymbol{x})$;
  $k \leftarrow k + 1$;
end while

Outline

The Broyden method (1/5)

- Newton method is a fast (q-order 2) numerical scheme to approximate the root of a function $\mathbf{F}(\boldsymbol{x})$ but needs the knowledge of the Jacobian $\nabla\mathbf{F}(\boldsymbol{x})$.
- Sometimes Jacobian is not available or too expensive to compute, in this case a numerical procedure to approximate the root which does not use derivative is mandatory.
- The Newton scheme find successively the root of the affine approximation

$$L_k(\boldsymbol{x}) \doteq \nabla\mathbf{F}(\boldsymbol{x}_k)(\boldsymbol{x}-\boldsymbol{x}_k)+\mathbf{F}(\boldsymbol{x}_k)=\mathbf{0}$$

- Substituting the Jacobian in the affine approximation by $\boldsymbol{A}_k$

$$M_k(\boldsymbol{x}) \doteq \boldsymbol{A}_k(\boldsymbol{x}-\boldsymbol{x}_k)+\mathbf{F}(\boldsymbol{x}_k)=\mathbf{0}$$

and solving successively this affine model produces the family of different methods:

The Broyden method (2/5)

Algorithm (Generic Secant iterative scheme)

Let $\boldsymbol{x}_0$ and $\boldsymbol{A}_0$ assigned, then for $k=0,1,2,\ldots$

1. *Solve for $\boldsymbol{p}_k$:*

$$M_k(\boldsymbol{p}_k+\boldsymbol{x}_k)=\boldsymbol{A}_k\boldsymbol{p}_k+\mathbf{F}(\boldsymbol{x}_k)=\mathbf{0}$$

2. *Update the root approximation*

$$\boldsymbol{x}_{k+1}=\boldsymbol{x}_k+\boldsymbol{p}_k$$

3. *Update the affine model and produce $\boldsymbol{A}_{k+1}$.*

The Broyden method (3/5)

1. The way an update of $M_k \to M_{k+1}$ determine the algorithm.
2. A simple update is the forcing of a number of the secant relation:
$$M_{k+1}(\boldsymbol{x}_{k+1-\ell}) = \mathbf{F}(\boldsymbol{x}_{k+1-\ell}), \qquad \ell = 1, 2, \ldots, m$$
notice that $M_{k+1}(\boldsymbol{x}_{k+1}) = \mathbf{F}(\boldsymbol{x}_{k+1})$ for all $\boldsymbol{A}_{k+1}$.
3. If $\boldsymbol{A}_{k+1} \in \mathbb{R}^{n\times n}$ and $m = n$ and $\boldsymbol{d}_\ell = \boldsymbol{x}_{k+1-\ell} - \boldsymbol{x}_{k+1}$ are linearly independent then we have enough linear relation to determine $\boldsymbol{A}_{k+1}$.
4. Unfortunately vectors $\boldsymbol{d}_\ell$ tends to become linearly dependent so that this approach is very ill conditioned.
5. A more feasible approach uses less secant relation and others conditions to determine M_{k+1}.

The Broyden method (4/5)

1. The way an update of $M_k \to M_{k+1}$ in Broyden scheme is the following:
   1. $M_{k+1}(\boldsymbol{x}_k) = \mathbf{F}(\boldsymbol{x}_k)$;
   2. $M_{k+1}(\boldsymbol{x}) - M_k(\boldsymbol{x})$ is small in some sense;
2. The first condition imply
$$\boldsymbol{A}_{k+1}(\boldsymbol{x}_k - \boldsymbol{x}_{k+1}) + \mathbf{F}(\boldsymbol{x}_{k+1}) = \mathbf{F}(\boldsymbol{x}_k)$$
which set n linear equation that do not determine the n^2 coefficients of $\boldsymbol{A}_{k+1}$.
3. The second condition become
$$M_{k+1}(\boldsymbol{x}) - M_k(\boldsymbol{x}) = (\boldsymbol{A}_{k+1} - \boldsymbol{A}_k)(\boldsymbol{x} - \boldsymbol{x}_k)$$
$$\|M_{k+1}(\boldsymbol{x}) - M_k(\boldsymbol{x})\| \leq \|\boldsymbol{A}_{k+1} - \boldsymbol{A}_k\| \, \|\boldsymbol{x} - \boldsymbol{x}_k\|$$
where $\|\cdot\|$ is some norm. The term $\|\boldsymbol{x} - \boldsymbol{x}_k\|$ is not controllable, so a condition should be $\|\boldsymbol{A}_{k+1} - \boldsymbol{A}_k\|$ is minimum.

The Broyden method (5/5)

1. Defining

$$\boldsymbol{y}_k = \mathbf{F}(\boldsymbol{x}_{k+1}) - \mathbf{F}(\boldsymbol{x}_k), \qquad \boldsymbol{s}_k = \boldsymbol{x}_{k+1} - \boldsymbol{x}_k$$

   the Broyden scheme find the update $\boldsymbol{A}_{k+1}$ which satisfy:
   1. $\boldsymbol{A}_{k+1}\boldsymbol{s}_k = \boldsymbol{y}_k$;
   2. $\|\boldsymbol{A}_{k+1} - \boldsymbol{A}_k\| \leq \|\boldsymbol{B} - \boldsymbol{A}_k\|$ for all $\boldsymbol{B}$ such that $\boldsymbol{B}\boldsymbol{s}_k = \boldsymbol{y}_k$.
2. If we choose for the norm $\|\cdot\|$ the Frobenius norm $\|\cdot\|_F$

$$\|\boldsymbol{A}\|_F = \left(\sum_{i,j=1}^{n} A_{ij}^2\right)^{1/2}$$

   then the problem admits a unique solution.

The Frobenius matrix norm (1/4)

The Frobenius norm $\|\cdot\|_F$

$$\|\boldsymbol{A}\|_F = \left(\sum_{i,j=1}^{n} A_{ij}^2\right)^{1/2}$$

is a matrix norm, i.e. it satisfy:

1. $\|\boldsymbol{A}\|_F \geq 0$ and $\|\boldsymbol{A}\|_F = 0 \iff \boldsymbol{A} = \mathbf{0}$;
2. $\|\lambda\boldsymbol{A}\|_F = |\lambda| \, \|\boldsymbol{A}\|_F$;
3. $\|\boldsymbol{A}+\boldsymbol{B}\|_F \leq \|\boldsymbol{A}\|_F + \|\boldsymbol{B}\|_F$;
4. $\|\boldsymbol{A}\boldsymbol{B}\|_F \leq \|\boldsymbol{A}\|_F \, \|\boldsymbol{B}\|_F$;

The Frobenius norm is the length of the vector $\boldsymbol{A}$ if we consider $\boldsymbol{A}$ as a vector in $\mathbb{R}^{n^2}$.

The Frobenius matrix norm (2/4)

The first two point of the Frobenius norm $\|\cdot\|_F$ are trivial, to prove point 3 and 4 we need two classical inequality:

Cauchy–Schwartz inequality

$$\sum_{i=1}^n a_i b_i \leq \left(\sum_{i=1}^n a_i^2\right)^{1/2} \left(\sum_{i=1}^n b_i^2\right)^{1/2}$$

The inequality is strict unless $a_i = \lambda b_i$ for $i = 1, 2, \ldots, n$.

Triangular inequality

$$\left(\sum_{i=1}^n (a_i + b_i)^2\right)^{1/2} \leq \left(\sum_{i=1}^n a_i^2\right)^{1/2} + \left(\sum_{i=1}^n b_i^2\right)^{1/2}$$

The inequality is strict unless $a_i = \lambda b_i$ for $i = 1, 2, \ldots, n$.

The Frobenius matrix norm (3/4)

Proof of $\|\boldsymbol{A} + \boldsymbol{B}\|_F \leq \|\boldsymbol{A}\|_F + \|\boldsymbol{B}\|_F$.
By using triangular inequality

$$\begin{aligned}
\|\boldsymbol{A} + \boldsymbol{B}\|_F &= \left(\sum_{i,j=1}^n (A_{ij} + B_{ij})^2\right)^{1/2} \\
&\leq \left(\sum_{i,j=1}^n A_{ij}^2\right)^{1/2} + \left(\sum_{i,j=1}^n B_{ij}^2\right)^{1/2} \\
&= \|\boldsymbol{A}\|_F + \|\boldsymbol{B}\|_F .
\end{aligned}$$

The Frobenius matrix norm (4/4)

Proof of $\|\boldsymbol{AB}\|_F \leq \|\boldsymbol{A}\|_F \|\boldsymbol{B}\|_F$.
By using Cauchy–Schwartz inequality with

$$
\begin{aligned}
\|\boldsymbol{AB}\|_F &= \Big(\sum_{i,j=1}^{n}\Big(\sum_{k=1}^{n} A_{ik}B_{kj}\Big)^2\Big)^{1/2} \\
&\leq \Big(\sum_{i,j=1}^{n}\Big(\sum_{k=1}^{n} A_{ik}^2\Big)\Big(\sum_{k'=1}^{n} B_{k'j}^2\Big)\Big)^{1/2} \\
&= \Big(\Big(\sum_{i=1}^{n}\sum_{k=1}^{n} A_{ik}^2\Big)\Big(\sum_{j=1}^{n}\sum_{k'=1}^{n} B_{k'j}^2\Big)\Big)^{1/2} \\
&= \|\boldsymbol{A}\|_F \|\boldsymbol{B}\|_F .
\end{aligned}
$$

With the Frobenius matrix norm it is possible to solve the following problem

Lemma

Let $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{s}, \boldsymbol{y} \in \mathbb{R}^n$ with $\boldsymbol{s} \neq \boldsymbol{0}$. Consider the set

$$
\mathcal{B} = \{\boldsymbol{B} \in \mathbb{R}^{n\times n} \mid \boldsymbol{Bs} = \boldsymbol{y}\}
$$

then there exists a unique matrix $\boldsymbol{B} \in \mathcal{B}$ such that

$$
\|\boldsymbol{A} - \boldsymbol{B}\|_F \leq \|\boldsymbol{A} - \boldsymbol{C}\|_F \qquad \text{for all } \boldsymbol{C} \in \mathcal{B}
$$

moreover $\boldsymbol{B}$ has the following form

$$
\boldsymbol{B} = \boldsymbol{A} + \frac{(\boldsymbol{y} - \boldsymbol{As})\boldsymbol{s}^T}{\boldsymbol{s}^T\boldsymbol{s}}
$$

i.e. $\boldsymbol{B}$ is a rank one perturbation of the matrix $\boldsymbol{A}$.

Proof. (1/4).

First of all notice that $\mathcal{B}$ is not empty, in fact

$$\frac{1}{\boldsymbol{s}^T\boldsymbol{s}}\boldsymbol{y}\boldsymbol{s}^T \in \mathcal{B} \qquad \left[\frac{1}{\boldsymbol{s}^T\boldsymbol{s}}\boldsymbol{y}\boldsymbol{s}^T\right]\boldsymbol{s} = \boldsymbol{y}$$

So that the problem is not empty. Next we reformulate the problem as a constrained minimum problem:

$$\arg\min_{\boldsymbol{B}\in\mathbb{R}^{n\times n}} \quad \frac{1}{2}\sum_{i,j=1}^{n}(A_{ij}-B_{ij})^2 \qquad \text{subject to } \boldsymbol{B}\boldsymbol{s}=\boldsymbol{y}.$$

The solution is a stationary point of the Lagrangian:

$$g(\boldsymbol{B},\boldsymbol{\lambda}) = \frac{1}{2}\sum_{i,j=1}^{n}(A_{ij}-B_{ij})^2 + \sum_{i=1}^{n}\lambda_i\left(\sum_{j=1}^{n}B_{ij}s_j - y_i\right)$$

Proof. (2/4).

taking the gradient we have

$$\frac{\partial}{\partial B_{ij}}g(\boldsymbol{B},\boldsymbol{\lambda}) = A_{ij} - B_{ij} + \lambda_i s_j = 0$$

$$\frac{\partial}{\partial \lambda_i}g(\boldsymbol{B},\boldsymbol{\lambda}) = \sum_{j=1}^{n}B_{ij}s_j - y_j = 0$$

The previous equality can be written in matrix form

$$\boldsymbol{B} = \boldsymbol{A} + \boldsymbol{\lambda}\boldsymbol{s}^T \qquad \boldsymbol{B}\boldsymbol{s} = \boldsymbol{y}$$

so that we can solve for $\boldsymbol{\lambda}$

$$\boldsymbol{B}\boldsymbol{s} = \boldsymbol{A}\boldsymbol{s} + \boldsymbol{\lambda}\boldsymbol{s}^T\boldsymbol{s} = \boldsymbol{y} \qquad \boldsymbol{\lambda} = \frac{\boldsymbol{y}-\boldsymbol{A}\boldsymbol{s}}{\boldsymbol{s}^T\boldsymbol{s}}$$

next we prove that $\boldsymbol{B}$ is the unique minimum.

Proof. (3/4).

The matrix $\boldsymbol{B}$ is a minimum, in fact

$$\|\boldsymbol{B}-\boldsymbol{A}\|_F = \left\|\boldsymbol{A}+\frac{(\boldsymbol{y}-\boldsymbol{A}\boldsymbol{s})\boldsymbol{s}^T}{\boldsymbol{s}^T\boldsymbol{s}}-\boldsymbol{A}\right\|_F = \left\|\frac{(\boldsymbol{y}-\boldsymbol{A}\boldsymbol{s})\boldsymbol{s}^T}{\boldsymbol{s}^T\boldsymbol{s}}\right\|_F$$

for all $\boldsymbol{C}\in\mathcal{B}$ we have $\boldsymbol{C}\boldsymbol{s}=\boldsymbol{y}$ so that

$$\begin{aligned}\|\boldsymbol{B}-\boldsymbol{A}\|_F &= \left\|\frac{(\boldsymbol{C}\boldsymbol{s}-\boldsymbol{A}\boldsymbol{s})\boldsymbol{s}^T}{\boldsymbol{s}^T\boldsymbol{s}}\right\|_F = \left\|(\boldsymbol{C}-\boldsymbol{A})\frac{\boldsymbol{s}\boldsymbol{s}^T}{\boldsymbol{s}^T\boldsymbol{s}}\right\|_F \\ &\leq \|\boldsymbol{C}-\boldsymbol{A}\|_F\left\|\frac{\boldsymbol{s}\boldsymbol{s}^T}{\boldsymbol{s}^T\boldsymbol{s}}\right\|_F = \|\boldsymbol{C}-\boldsymbol{A}\|_F\end{aligned}$$

because in general

$$\left\|\boldsymbol{u}\boldsymbol{v}^T\right\|_F = \left(\sum_{i,j=1}^n u_i^2 v_j^2\right)^{\frac{1}{2}} = \left(\sum_{i=1}^n u_i^2\sum_{j=1}^n v_j^2\right)^{\frac{1}{2}} = \|\boldsymbol{u}\|\,\|\boldsymbol{v}\|$$

Proof. (4/4).

Let $\boldsymbol{B}'$ and $\boldsymbol{B}''$ two different minimum. Then $\frac{1}{2}(\boldsymbol{B}'+\boldsymbol{B}'')\in\mathcal{B}$ moreover

$$\left\|\boldsymbol{A}-\frac{1}{2}(\boldsymbol{B}'+\boldsymbol{B}'')\right\|_F \leq \frac{1}{2}\|\boldsymbol{A}-\boldsymbol{B}'\|_F+\frac{1}{2}\|\boldsymbol{A}-\boldsymbol{B}''\|_F$$

If the inequality is strict we have a contradiction. From the Cauchy–Schwartz inequality we have an equality only when $\boldsymbol{A}-\boldsymbol{B}'=\lambda(\boldsymbol{A}-\boldsymbol{B}'')$ so that

$$\boldsymbol{B}'-\lambda\boldsymbol{B}''=(1-\lambda)\boldsymbol{A}$$

and

$$\boldsymbol{B}'\boldsymbol{s}-\lambda\boldsymbol{B}''\boldsymbol{s}=(1-\lambda)\boldsymbol{A}\boldsymbol{s} \quad\Rightarrow\quad (1-\lambda)\boldsymbol{y}=(1-\lambda)\boldsymbol{A}\boldsymbol{s}$$

but this is true only when $\lambda=1$, i.e. $\boldsymbol{B}'=\boldsymbol{B}''$. □

1. The update

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

satisfy the secant condition: $A_{k+1}s_k = y_k$ and A_{k+1} is the nearest matrix in the Frobenius norm that satisfy the secant condition.

2. Changing the norm we can have different results and in general you can loose uniqueness of the update.

The Broyden method (1/2)

Algorithm (The Broyden method)

$k \leftarrow 0$; x_0 and A_0 assigned;
$f_0 \leftarrow \mathbf{F}(x_0)$;
while $\|f_k\| > \epsilon$ **do**
  Solve for s_k the linear system $A_k s_k + f_k = \mathbf{0}$;
  $x_{k+1} \leftarrow x_k + s_k$;
  $f_{k+1} \leftarrow \mathbf{F}(x_{k+1})$;
  $y_k \leftarrow f_{k+1} - f_k$;
  Update: $A_{k+1} \leftarrow A_k + \dfrac{(y_k - A_k s_k)s_k^T}{s_k^T s_k}$;
  $k \leftarrow k + 1$;
end while

The Broyden method (2/2)

Notice that $\boldsymbol{y}_k - \boldsymbol{A}_k \boldsymbol{s}_k = \boldsymbol{f}_{k+1} - \boldsymbol{f}_k + \boldsymbol{f}_k$ so that the update can be written as $\boldsymbol{A}_{k+1} \leftarrow \boldsymbol{A}_k + \boldsymbol{f}_{k+1}\boldsymbol{s}_k^T / \boldsymbol{s}_k^T \boldsymbol{s}_k$ and $\boldsymbol{y}_k$ can be eliminated.

Algorithm (The Broyden method (alternative version))

$k \leftarrow 0$; $\boldsymbol{x}$ and $\boldsymbol{A}$ assigned;
$\boldsymbol{f} \leftarrow \mathbf{F}(\boldsymbol{x})$;
while $\|\boldsymbol{f}\| > \epsilon$ **do**
 Solve for $\boldsymbol{s}$ the linear system $\boldsymbol{A}\boldsymbol{s} + \boldsymbol{f} = \mathbf{0}$;
 $\boldsymbol{x} \leftarrow \boldsymbol{x} + \boldsymbol{s}$;
 $\boldsymbol{f} \leftarrow \mathbf{F}(\boldsymbol{x})$;
 Update: $\boldsymbol{A} \leftarrow \boldsymbol{A} + \dfrac{\boldsymbol{f}\boldsymbol{s}^T}{\boldsymbol{s}^T\boldsymbol{s}}$;
 $k \leftarrow k + 1$;
end while

Broyden algorithm properties (1/2)

Theorem

Let $\mathbf{F}(\boldsymbol{x})$ satisfy the standard regularity conditions with $\nabla\mathbf{F}(\boldsymbol{x}_\star)$ nonsingular. Then there exists positive constants ϵ, δ such that if $\|\boldsymbol{x}_0 - \boldsymbol{x}_\star\| \leq \epsilon$ and $\|\boldsymbol{A}_0 - \nabla\mathbf{F}(\boldsymbol{x}_\star)\| \leq \delta$, then the sequence $\{\boldsymbol{x}_k\}$ generated by the Broyden method is well defined and converge q-superlinearly to $\boldsymbol{x}_\star$, i.e.

$$\lim_{k\to\infty} \frac{\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|}{\|\boldsymbol{x}_k - \boldsymbol{x}_\star\|} = 0$$

C.G.Broyden, J.E.Dennis, J.J.Moré
On the local and super-linear convergence of quasi-Newton methods.
J. Inst. Math. Appl, **6** 222–236, 1973.

Broyden algorithm properties (2/2)

Theorem

Let $\mathbf{F}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}$ where $\boldsymbol{A} \in \mathbb{R}^{n\times n}$. Then the Broyden method converge in at most $2n$ steps.

Theorem

Let $\mathbf{F} : \mathbb{R}^n \mapsto \mathbb{R}^n$ satisfy the standard regularity conditions with $\nabla\mathbf{F}(\boldsymbol{x}_\star)$ nonsingular. Then there exists positive constants ϵ, δ such that if $\|\boldsymbol{x}_0 - \boldsymbol{x}_\star\| \leq \epsilon$ and $\|\boldsymbol{A}_0 - \nabla\mathbf{F}(\boldsymbol{x}_\star)\| \leq \delta$, then the sequence $\{\boldsymbol{x}_k\}$ generated by the Broyden method satisfy

$$\|\boldsymbol{x}_{k+2n} - \boldsymbol{x}_\star\| \leq C \|\boldsymbol{x}_k - \boldsymbol{x}_\star\|^2$$

D.M.Gay
Some convergence properties of Broyden's method.
SIAM J. Numer. Anal., **16** 623–630, 1979.

Reorganizing Broyden update

- Broyden method needs to solve a linear system for $\boldsymbol{A}_k$ at each step
- This can be onerous in terms of CPU cost
- it is possible to update directly the inverse of $\boldsymbol{A}_k$ i.e. it is possible to update $\boldsymbol{H}_k = \boldsymbol{A}_k^{-1}$.
- The update of $\boldsymbol{A}_k$ solve the problem of efficiency but do not alleviate the memory occupation
- The matrix $\boldsymbol{A}_k$ can be written as a product of simple matrix, this can save memory if the update are lesser respect to the system dimension.

Sherman-Morrison formula

Sherman-Morrison formula permit to explicit write the inverse of a matrix changed with a rank 1 perturbation

Proposition (Sherman–Morrison formula)

$$(\boldsymbol{A}+\boldsymbol{u}\boldsymbol{v}^T)^{-1} = \boldsymbol{A}^{-1} - \frac{1}{\alpha}\boldsymbol{A}^{-1}\boldsymbol{u}\boldsymbol{v}^T\boldsymbol{A}^{-1}$$

where

$$\alpha = 1 + \boldsymbol{v}^T\boldsymbol{A}^{-1}\boldsymbol{u}$$

The Sherman–Morrison formula can be checked by a direct calculation.

Application of Sherman-Morrison formula (1/2)

- From the Broyden update formula

$$\boldsymbol{A}_{k+1} = \boldsymbol{A}_k + \frac{\boldsymbol{f}_{k+1}\boldsymbol{s}_k^T}{\boldsymbol{s}_k^T\boldsymbol{s}_k}$$

- By using Sherman–Morrison formula

$$\boldsymbol{A}_{k+1}^{-1} = \boldsymbol{A}_k^{-1} - \frac{1}{\beta_k}\boldsymbol{A}_k^{-1}\boldsymbol{f}_{k+1}\boldsymbol{s}_k^T\boldsymbol{A}_k^{-1}$$

$$\beta_k = \boldsymbol{s}_k^T\boldsymbol{s}_k + \boldsymbol{s}_k^T\boldsymbol{A}_k^{-1}\boldsymbol{f}_{k+1}$$

- By setting $\boldsymbol{H}_k = \boldsymbol{A}_k^{-1}$ we have the update formula for $\boldsymbol{H}_k$:

$$\boldsymbol{H}_{k+1} = \boldsymbol{H}_k - \frac{1}{\beta_k}\boldsymbol{H}_k\boldsymbol{f}_{k+1}\boldsymbol{s}_k^T\boldsymbol{H}_k$$

$$\beta_k = \boldsymbol{s}_k^T\boldsymbol{s}_k + \boldsymbol{s}_k^T\boldsymbol{H}_k\boldsymbol{f}_{k+1}$$

Application of Sherman-Morrison formula (2/2)

- The update formula for $\boldsymbol{H}_k$:

$$\boldsymbol{H}_{k+1} = \boldsymbol{H}_k - \frac{1}{\beta_k}\boldsymbol{H}_k \boldsymbol{f}_{k+1}\boldsymbol{s}_k^T \boldsymbol{H}_k$$

$$\beta_k = \boldsymbol{s}_k^T \boldsymbol{s}_k + \boldsymbol{s}_k^T \boldsymbol{H}_k \boldsymbol{f}_{k+1}$$

- Can be reorganized as follows
  1. Compute $\boldsymbol{z}_{k+1} = \boldsymbol{H}_k \boldsymbol{f}_{k+1}$;
  2. Compute $\beta_k = \boldsymbol{s}_k^T \boldsymbol{s}_k + \boldsymbol{s}_k^T \boldsymbol{z}_{k+1}$;
  3. Compute $\boldsymbol{H}_{k+1} = \left(\boldsymbol{I} - \beta_k^{-1}\boldsymbol{z}_{k+1}\boldsymbol{s}_k^T\right)\boldsymbol{H}_k$;

The Broyden method with inverse updated

Algorithm (The Broyden method (updating inverse))

$k \leftarrow 0$; $\boldsymbol{x}_0$ *assigned;*
$\boldsymbol{f}_0 \leftarrow \mathbf{F}(\boldsymbol{x}_0)$;
$\boldsymbol{H}_0 \leftarrow \boldsymbol{I}$ *or better* $\boldsymbol{H}_0 \leftarrow \nabla \mathbf{F}(\boldsymbol{x}_0)^{-1}$;
while $\|\boldsymbol{f}_k\| > \epsilon$ **do**
  — perform step
  $\boldsymbol{s}_k \leftarrow -\boldsymbol{H}_k \boldsymbol{f}_k$;
  $\boldsymbol{x}_{k+1} \leftarrow \boldsymbol{x}_k + \boldsymbol{s}_k$;
  $\boldsymbol{f}_{k+1} \leftarrow \mathbf{F}(\boldsymbol{x}_{k+1})$;
  — update $\boldsymbol{H}$
  $\boldsymbol{z}_{k+1} \leftarrow \boldsymbol{H}_k \boldsymbol{f}_{k+1}$;
  $\beta_k \leftarrow \boldsymbol{s}_k^T \boldsymbol{s}_k + \boldsymbol{s}_k^T \boldsymbol{z}_{k+1}$;
  $\boldsymbol{H}_{k+1} \leftarrow \left(\boldsymbol{I} - \beta_k^{-1}\boldsymbol{z}_{k+1}\boldsymbol{s}_k^T\right)\boldsymbol{H}_k$;
  $k \leftarrow k+1$;
end while

- If n is very large then the storing of $\boldsymbol{H}_k$ can be very expensive.
- Moreover when n is very large we hope to find a good solution with a number m of iteration with $m \lll n$
- So that instead of storing $\boldsymbol{H}_k$ we can decide to store the vectors $\boldsymbol{z}_k$ and $\boldsymbol{s}_k$ plus the scalars β_k. With this vectors and scalars we can write

$$\boldsymbol{H}_k = \big(\boldsymbol{I} - \beta_{k-1}\boldsymbol{z}_k\boldsymbol{s}_{k-1}^T\big)\cdots\big(\boldsymbol{I} - \beta_1\boldsymbol{z}_2\boldsymbol{s}_1^T\big)\big(\boldsymbol{I} - \beta_0\boldsymbol{z}_1\boldsymbol{s}_0^T\big)\boldsymbol{H}_0$$

- Assuming $\boldsymbol{H}_0 = \boldsymbol{I}$ or can be computed on the fly we must store only $2\,n\,m + m$ real number instead of n^2 saving a lot of memory.
- However we can do better. It is possible to eliminate $\boldsymbol{z}_k$ ad store only $n\,m + m$ real numbers.

Elimination of $\boldsymbol{z}_k$ (1/3)

1. A step of the broyden iterative scheme can be rewritten as

$$\begin{aligned}
\boldsymbol{d}_k &\leftarrow \boldsymbol{H}_k\boldsymbol{f}_k \\
\boldsymbol{x}_{k+1} &\leftarrow \boldsymbol{x}_k - \boldsymbol{d}_k \\
\boldsymbol{f}_{k+1} &\leftarrow \mathbf{F}(\boldsymbol{x}_{k+1}) \\
\boldsymbol{z}_{k+1} &\leftarrow \boldsymbol{H}_k\boldsymbol{f}_{k+1} \\
\boldsymbol{H}_{k+1} &\leftarrow \left(\boldsymbol{I} + \frac{\boldsymbol{z}_{k+1}\boldsymbol{d}_k^T}{\boldsymbol{d}_k^T\boldsymbol{d}_k - \boldsymbol{d}_k^T\boldsymbol{z}_{k+1}}\right)\boldsymbol{H}_k
\end{aligned}$$

2. you can notice that $\boldsymbol{z}_k$ and $\boldsymbol{d}_k$ are similar and contains a lot of common information.
3. It is possible exploring the iteration to eliminate $\boldsymbol{z}_k$ from the update formula of $\boldsymbol{H}_k$ so that we can store the whole sequence without the vectors $\boldsymbol{z}_k$.

Elimination of z_k (2/3)

$$
\begin{aligned}
d_{k+1} = H_{k+1} f_{k+1} &= \left(I + \frac{z_{k+1} d_k^T}{d_k^T d_k - d_k^T z_{k+1}} \right) H_k f_{k+1} \\
&= \left(I + \frac{z_{k+1} d_k^T}{d_k^T d_k - d_k^T z_{k+1}} \right) z_{k+1} \\
&= z_{k+1} + \frac{z_{k+1} d_k^T z_{k+1}}{d_k^T d_k - d_k^T z_{k+1}} \\
&= \frac{d_k^T d_k}{d_k^T d_k - d_k^T z_{k+1}} z_{k+1}
\end{aligned}
$$

substituting in the update formula for H_{k+1} we obtain

$$
H_{k+1} \leftarrow \left(I + \frac{d_{k+1} d_k^T}{d_k^T d_k} \right) H_k
$$

Elimination of z_k (3/3)

Substituting into the step of the broyden iterative scheme and assuming d_k known

$$
\begin{aligned}
x_{k+1} &\leftarrow x_k - d_k \\
f_{k+1} &\leftarrow \mathbf{F}(x_{k+1}) \\
z_{k+1} &\leftarrow H_k f_{k+1} \\
d_{k+1} &\leftarrow \frac{d_k^T d_k}{d_k^T d_k - d_k^T z_{k+1}} z_{k+1} \\
H_{k+1} &\leftarrow \left(I + \frac{d_{k+1} d_k^T}{d_k^T d_k} \right) H_k
\end{aligned}
$$

notice that x_{k+1}, f_{k+1} and z_{k+1} are not used in H_{k+1} so that only d_k and its length need to be stored.

Algorithm (The Broyden method (low memory usage))

$k \leftarrow 0$; $\boldsymbol{x}$ *assigned*;
$\boldsymbol{f} \leftarrow \mathbf{F}(\boldsymbol{x})$; $\boldsymbol{H}_0 \leftarrow \nabla\mathbf{F}(\boldsymbol{x})^{-1}$; $\boldsymbol{d}_0 \leftarrow \boldsymbol{H}_0\boldsymbol{f}$; $\ell_0 \leftarrow \boldsymbol{d}_0^T\boldsymbol{d}_0$;
while $\|\boldsymbol{f}\| > \epsilon$ **do**
  — *perform step*
  $\boldsymbol{x} \leftarrow \boldsymbol{x} - \boldsymbol{d}_k$;
  $\boldsymbol{f} \leftarrow \mathbf{F}(\boldsymbol{x})$;
  — *evaluate* $\boldsymbol{H}_k\boldsymbol{f}$
  $\boldsymbol{z} \leftarrow \boldsymbol{H}_0\boldsymbol{f}$;
  for $j = 0, 1, \ldots, k-1$ **do**
    $\boldsymbol{z} \leftarrow \boldsymbol{z} + \left[(\boldsymbol{d}_j^T\boldsymbol{z})/\ell_j\right]\boldsymbol{d}_{j+1}$;
  end for
  — *update* $\boldsymbol{H}_{k+1}$
  $\boldsymbol{d}_{k+1} \leftarrow \left[\ell_k/(\ell_k - \boldsymbol{d}_k^T\boldsymbol{z})\right]\boldsymbol{z}$;
  $\ell_{k+1} \leftarrow \boldsymbol{d}_{k+1}^T\boldsymbol{d}_{k+1}$;
  $k \leftarrow k+1$;
end while

Outline

Algorithm (The dumped Broyden method)

$k \leftarrow 0$; $\boldsymbol{x}_0$ *assigned*;
$\boldsymbol{f}_0 \leftarrow \mathbf{F}(\boldsymbol{x}_0)$; $\boldsymbol{H}_0 \leftarrow \nabla\mathbf{F}(\boldsymbol{x}_0)^{-1}$;
while $\|\boldsymbol{f}_k\| > \epsilon$ **do**
  — *compute search direction*
  $\boldsymbol{d}_k \leftarrow \boldsymbol{H}_k \boldsymbol{f}_k$;
  Approximate $\arg\min_{\lambda>0} \|\mathbf{F}(\boldsymbol{x}_k - \lambda \boldsymbol{d}_k)\|^2$ *by line-search*;
  — *perform step*
  $\boldsymbol{s}_k \leftarrow -\lambda_k \boldsymbol{d}_k$;
  $\boldsymbol{x}_{k+1} \leftarrow \boldsymbol{x}_k + \boldsymbol{s}_k$;
  $\boldsymbol{f}_{k+1} \leftarrow \mathbf{F}(\boldsymbol{x}_{k+1})$;
  $\boldsymbol{y}_k \leftarrow \boldsymbol{f}_{k+1} - \boldsymbol{f}_k$;
  — *update* $\boldsymbol{H}_{k+1}$
  $\boldsymbol{H}_{k+1} \leftarrow \boldsymbol{H}_k + \dfrac{(\boldsymbol{s}_k - \boldsymbol{H}_k\boldsymbol{y}_k)\boldsymbol{s}_k^T}{\boldsymbol{s}_k^T \boldsymbol{H}_k \boldsymbol{y}_k} \boldsymbol{H}_k$;
  $k \leftarrow k+1$;
end while

Elimination of $\boldsymbol{z}_k$ (1/5)

Notice that

$$\boldsymbol{H}_k\boldsymbol{y}_k = \boldsymbol{H}_k\boldsymbol{f}_{k+1} - \boldsymbol{H}_k\boldsymbol{f}_k = \boldsymbol{z}_{k+1} - \boldsymbol{d}_k, \quad \text{and} \quad \boldsymbol{s}_k = -\lambda_k \boldsymbol{d}_k$$

and

$$\begin{aligned}
\boldsymbol{H}_{k+1} &\leftarrow \boldsymbol{H}_k + \frac{(\boldsymbol{s}_k - \boldsymbol{H}_k\boldsymbol{y}_k)\boldsymbol{s}_k^T}{\boldsymbol{s}_k^T\boldsymbol{H}_k\boldsymbol{y}_k}\boldsymbol{H}_k \\
&\leftarrow \boldsymbol{H}_k + \frac{(-\lambda_k\boldsymbol{d}_k - \boldsymbol{z}_{k+1} + \boldsymbol{d}_k)(-\lambda_k\boldsymbol{d}_k^T)}{-\lambda_k\boldsymbol{d}_k^T(\boldsymbol{z}_{k+1} - \boldsymbol{d}_k)}\boldsymbol{H}_k \\
&\leftarrow \left(\boldsymbol{I} + \frac{(-\lambda_k\boldsymbol{d}_k - \boldsymbol{z}_{k+1} + \boldsymbol{d}_k)\boldsymbol{d}_k^T}{\boldsymbol{d}_k^T(\boldsymbol{z}_{k+1} - \boldsymbol{d}_k)}\right)\boldsymbol{H}_k \\
&\leftarrow \left(\boldsymbol{I} + \frac{(\boldsymbol{z}_{k+1} + (\lambda_k - 1)\boldsymbol{d}_k)\boldsymbol{d}_k^T}{\boldsymbol{d}_k^T\boldsymbol{d}_k - \boldsymbol{d}_k^T\boldsymbol{z}_{k+1}}\right)\boldsymbol{H}_k
\end{aligned}$$

Elimination of $\boldsymbol{z}_k$ (2/5)

A step of the broyden iterative scheme can be rewritten as

$$
\begin{aligned}
\boldsymbol{d}_k &\leftarrow \boldsymbol{H}_k \boldsymbol{f}_k \\
\boldsymbol{x}_{k+1} &\leftarrow \boldsymbol{x}_k - \lambda_k \boldsymbol{d}_k \\
\boldsymbol{f}_{k+1} &\leftarrow \mathsf{F}(\boldsymbol{x}_{k+1}) \\
\boldsymbol{z}_{k+1} &\leftarrow \boldsymbol{H}_k \boldsymbol{f}_{k+1} \\
\boldsymbol{H}_{k+1} &\leftarrow \left(\boldsymbol{I} + \frac{(\boldsymbol{z}_{k+1} + (\lambda_k - 1)\boldsymbol{d}_k)\boldsymbol{d}_k^T}{\boldsymbol{d}_k^T \boldsymbol{d}_k - \boldsymbol{d}_k^T \boldsymbol{z}_{k+1}} \right) \boldsymbol{H}_k
\end{aligned}
$$

Elimination of $\boldsymbol{z}_k$ (3/5)

$$
\begin{aligned}
\boldsymbol{d}_{k+1} &= \boldsymbol{H}_{k+1} \boldsymbol{f}_{k+1} \\
&= \left(\boldsymbol{I} + \frac{(\boldsymbol{z}_{k+1} + (\lambda_k - 1)\boldsymbol{d}_k)\boldsymbol{d}_k^T}{\boldsymbol{d}_k^T \boldsymbol{d}_k - \boldsymbol{d}_k^T \boldsymbol{z}_{k+1}} \right) \boldsymbol{H}_k \boldsymbol{f}_{k+1} \\
&= \left(\boldsymbol{I} + \frac{(\boldsymbol{z}_{k+1} + (\lambda_k - 1)\boldsymbol{d}_k)\boldsymbol{d}_k^T}{\boldsymbol{d}_k^T \boldsymbol{d}_k - \boldsymbol{d}_k^T \boldsymbol{z}_{k+1}} \right) \boldsymbol{z}_{k+1} \\
&= \boldsymbol{z}_{k+1} + \frac{(\boldsymbol{z}_{k+1} + (\lambda_k - 1)\boldsymbol{d}_k)\boldsymbol{d}_k^T \boldsymbol{z}_{k+1}}{\boldsymbol{d}_k^T \boldsymbol{d}_k - \boldsymbol{d}_k^T \boldsymbol{z}_{k+1}} \\
&= \frac{(\boldsymbol{d}_k^T \boldsymbol{d}_k)\boldsymbol{z}_{k+1} + (\lambda_k - 1)(\boldsymbol{d}_k^T \boldsymbol{z}_{k+1})\boldsymbol{d}_k}{\boldsymbol{d}_k^T \boldsymbol{d}_k - \boldsymbol{d}_k^T \boldsymbol{z}_{k+1}}
\end{aligned}
$$

Elimination of z_k (4/5)

Solving for z_{k+1}

$$z_{k+1} = \frac{(d_k^T d_k - d_k^T z_{k+1}) d_{k+1} - (\lambda_k - 1)(d_k^T z_{k+1}) d_k}{d_k^T d_k}$$

and substituting in H_{k+1} we have

$$\begin{aligned} H_{k+1} &\leftarrow \left(I + \frac{(z_{k+1} + (\lambda_k - 1) d_k) d_k^T}{d_k^T d_k - d_k^T z_{k+1}} \right) H_k \\ &\leftarrow \left(I + \frac{(d_{k+1} + (\lambda_k - 1) d_k) d_k^T}{d_k^T d_k} \right) H_k \end{aligned}$$

Elimination of z_k (5/5)

Substituting into the step of the broyden iterative scheme and assuming d_k known

$$\begin{aligned} x_{k+1} &\leftarrow x_k - \lambda_k d_k \\ f_{k+1} &\leftarrow \mathsf{F}(x_{k+1}) \\ z_{k+1} &\leftarrow H_k f_{k+1} \\ d_{k+1} &\leftarrow \frac{(d_k^T d_k) z_{k+1} + (\lambda_k - 1)(d_k^T z_{k+1}) d_k}{d_k^T d_k - d_k^T z_{k+1}} \\ H_{k+1} &\leftarrow \left(I + \frac{(d_{k+1} + (\lambda_k - 1) d_k) d_k^T}{d_k^T d_k} \right) H_k \end{aligned}$$

notice that x_{k+1}, f_{k+1} and z_{k+1} are not used in H_{k+1} so that only d_k and its length need to be stored.

Algorithm (The dumped Broyden method)

$k \leftarrow 0$; $\boldsymbol{x}$ *assigned;*
$\boldsymbol{f} \leftarrow \mathbf{F}(\boldsymbol{x})$; $\boldsymbol{H}_0 \leftarrow \nabla\mathbf{F}(\boldsymbol{x})^{-1}$; $\boldsymbol{d}_0 \leftarrow \boldsymbol{H}_0\boldsymbol{f}$; $\ell_0 \leftarrow \boldsymbol{d}_0^T\boldsymbol{d}_0$;
while $\|\boldsymbol{f}_k\| > \epsilon$ **do**
  Approximate $\arg\min_{\lambda>0} \|\mathbf{F}(\boldsymbol{x} - \lambda\boldsymbol{d}_k)\|^2$ *by line-search;*
  — perform step
  $\boldsymbol{x} \leftarrow \boldsymbol{x} - \lambda_k\boldsymbol{d}_k$; $\boldsymbol{f} \leftarrow \mathbf{F}(\boldsymbol{x})$;
  —- evaluate $\boldsymbol{H}_k\boldsymbol{f}$
  $\boldsymbol{z} \leftarrow \boldsymbol{H}_0\boldsymbol{f}$;
  for $j = 0, 1, \ldots, k-1$ **do**
    $\boldsymbol{z} \leftarrow \boldsymbol{z} + \left[(\boldsymbol{d}_j^T\boldsymbol{z})/\ell_j\right]\left(\boldsymbol{d}_{j+1} + (\lambda_j - 1)\boldsymbol{d}_j\right)$;
  end for
  — update $\boldsymbol{H}_{k+1}$
  $\boldsymbol{d}_{k+1} \leftarrow \left[\ell_k\boldsymbol{z} + (\lambda_k - 1)(\boldsymbol{d}_k^T\boldsymbol{z})\boldsymbol{d}_k\right]/(\ell_k - \boldsymbol{d}_k^T\boldsymbol{z})$;
  $\ell_{k+1} \leftarrow \boldsymbol{d}_{k+1}^T\boldsymbol{d}_{k+1}$;
  $k \leftarrow k+1$;
end while

References

J. Stoer and R. Bulirsch
Introduction to numerical analysis
Springer-Verlag, Texts in Applied Mathematics, **12**, 2002.

J. E. Dennis, Jr. and Robert B. Schnabel
Numerical Methods for Unconstrained Optimization and Nonlinear Equations
SIAM, Classics in Applied Mathematics, **16**, 1996.