(Quasi) Newton methods
Theory and sparse implementation

Luca Bergamaschi
Dipartimento di Ingegneria Civile Edile ed Ambientale

Università di Padova, via Marzolo 9, Padova
e-mail luca.bergamaschi@unipd.it

December 2019

Summary

- Nonlinear systems of equations. A few examples
- Newton's method for $f(x) = 0$.
- Newton's method for systems.
- Local convergence. Exit tests.
- Global convergence.Backtracking. Line search algorithms
- Two computationally useful variants: the Inexact Newton method and the Quasi Newton method.

Some examples

- Intersection of curves in $\mathbb{R}^n$.
  Find the intersections between the circumference and the hyperbola:

$$\begin{cases} x^2 + y^2 = 4 \\ xy = 1 \end{cases}$$

- Equation describing the flow in porous media (Richards' equation):

$$\frac{\partial \psi}{\partial t} - \vec{\nabla} \cdot \left(K(\psi) \vec{\nabla} \psi \right) = f \tag{1}$$

- Function minimization (applications in data science, machine learning)

$$\min G(x) \quad \Longrightarrow \quad \text{Solve } G'(x) = 0$$

Newton's method

Given a function $f \in C^1$, we aim at finding one solution of the equation

$$f(x) = 0$$

Given x_k, an approximation to the solution ξ, we correct it to find $x_{k+1} = x_k + s$

We impose the condition $f(x_{k+1}) = 0$ and expand $f(x_{k+1})$ in Taylor series neglecting the terms of order greater or equal than 2.

$$0 = f(x_{k+1}) = f(x_k) + sf'(x_k)$$

from which

$$s = -\frac{f(x_k)}{f'(x_k)}.$$

The Newton's method can therefore be written as

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

Newton's method for system of nonlinear equations

Let us now solve the following nonlinear system

$$\left\{ \begin{array}{lcl} F_1(x_1, x_2, \cdots, x_n) & = & 0 \\ F_2(x_1, x_2, \cdots, x_n) & = & 0 \\ \cdots & = & 0 \\ F_n(x_1, x_2, \cdots, x_n) & = & 0 \end{array} \right. \tag{2}$$

more synthetically

$$F(x) = 0$$

where

$$F = \begin{pmatrix} F_1 \\ F_2 \\ \cdots \\ F_n \end{pmatrix} \qquad x = \begin{pmatrix} x_1 \\ x_2 \\ \cdots \\ x_n \end{pmatrix}$$

Let us assume that F be differentiable in an open subset of $\mathbb{R}^n$.

Newton's method for system of nonlinear equations

As in the scalar case, we try to correct an approximation x_k as $x_{k+1} = x_k + \mathbf{s}$.

Let us impose $F(x_{k+1}) = 0$ and as before expand in Taylor series the function $F(x_{k+1})$.

$$0 = F(x_{k+1}) = F(x_k) + F'(x_k)\mathbf{s}$$

where $F'(x_k)$ is the Jacobian of system (7) evaluated in x_k i. e.

$$\left(F'(x)\right)_{ij} = \frac{\partial F_i}{\partial x_j}(x)$$

As before the problem is to compute the increment $\mathbf{s}$ which is now a vector of n components.

$$\mathbf{s} = -\left(F'(x_k)\right)^{-1} F(x_k)$$

The kth iteration of the Newton's method is thus written as

$$x_{k+1} = x_k - \left(F'(x_k)\right)^{-1} F(x_k)$$

Newton's method for system of nonlinear equations

Some observations:

- The Jacobian matrix $F'(x_k)$ must be invertible.
- Local convergence of the Newton's method can be proved provided that the initial approximation x_0 is sufficiently close to the solution.
- Computation of x_{k+1} starting from x_k requires inversion of (possibly large and sparse) Jacobian matrix. This operation is inefficient as known. In practice vector **s** is evaluated by solving the following linear system

$$F'(x_k)\mathbf{s} = -F(x_k)$$

- F' is often non symmetric, so GMRES iterative method is suggested for the solution of the Newton system

Algorithm

Let us write a first version of the Algorithm, by taking into account previous comments.

Newton Algorithm

Given an initial approximation x_0 , $k := 0$.

repeat until convergence

- solve: $F'(x_k)\mathbf{s} = -F(x_k)$
- $x_{k+1} := x_k + \mathbf{s}$
- $k := k + 1$

Newton method:convergence

Standard Assumptions.

- Equation (1) has a unique solution which we denote with x^*.
- The Jacobian F' is Lipschitz continuous. There exists a real scalar γ such that

$$\|F'(y) - F'(x)\| \leq \gamma \|y - x\|$$

- $F'(x^*)$ is nonsingular.

Notations. Let us define:

- the error at the iteration k: $e_k = x_k - x^*$

Theorem

There exists $\delta > 0$ *such that*

$$\|e_0\| < \delta \implies \|e_{k+1}\| \leq K \|e_k\|^2$$

with

$$K = \gamma \|(F'(x^*))^{-1}\|$$

Proof of Newton convergence

We premise the following

Theorem

Let F be differentiable in an open set $\Omega \in \mathbb{R}^n$, and $x^* \in \Omega$. Then for $x \in \Omega$ sufficiently close to x^*

$$F(x) - F(x^*) = \int_0^1 F'((x^* + t(x - x^*))(x - x^*)dt$$

Proof.

Let $g(t) = F((x^* + t(x - x^*))$. Using the chain rule

$$g'(t) = F'((x^* + t(x - x^*))(x - x^*)$$

Hence by the Fundamental Theorem of Calculus

$$g(1) - g(0) = \int_0^1 g'(t)dt = F(x) - F(x^*).$$

$\square$

We now need two further results. The first one is also known as *Banach Lemma*

Lemma

Let A, B square $n \times n$ matrices, and B such that $\|I - BA\| < 1$. Then A, B are both nonsingular and

$$\|A^{-1}\| \leq \frac{\|B\|}{1 - \|I - BA\|}.$$

Lemma

Let the standard assumptions hold. Then there is $\delta > 0$ so that for all $x \in \mathscr{B}(\delta) = \{x : \|x - x^*\| < \delta\}$ the following hold true:

$$\begin{aligned} \|F'(x)\| &\leq 2\|F'(x^*)\| \qquad (3) \\ \|F'(x)^{-1}\| &\leq 2\|F'(x^*)^{-1}\| \qquad (4) \end{aligned}$$

Proof of Newton convergence

Proof.

(3). By triangular inequality and Lipschitz continuity

$$\|F'(x)\| - \|F'(x^*)\| \le \|F'(x) - F'(x^*)\| \le \gamma\|x - x^*\| = \gamma\|e\| \le \gamma\delta.$$

Now if $\delta < \dfrac{\|F'(x^*)\|}{\gamma}$ then $\|F'(x)\| \le \|F'(x^*)\| + \gamma\delta \le 2\|F'(x^*)\|$.

(4). Choosing now $\delta < \dfrac{1}{2\gamma\|F'(x^*)^{-1}\|}$ then

$$\|I - F'(x^*)^{-1}F'(x)\| = \|F'(x^*)^{-1}\|\|F'(x^*) - F'(x)\| \le \|F'(x^*)^{-1}\|\gamma\|e\| \le \frac{1}{2}.$$

Then applying Banach's Lemma

$$\|F'(x)^{-1}\| \le \frac{\|F'(x^*)^{-1}}{1 - \|I - F'(x^*)^{-1}F'(x)\|} \le \frac{\|F'(x^*)^{-1}\|}{1 - 1/2} = 2\|F'(x^*)^{-1}\|.$$

□

Proof of Newton convergence

We are now able to prove the main theorem.

Proof.

Let δ be smaller enough so that the previous Lemma holds.

$$\begin{aligned} e_{k+1} &= x_{k+1} - x^* = x_k - x^* - F'(x_k)^{-1} F(x_k) \\ &= e_k - F'(x_k)^{-1} \int_0^1 F'((x^* + t(x_k - x^*))\, e_k dt \\ &= F'(x_k)^{-1} \int_0^1 \left(F'(x_k) - F'((x^* + t(x_k - x^*))\right) e_k dt. \end{aligned}$$

Taking norms and again using Lipschitz continuity

$$\begin{aligned} \|e_{k+1}\| &\le \|F'(x_k)^{-1}\| \int_0^1 \gamma \|x_k - x^* - t(x_k - x^*)\| \|e_k\| dt \\ &= \|F'(x_k)^{-1}\| \gamma \|e_k\|^2 \int_0^1 (1-t) dt = \gamma \|F'(x^*)^{-1}\| \|e_k\|^2 = K \|e_k\|^2. \end{aligned}$$

$\square$

Exit test

Ideal exit test $\|e_k\| < \varepsilon$ (absolute error)

or $\|e_k\| < \varepsilon \|e_0\|$ (relative error); where ε is a prescribed tolerance.

As however x^* is not known

- Exit test on the relative residual. Stop when

$$\frac{\|F(x_k)\|}{\|F(x_0)\|} < \varepsilon$$

- Exit test on the difference. Stop when

$$\|\mathbf{s}\| = \|x_{k+1} - x_k\| < \varepsilon$$

Exit tests

Motivations

- Test on the residual. It can be proved that for a sufficiently small δ

$$\frac{1}{4\kappa}\frac{\|e_k\|}{\|e_0\|} \le \frac{\|F(x_k)\|}{\|F(x_0)\|} \le 4\kappa\frac{\|e_k\|}{\|e_0\|}$$

  where $\kappa = \|F'(x^*)\|\|(F'(x^*))^{-1}\|$ is the condition number of $F'(x^*)$.

  If $F'(x^*)$ is well-conditioned $(\kappa \approx 1)$, the test is reliable.

- Test on the difference

$$\begin{aligned} x_{k+1} - x_k &= x_{k+1} - x^* + x^* - x_k = e_{k+1} - e_k \\ \|x_{k+1} - x_k\| &= \|e_k\| + O(\|e_k\|^2) \end{aligned}$$

  The difference at step $k+1$ is of the same order of magnitude as the error at step k. (Pessimistic test)

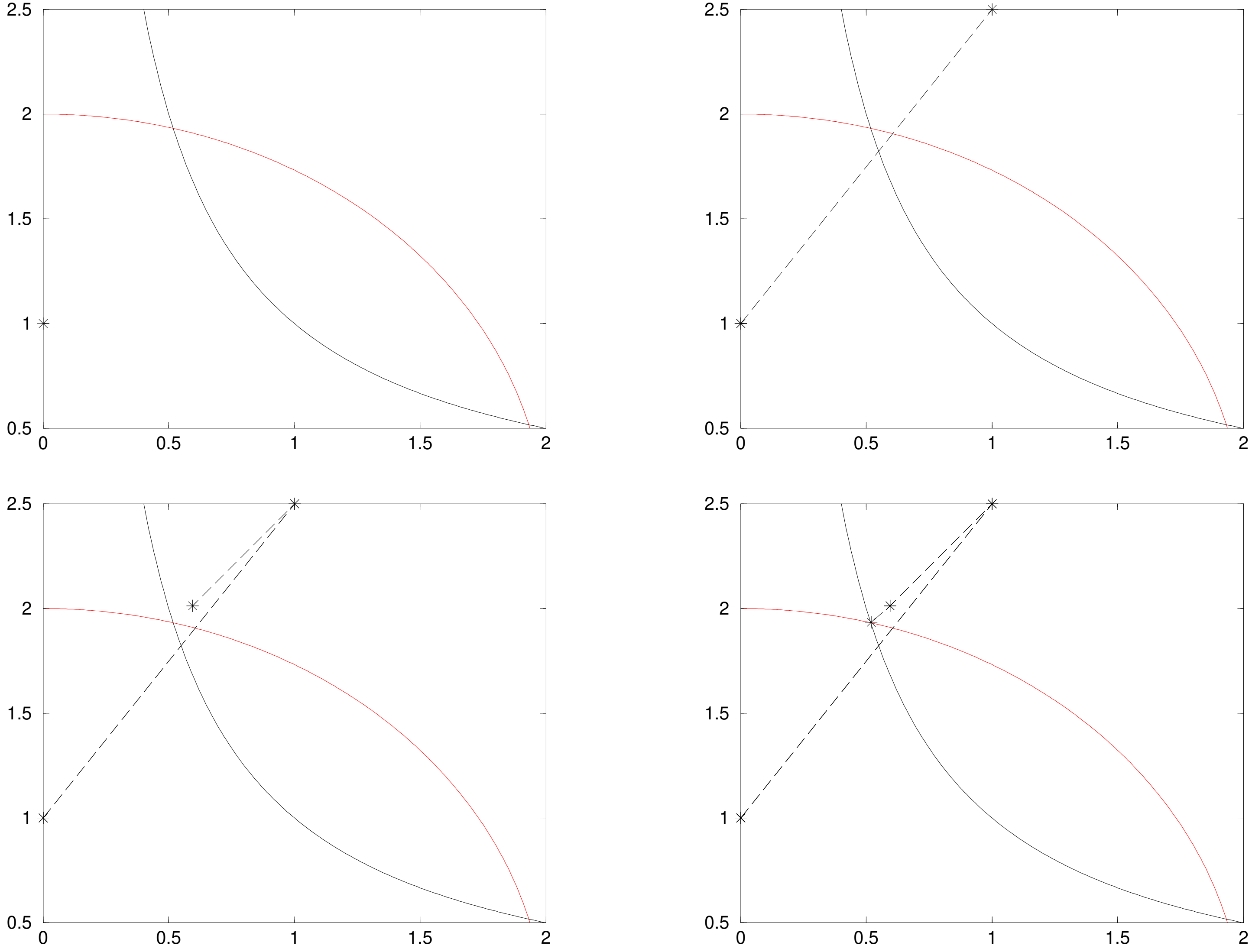
Example

$$\begin{cases} x^2 + y^2 - 4 = 0 \\ xy - 1 = 0 \end{cases} \qquad x^{(0)} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

k	$x_1^{(k)}$	$x_2^{(k)}$	$\lVert e^{(k)} \rVert$	$\lVert x^{(k+1)} - x^{(k)} \rVert$	$\dfrac{\lVert e^{(k+1)} \rVert}{\lVert e^{(k)} \rVert^2}$
0	0.000000000	1.000000000	$0.107 \times 10^{+01}$		
1	1.000000000	2.500000000	$0.745 \times 10^{+00}$	$0.180 \times 10^{+01}$	0.655899
2	0.595238095	2.011904761	$0.111 \times 10^{+00}$	$0.634 \times 10^{+00}$	0.200716
3	0.520020336	1.934236023	0.337×10^{-02}	0.108×10^{-00}	0.271153
4	0.517640404	1.931853966	0.327×10^{-05}	0.337×10^{-02}	0.288114
5	0.517638090	1.931851652	0.309×10^{-11}	0.327×10^{-05}	0.288656

$$\lVert F(x^{(0)}) \rVert = 3.16 \qquad \lVert F(x^{(1)}) \rVert = 3.58$$

Example



Global Convergence

- Convergence of Newton's method not guaranteed. Frequently Newton's step moves away from the solution
- To avoid divergence we accept Newton's step if the following condition holds: $\|F(x_{k+1})\| < \|F(x_k)\|$
- If the above condition is not satisfied, then the Newton step is reduced $\Longrightarrow$ "backtracking" or "linesearch".

Algorithm: Newton 2.

Given an initial approximation x_0 , $k := 0$.

repeat until convergence

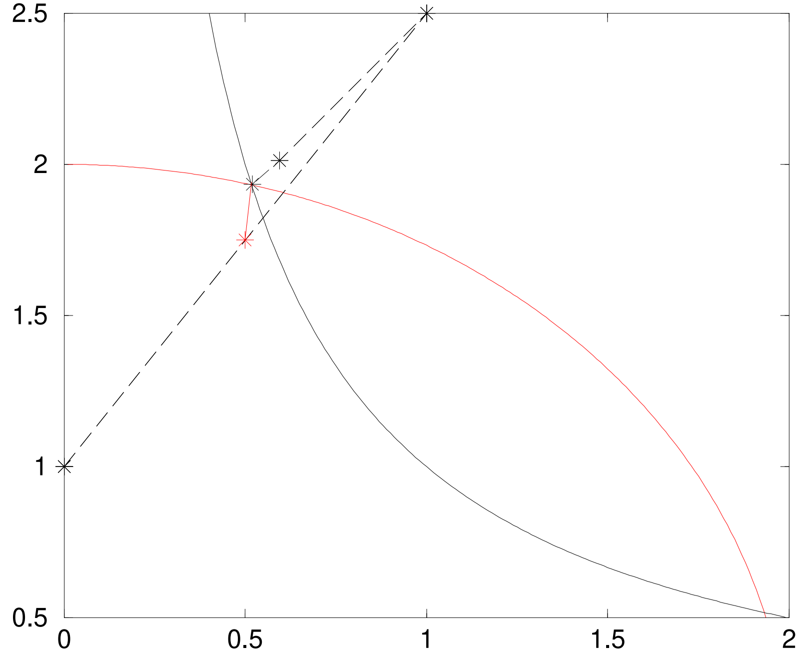
- solve: $F'(x_k)\mathbf{s} = -F(x_k)$
- $x_t := x_k + \mathbf{s}$
- if $\|F(x_t)\| < \|F(x_k)\|$ then $x_{k+1} := x_t$
- else $\mathbf{s} := \mathbf{s}/2$, go to (•)
- $k := k + 1$

Esempio
Newton con backtracking

$$\begin{cases} x^2 + y^2 - 4 = 0 \\ xy - 1 = 0 \end{cases} \qquad x^{(0)} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

k	$x_1^{(k)}$	$x_2^{(k)}$	$\|e^{(k)}\|$	$x^{(k+1)} - x^{(k)}$	$\dfrac{\|e^{(k+1)}\|}{\|e^{(k)}\|^2}$
0	0.000000000	1.000000000	$0.106597 \times 10^{+01}$		
1	0.500000000	1.750000000	$0.182705 \times 10^{+00}$	0.901388E+01	0.160790

$$\|F(x^{(0)}))\| = 3.16 \qquad \|F(x^{(1)})\| = 0.699$$



Inexact Newton Methods (Hint)

- Idea: Avoid **Oversolving** linear systems when too far from the nonlinear solution.
- Example. FE Steady-state Richards Equation

$$A(\psi)\psi = b(\psi), \qquad F(x) = A(x)x - b(x), \qquad F' = A + \frac{\partial(A)}{\partial x}x$$

- F' inherit the same size and sparsity of A.
- Generic Newton iteration:
  - Solve: $F'(x_k)\mathbf{s} = -F(x_k)$
  - $x_{k+1} := x_k + \mathbf{s}$
  - $k := k+1$
- We want to solve the linear system by an iterative method with **variable tolerance**.

$$\|F'(x_k)\mathbf{s} + F(x_k)\| \leq \eta_k \|F(x_k)\|$$

- η_k can be "large" at the beginning of Newton iteration and must be small towards the end.
- If $\eta_k \to 0$ then Newton convergence is è superlinear.
- If $\eta_k = O(\|F(x_k)\|)$ then again quadratic convergence can be proved.

Quasi-Newton Methods

Motivation: Jacobian matrix

- Not always explicitly available (sometimes function F is known as a set of data)

  or

- Differentiation of F may be too costly to be afforded at every Newton iteration

A possible answer to this problem is given by the quasi-Newton methods which compute a sequence of approximate Jacobians possibly starting from the 'true' initial Jacobian.

Instead of solving

$$x_{k+1} = x_k - F'(x_k)^{-1} F(x_k)$$

we solve

$$x_{k+1} = x_k - B_k^{-1} F(x_k)$$

Quasi-Newton Methods

Sequence of B_k can be constructed in many ways. The simplest approach is due to Broyden:

$$B_{k+1} = B_k + \frac{(y - B_k s)s^T}{s^T s} = B_k + \frac{F(x_{k+1})s^T}{s^T s}$$

where $y = F(x_{k+1}) - F(x_k)$ and using $B_k s_k = -F(x_k)$.

The Broyden update formula satisfies:

1. the secant condition, namely $B_{k+1} s = y$.
2. B_{k+1} is the closest matrix to B_k in the Frobenius norm among all the matrices satisfying the secant condition.

$$B_{k+1} = \begin{array}{c}\text{argmin} \\ B : Bs = y\end{array} \|B - B_k\|$$

NOTE: Frobenius norm of a matrix is defined as

$$\|A\|_F = \sqrt{\sum_i \sum_j a_{ij}^2} = \mathrm{tr}(A^T A)$$

Quasi-Newton Methods

The secant condition $\boxed{B_{k+1}s = y}$ is a generalization of the secant method (Regula falsi):

$$x_{k+1} = x_k - \frac{f(x_k)}{b_k}$$

where

$$b_k = \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}} = \frac{y}{s}.$$

- In n dimension, the secant condition has an infinite number of solutions.
- Broyden's choice satisfies secant condition, in fact

$$B_{k+1}s = B_k s + \frac{(y - B_k s)s^T}{s^T s} s = B_k s + y - B_k s = y.$$

- Broyden's choice satisfies the least change condition.

$$\begin{aligned} \|B_{k+1} - B_k\| &= \left\| \frac{(y - B_k s)s^T}{s^T s} \right\|_F = \left\| \frac{(Bs - B_k s)s^T}{s^T s} \right\|_F = \\ &= \left\| \frac{(B - B_k)ss^T}{s^T s} \right\|_F \le \|B - B_k\|_F \left\| \frac{ss^T}{s^T s} \right\|_F = \|B - B_k\|_F . \end{aligned}$$

using $\left\| \frac{ss^T}{s^T s} \right\|_F = 1$ (proof by exercise).

Convergence Results

Definition

A sequence $\{x_k\}$ converges superlinearly to x^* if there are $\alpha > 1$ and $K > 0$ such that

$$\|x_{k+1} - x^*\| \leq K\|x_k - x^*\|^\alpha$$

Let us now define the error in jacobian approximations:

$$E_k = B_k - F'(x^*)$$

The first Theorem states that the difference between the exact and the approximate Jacobian does not grow with the Newton iteration. This property is also called *bounded deterioration*.

Theorem

$$\|E_{k+1}\| \leq \|E_k\| + \frac{\gamma}{2}(\|e_k\| + \|e_{k+1}\|)$$

Convergence Results and implementation

Theorem

Let the standard assumption holds. Then there are δ and δ_B such that if $\|e_0\| < \delta$ and $\|E_0\| < \delta_B$ the Broyden sequence exists and $x_n \to x^*$ superlinearly.

This theorem states that we can make $\|E_k\|$ as small as we want by properly choosing the initial vector x_0 and the initial Jacobian approximation B_0.

If it is the case, the convergence of the iteration remains very fast (superlinear convergence).

Problem.
How to implement solution of Newton system with B_k^{-1} instead of $J(x_k)^{-1}$?
Note that even if B_0 is sparse B_1 is not.

Careful implementation should avoid inversion of dense matrices.

Sparse implementation of Broyden method: Sherman Morrison formula.

Theorem

$$(B+uv^T)^{-1} = \left(I - \frac{(B^{-1}u)v^T}{1+v^TB^{-1}u}\right)B^{-1} = B^{-1} - \frac{(B^{-1}u)v^TB^{-1}}{1+v^TB^{-1}u}$$

Proof.

Let us look for the inverse of $B+uv^T$ as $B^{-1}+xy^T$. The following conditions must hold:

$$\begin{array}{llll} (1) & I & = & (B+uv^T)\cdot(B^{-1}+xy^T) = I+uv^TB^{-1}+Bxy^T+uv^Txy^T \\ (2) & I & = & (B^{-1}+xy^T)\cdot(B+uv^T) = I+xy^TB+B^{-1}uv^T+xy^Tuv^T \end{array}$$

Multiplying the first by u on the right and the second by v^T on the left yields:

$$\begin{array}{rcc} u\left(v^TB^{-1}u\right)+Bx(y^Tu)+u(v^Txy^Tu)=0 & \Longrightarrow & x=\alpha B^{-1}u. \\ (v^Tx)y^TB+(v^TB^{-1}u)v^T+(v^Txy^Tu)v^T=0 & \Longrightarrow & y=\beta B^{-1}v. \end{array}$$

Without loss of generality set $\beta=1$ hence substituting in (1) we obtain

$$0 = uv^TB^{-1}+\alpha uv^TB^{-1}+uv^T\alpha B^{-1}uv^TB^{-1} = uv^TB^{-1}\left(1+\alpha(1+v^TB^{-1}u)\right)$$

Finally we get $\alpha = \dfrac{-1}{1+v^TB^{-1}u}$ which completes the proof. □

Sparse implementation of Broyden method

Need to compute $B_k^{-1}F(x_k)$ without

1. Computing B_k^{-1} since we do not want to invert matrices.
2. Computing B_k since it is dense.

In our context we need to evaluate B_{k+1}^{-1} in terms of B_k^{-1} starting from

$$B_{k+1} = B_k + u_k v_k,$$

where we can define among the others

$$u_k = \frac{F(x_{k+1})}{\|s_k\|}, \qquad v_k = \frac{s_k}{\|s_k\|}, \qquad \text{so that}$$

Sparse implementation of Broyden method

$$
\begin{aligned}
B_{k+1}^{-1} &= (B_k + u_k v_k^T)^{-1} = \left(I - \frac{(B_k^{-1} u_k) v_k^T}{1 + v_k^T B_k^{-1} u_k} \right) B_k^{-1} \\
&= \left(I - w_k v_k^T \right) B_k^{-1}
\end{aligned}
$$

Where we have defined $w_k = \dfrac{B_k^{-1} u_k}{1 + v_k^T B_k^{-1} u_k}$. Now by induction

$$
B_k^{-1} = \left(I - w_{k-1} v_{k-1}^T \right) \left(I - w_{k-2} v_{k-2}^T \right) \cdots \left(I - w_0 v_0^T \right) B_0^{-1}
$$

Sparse implementation of Broyden method

Important results: $s_k = -B_k^{-1} F_k$ is accomplished by

1. Solving the system $B_0 z_0 = -F_k$
2. Computing $\alpha_0 = w_0^T z_0$, then $z_1 = z_0 - \alpha_0 w_0$
   Computing $\alpha_1 = w_1^T z_1$, then $z_2 = z_1 - \alpha_1 w_1$
   $\dots$
   Computing $\alpha_{k-1} = w_{k-1}^T z_{k-1}$, then $z_k = z_{k-1} - \alpha_{k-1} w_{k-1}$

Problem. We do not know how to compute $w_j, j = 1, \cdots, k-1$. Let us define

$p = B_{k-1}^{-1} F(x_k) = \left(I - w_{k-2} v_{k-2}^T\right) \cdots \left(I - w_0 v_0^T\right) B_0^{-1} F(x_k)$

It follows that

$$\begin{aligned} s_k &= -B_k^{-1} F_k = -\left(I - w_{k-1} v_{k-1}^T\right) p = w_{k-1}(v_{k-1}^T p) - p \\ B_{k-1}^{-1} u_{k-1} &= B_{k-1}^{-1} \frac{F_k}{\|s_{k-1}\|} = \frac{p}{\|s_{k-1}\|} \\ w_{k-1} &= \frac{B_{k-1}^{-1} u_{k-1}}{1 + v_{k-1}^T B_{k-1}^{-1} u_{k-1}} = \frac{p}{\|s_{k-1}\| + v_{k-1}^T p} \end{aligned}$$

Sparse implementation of Broyden method

Now combining $s_k = w_{k-1}(v_{k-1}^T p) - p$ with $w_{k-1} = \dfrac{p}{\|s_{k-1}\| + v_{k-1}^T p}$ we obtain

$$\|s_{k-1}\| w_{k-1} = p - w_{k-1} v_{k-1}^T p = -s_k$$

hence

$$w_{k-1} = -\frac{s_k}{\|s_{k-1}\|}$$

Hence B_k^{-1} can be written in terms of sequence $\{s_j\}$ only as

$$B_k^{-1} = \prod_{j=0}^{k-1} (I - w_j v_j) = \prod_{j=0}^{k-1} \left(I + \frac{s_{j+1} s_j^T}{\|s_j\|_2^2} \right)$$

NOTE: We know s_k as a function of B_k^{-1} and B_k^{-1} as a function of s_k.

Sparse implementation of Broyden method

Let us write s_{k+1} as

$$
\begin{aligned}
s_{k+1} = -B_{k+1}^{-1} F_{k+1} \quad &= \quad -\left(I + \frac{s_{k+1} s_k^T}{\|s_k\|_2^2} \right) \prod_{j=1}^{k-1} \left(I + \frac{s_{j+1} s_j^T}{\|s_j\|_2^2} \right) B_0^{-1} F_k \\
&= \quad -\left(I + \frac{s_{k+1} s_k^T}{\|s_k\|_2^2} \right) B_k^{-1} F_{k+1}
\end{aligned}
$$

or

$$
s_{k+1} = -B_k^{-1} F_{k+1} - s_{k+1} \frac{s_k^T B_k^{-1} F_{k+1}}{\|s_k\|_2^2}
$$

Finally solve for s_{k+1} to obtain

$$
s_{k+1} = \frac{-B_k^{-1} F_{k+1}}{1 + s_k^T B_k^{-1} F_{k+1} / \|s_k\|_2^2}
$$

Broyden Algorithm (sketch)

- INPUT: x_0, B_0. Set $k := 0, x := x_0$.
- First step: Solve $B_0 s_0 = -F(x_0)$
- repeat until convergence
  - $x := x + s_k$
  - Solve $B_0 z = -F(x)$
  - $k := k + 1$.
  - for $j := 0$ to $k - 1$

$$z := z + s_{j+1} \frac{s_j^T z}{\|s_j\|_2^2}$$

  - end for
  - $s_{k+1} := \dfrac{z}{1 - s_k^T z / \|s_k\|_2^2}$
- end repeat

---

1. Only a system with B_0 needed to be solved at each Newton step.
2. k scalar products and 1 vector norm at k-th step. Complexity increasing with k.