Lecture 7: Quasi-Newton methods. Nonlinear least-squares problems and the Gauss-Newton method

Coralia Cartis, Mathematical Institute, University of Oxford

C6.2/B2: Continuous Optimization

Practical comments: calculating derivatives

How to compute/provide derivatives to a solver?

- Calculate derivatives by hand when easy/simple objective and constraints; user provides code that computes them.
- Calculate or approximate derivatives automatically:
  - Automatic differentiation: breaks down computer code for evaluating f into elementary arithmetic operations + differentiate by chain rule. Software: ADIFOR, ADOL-C.
  - Symbolic differentiation: manipulate the algebraic expression of f (if available). Software: symbolic packages of MAPLE, MATHEMATICA, MATLAB.
  - Finite differencing $\longrightarrow$ approximate derivatives.

See Nocedal & Wright, Numerical Optimization (2nd edition, 2006) for more details of the above procedures.

Approximating the Hessian matrix by finite differences

Approximating the Hessian from gradient vals: $i \in \{1, \ldots, n\}$;

$$[\nabla^2 f(x)]e^i \approx \frac{1}{h}[\nabla f(x + he^i) - \nabla f(x)]$$

Cost of approximating $\nabla^2 f(x)$ is $n + 1$ gradient evaluations.

For all finite-differencing, careful with the choice of h in computations:

- “too large” $h \to$ inaccurate approximations,
- “too small” $h \to$ numerical cancellation errors.

But successful techniques exist for smooth noiseless problems when sufficient function and/or gradient values can be computed.

For noisy problems, use derivative-free optimization methods (if problem size is not too large).

Quasi-Newton methods

Secant approximations for computing $B^k \approx \nabla^2 f(x^k)$

At the start of the GLM, choose B^0 (say, $B^0 := I$). After computing $s^k = -(B^k)^{-1}\nabla f(x^k)$ and $x^{k+1} = x^k + \alpha^k s^k$, compute update B^{k+1} of B^k.

Wish list:

Quasi-Newton methods

Secant approximations for computing $B^k \approx \nabla^2 f(x^k)$

At the start of the GLM, choose B^0 (say, $B^0 := I$). After computing $s^k = -(B^k)^{-1}\nabla f(x^k)$ and $x^{k+1} = x^k + \alpha^k s^k$, compute update B^{k+1} of B^k.

Wish list:

Compute B^{k+1} as a function of already-computed quantities $\nabla f(x^{k+1})$, $\nabla f(x^k)$, ..., $\nabla f(x^0)$, B^k, s^k,

Quasi-Newton methods

Secant approximations for computing $B^k \approx \nabla^2 f(x^k)$

At the start of the GLM, choose B^0 (say, $B^0 := I$). After computing $s^k = -(B^k)^{-1}\nabla f(x^k)$ and $x^{k+1} = x^k + \alpha^k s^k$, compute update B^{k+1} of B^k.

Wish list:

Compute B^{k+1} as a function of already-computed quantities $\nabla f(x^{k+1})$, $\nabla f(x^k)$, ..., $\nabla f(x^0)$, B^k, s^k,

B^{k+1} should be symmetric, nonsingular (pos. def.),

Quasi-Newton methods

Secant approximations for computing $B^k \approx \nabla^2 f(x^k)$

At the start of the GLM, choose B^0 (say, $B^0 := I$). After computing $s^k = -(B^k)^{-1}\nabla f(x^k)$ and $x^{k+1} = x^k + \alpha^k s^k$, compute update B^{k+1} of B^k.

<u>Wish list:</u>

Compute B^{k+1} as a function of already-computed quantities $\nabla f(x^{k+1})$, $\nabla f(x^k)$, ..., $\nabla f(x^0)$, B^k, s^k,

B^{k+1} should be symmetric, nonsingular (pos. def.),

B^{k+1} "close" to B^k, a "cheap" update of B^k, $B^k \to \nabla^2 f(x^k)$, etc.

Quasi-Newton methods

Secant approximations for computing $B^k \approx \nabla^2 f(x^k)$

At the start of the GLM, choose B^0 (say, $B^0 := I$). After computing $s^k = -(B^k)^{-1}\nabla f(x^k)$ and $x^{k+1} = x^k + \alpha^k s^k$, compute update B^{k+1} of B^k.

Wish list:

Compute B^{k+1} as a function of already-computed quantities $\nabla f(x^{k+1})$, $\nabla f(x^k)$, $\ldots$, $\nabla f(x^0)$, B^k, s^k,

B^{k+1} should be symmetric, nonsingular (pos. def.),

B^{k+1} "close" to B^k, a "cheap" update of B^k, $B^k \to \nabla^2 f(x^k)$, etc.

$\Longrightarrow$ a new class of methods: faster than steepest descent method, cheaper to compute per iteration than Newton's.

Quasi-Newton methods

Secant approximations for computing $B^k \approx \nabla^2 f(x^k)$

At the start of the GLM, choose B^0 (say, $B^0 := I$). After computing $s^k = -(B^k)^{-1}\nabla f(x^k)$ and $x^{k+1} = x^k + \alpha^k s^k$, compute update B^{k+1} of B^k.

Wish list:

Compute B^{k+1} as a function of already-computed quantities $\nabla f(x^{k+1})$, $\nabla f(x^k)$, ..., $\nabla f(x^0)$, B^k, s^k,

B^{k+1} should be symmetric, nonsingular (pos. def.),

B^{k+1} "close" to B^k, a "cheap" update of B^k, $B^k \to \nabla^2 f(x^k)$, etc.

$\Longrightarrow$ a new class of methods: faster than steepest descent method, cheaper to compute per iteration than Newton's.

For the first wish, choose B^{k+1} to satisfy the secant equation:

$$\gamma^k := \nabla f(x^{k+1}) - \nabla f(x^k) = B^{k+1}(x^{k+1} - x^k) = B^{k+1}\alpha^k s^k.$$

Quasi-Newton methods ...

Interpretation of the secant equation:

$$\gamma^k := \nabla f(x^{k+1}) - \nabla f(x^k) = B^{k+1}(x^{k+1} - x^k) = B^{k+1}\alpha^k s^k.$$

- It is satisfied by $B^{k+1} := \nabla^2 f$ when f is a quadratic function:
  Let $f(x) = g^T x + \frac{1}{2} x^T H x$; then $\nabla f(x) = Hx + g$ and $\nabla^2 f = H$. Thus $\nabla f(x^{k+1}) - \nabla f(x^k) = H(x^{k+1} - x^k)$ and so the secant equation holds with $B^{k+1} := \nabla^2 f$.

Quasi-Newton methods ...

Interpretation of the secant equation:

$$\gamma^k := \nabla f(x^{k+1}) - \nabla f(x^k) = B^{k+1}(x^{k+1} - x^k) = B^{k+1}\alpha^k s^k.$$

- It is satisfied by $B^{k+1} := \nabla^2 f$ when f is a quadratic function:
  Let $f(x) = g^T x + \frac{1}{2}x^T H x$; then $\nabla f(x) = Hx + g$ and $\nabla^2 f = H$. Thus $\nabla f(x^{k+1}) - \nabla f(x^k) = H(x^{k+1} - x^k)$ and so the secant equation holds with $B^{k+1} := \nabla^2 f$.
- In general (not just in the quadratic case), the change in gradient contains information about the Hessian (recall finite differences).

Quasi-Newton methods ...

Interpretation of the secant equation:

$$\gamma^k := \nabla f(x^{k+1}) - \nabla f(x^k) = B^{k+1}(x^{k+1} - x^k) = B^{k+1}\alpha^k s^k.$$

- It is satisfied by $B^{k+1} := \nabla^2 f$ when f is a quadratic function:
  Let $f(x) = g^T x + \frac{1}{2} x^T H x$; then $\nabla f(x) = Hx + g$ and $\nabla^2 f = H$. Thus $\nabla f(x^{k+1}) - \nabla f(x^k) = H(x^{k+1} - x^k)$ and so the secant equation holds with $B^{k+1} := \nabla^2 f$.
- In general (not just in the quadratic case), the change in gradient contains information about the Hessian (recall finite differences).
- The new model must predict correctly the change in gradient (see next slide for details).

Quasi-Newton methods ...

Interpretation of the secant equation: (continued)

The gradient change predicted by the current quadratic model is

$\nabla f(x^{k+1}) - \nabla f(x^k) \approx \nabla m(x^k + \alpha^k s^k) - \nabla m(x^k) = -\alpha^k \nabla f(x^k)$,

where $\quad m(x^k + s) = f(x^k) + \nabla f(x^k)^\top s + \frac{1}{2} s^\top B^k s$

and $\quad s^k = -(B^k)^{-1} \nabla f(x^k)$.

Quasi-Newton methods ...

Interpretation of the secant equation: (continued)

The gradient change predicted by the current quadratic model is

$\nabla f(x^{k+1}) - \nabla f(x^k) \approx \nabla m(x^k + \alpha^k s^k) - \nabla m(x^k) = -\alpha^k \nabla f(x^k)$,

where $\quad m(x^k + s) = f(x^k) + \nabla f(x^k)^\top s + \frac{1}{2} s^\top B^k s$

and $\quad s^k = -(B^k)^{-1} \nabla f(x^k)$.

Want the new quadratic model

$$m_+(x^k + s) := f(x^k) + \nabla f(x^k)^\top s + \frac{1}{2} s^\top B^{k+1} s$$

to predict correctly the change in gradient γ^k at $s := x^{k+1} - x^k$

Quasi-Newton methods ...

Interpretation of the secant equation: (continued)

The gradient change predicted by the current quadratic model is

$\nabla f(x^{k+1}) - \nabla f(x^k) \approx \nabla m(x^k + \alpha^k s^k) - \nabla m(x^k) = -\alpha^k \nabla f(x^k)$,

where $\quad m(x^k + s) = f(x^k) + \nabla f(x^k)^\top s + \frac{1}{2} s^\top B^k s$

and $\quad s^k = -(B^k)^{-1} \nabla f(x^k)$.

Want the new quadratic model

$$m_+(x^k + s) := f(x^k) + \nabla f(x^k)^\top s + \frac{1}{2} s^\top B^{k+1} s$$

to predict correctly the change in gradient γ^k at $s := x^{k+1} - x^k$

$$\begin{aligned} \gamma^k &= \nabla f(x^{k+1}) - \nabla f(x^k) \\ &= \nabla m_+(x^{k+1}) - \nabla m_+(x^k) \\ &= B^{k+1}(x^{k+1} - x^k) + \nabla f(x^k) - \nabla f(x^k) \\ &= B^{k+1}(x^{k+1} - x^k) \end{aligned}$$

Quasi-Newton methods ...

Many ways to compute B^{k+1} to satisfy the secant equation.
Trade-off between "wishes" on the list for some of the methods.

Quasi-Newton methods ...

Many ways to compute B^{k+1} to satisfy the secant equation. Trade-off between "wishes" on the list for some of the methods.

Symmetric rank 1 updates. [see Prob Sheet 3]

Set $B^{k+1} := B^k + u^k(u^k)^\top$, for some $u^k \in \mathbb{R}^n$, and all $k \geq 0$.

- B^{k+1} symmetric, "close" to B^k.
- Work per iteration: $\mathcal{O}(n^2)$ (as opposed to the $\mathcal{O}(n^3)$ of Newton), due to Sherman-Morrison-Woodbury formula!

Quasi-Newton methods ...

Many ways to compute B^{k+1} to satisfy the secant equation.
Trade-off between "wishes" on the list for some of the methods.

Symmetric rank 1 updates.

[see Prob Sheet 3]

Set $B^{k+1} := B^k + u^k(u^k)^\top$, for some $u^k \in \mathbb{R}^n$, and all $k \geq 0$.

- B^{k+1} symmetric, "close" to B^k.
- Work per iteration: $\mathcal{O}(n^2)$ (as opposed to the $\mathcal{O}(n^3)$ of Newton), due to Sherman-Morrison-Woodbury formula!

The secant equation $\Longrightarrow u^k = (\gamma^k - B^k\delta^k)/\rho^k$,
where $\delta^k := x^{k+1} - x^k = \alpha^k s^k, \;\; (\rho^k)^2 := (\gamma^k - B^k\delta^k)^\top \delta^k > 0$.

Quasi-Newton methods ...

Many ways to compute B^{k+1} to satisfy the secant equation.
Trade-off between "wishes" on the list for some of the methods.

Symmetric rank 1 updates. [see Prob Sheet 3]

Set $B^{k+1} := B^k + u^k(u^k)^\top$, for some $u^k \in \mathbb{R}^n$, and all $k \geq 0$.

- B^{k+1} symmetric, "close" to B^k.
- Work per iteration: $\mathcal{O}(n^2)$ (as opposed to the $\mathcal{O}(n^3)$ of Newton), due to Sherman-Morrison-Woodbury formula!

The secant equation $\Longrightarrow u^k = (\gamma^k - B^k\delta^k)/\rho^k$,
where $\delta^k := x^{k+1} - x^k = \alpha^k s^k, \ \ (\rho^k)^2 := (\gamma^k - B^k\delta^k)^\top\delta^k > 0$.

- B^k may not be positive definite, s^k may not be descent.
- ρ^k may be close to zero leading to large updates.

Other updates: BFGS, DFP, Broyden family, etc.

Quasi-Newton methods ...

BFGS updates.

[see Prob Sheet 3]

- Broyden-Fletcher-Goldfarb-Shanno (independently).

Set $B_{k+1} := B_k + u_k u_k^\top + v_k v_k^\top$, for some $u_k \in \mathbb{R}^n$, $v_k \in \mathbb{R}^n$.

- It is a rank 2 update (if u_k and v_k are linearly independent).
- SWM formula yields $\mathcal{O}(n^2)$ operations/iteration.
- In practice, update the Cholesky factors of B_k (still $\mathcal{O}(n^2)$).

Quasi-Newton methods ...

BFGS updates.

[see Prob Sheet 3]

- Broyden-Fletcher-Goldfarb-Shanno (independently).

Set $B_{k+1} := B_k + u_k u_k^\top + v_k v_k^\top$, for some $u_k \in \mathbb{R}^n$, $v_k \in \mathbb{R}^n$.

- It is a rank 2 update (if u_k and v_k are linearly independent).
- SWM formula yields $\mathcal{O}(n^2)$ operations/iteration.
- In practice, update the Cholesky factors of B_k (still $\mathcal{O}(n^2)$).

Given $B_k = J_k J_k^\top$, where J_k arbitrary nonsingular, and $\|\cdot\|_F$ Frobenius norm, let J_{k+1} solve

$$\min_J \|J - J_k\|_F \quad \text{subject to} \quad J\delta_k = \gamma_k.$$

$$\Rightarrow \quad B_{k+1} := J_{k+1} J_{k+1}^\top = B_k + u_k u_k^\top + v_k v_k^\top,$$

where $u_k u_k^\top = -B_k \delta_k \delta_k^\top B_k / (\delta_k^\top B_k \delta_k)$, $v_k v_k^\top = \gamma_k \gamma_k^\top / (\gamma_k^\top \delta_k)$.

- Let $J_k := L_k$ the lower triangular Cholesky factor of B_k.

Quasi-Newton methods ...

BFGS updates. (continued)

- Thus B_{k+1} is "close" to B_k.
- B_k symmetric pos. def. $\Rightarrow B_{k+1}$ symmetric pos. def. (provided $(\delta^k)^T\gamma^k > 0$, ensured by say, Wolfe linesearch)

Quasi-Newton methods ...

BFGS updates. (continued)

- Thus B_{k+1} is "close" to B_k.
- B_k symmetric pos. def. $\Rightarrow B_{k+1}$ symmetric pos. def. (provided $(\delta^k)^T\gamma^k > 0$, ensured by say, Wolfe linesearch)
- BFGS method: GLM with $s_k := -B_k^{-1}\nabla f(x_k)$, with B_k updated by BFGS formula on each iteration.
- For global convergence of BFGS method, must use Wolfe linesearch to compute stepsize instead of bArmijo linesearch.
- The BFGS method has local Q-superlinear convergence!
- When applying the BFGS method with exact linesearches, to a strictly convex quadratic function f, then $B_k = \nabla^2 f$ after n iterations.
- Satisfies all the wishes on the wish list! Has been very popular when second derivatives of f are not available.

Nonlinear least-squares problems

- a way to solve overdetermined (linear and nonlinear) systems of equations:

  $r : \mathbb{R}^n \to \mathbb{R}^m$ with $m \geq n$; $r(x) = 0$ or $r(x) \approx 0$.

  $$\Downarrow$$

  $$\min_{x\in\mathbb{R}^n} f(x) := \frac{1}{2}\sum_{j=1}^{m}[r_j(x)]^2 = \frac{1}{2}\|r(x)\|^2. \qquad \text{(L/N LS)}$$

  $\Longrightarrow$ unconstrained optimization problems with special structure.

- often, computationally cheaper to solve if structure is exploited:

  $\longrightarrow$ “simplify” damped Newton’s method to exploit this structure.

- many applications: data fitting, data assimilation for weather forecasting, climate modelling, etc.

Data fitting application

Times $t_j \quad \longrightarrow \quad y_j$, $j = \overline{1, m}$, measurements.

Model: $\Phi(x, t)$, continuous in t; parameters $x \in \mathbb{R}^n$, $n < m$.

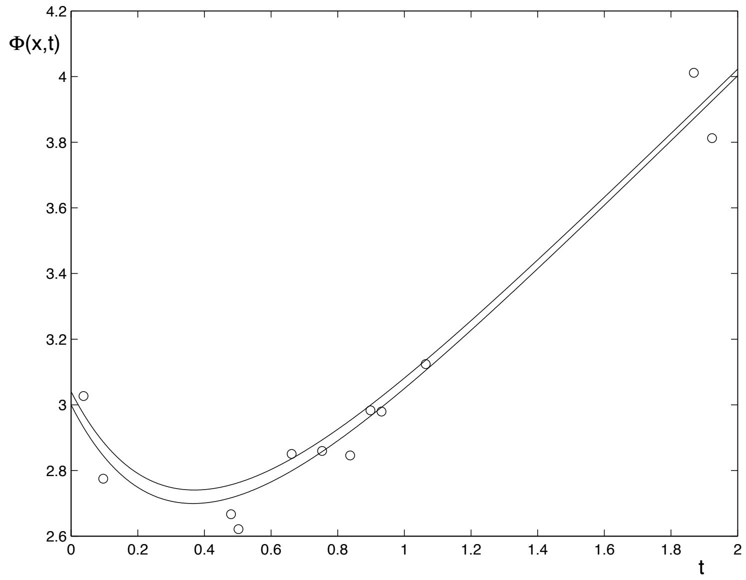
Find x: $\Phi(x, t_j)$ "close to" y_j, $j = \overline{1, m}$;

Choice of model: $\Phi(x, t) = x_1 + x_2 t + e^{-x_3 t}$, where $x = (x_1, x_2, x_3) \in \mathbb{R}^3$.

$$\min_{x \in \mathbb{R}^3} \frac{1}{2} \sum_{j=1}^{m} (\Phi(x, t_j) - y_j)^2.$$

$\longrightarrow x^*$.

Optimal model: $\Phi(x^*, t)$.



- In (NLS), let $r_j(x) := \Phi(x, t_j) - y_j$, $j = \overline{1, m}$: residuals.

The Linear Least-Squares (LLS) problem

- $r(x) := Jx + r, \forall\, x \in \mathbb{R}^n; J \in \mathbb{R}^{m\times n}, r \in \mathbb{R}^m, m \geq n.$

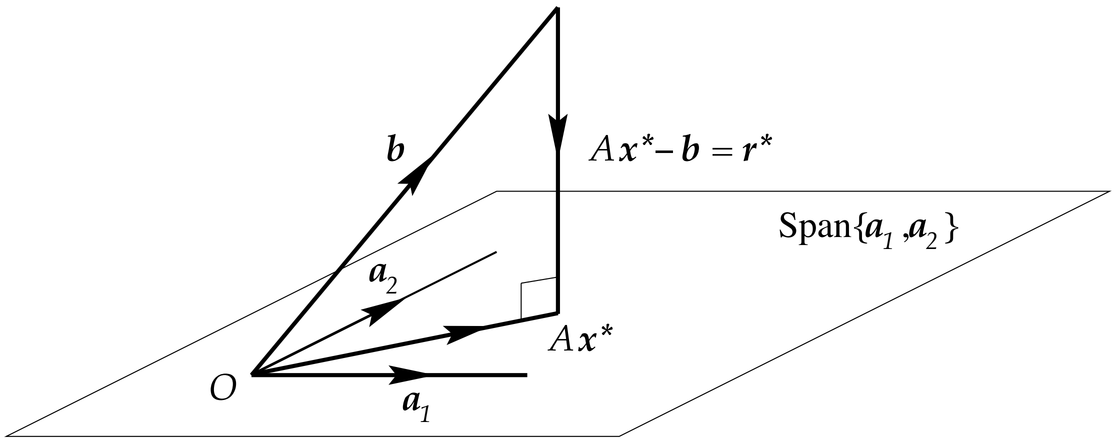
(LLS) $\min_{x\in\mathbb{R}^n} f(x) := \frac{1}{2}\|Jx + r\|^2 = \frac{1}{2}x^T J^T Jx + x^T J^T r + \frac{1}{2}\|r\|^2$

- f convex quadratic; (global) minimizer x^* of f == solution of linear system (normal equations)

$$J^T(Jx^* + r) = 0 \iff J^T Jx^* = -J^T r.$$

Geometrical interpretation:

- $r(x) = Ax - b$.
LLS: find orthogonal projection of b onto the subspace/plane determined by the columns of A.



- computing x^*: Cholesky factorization of $J^T J$; QR or SVD of J.

A simple LLS example

Fit a line to the data $(t_i, y_i) \in \{(-1,3),(0,2),(1,0),(2,4)\}$.

- for some $x = (x_1 \; x_2)^T \in \mathbb{R}^2$, $\Phi(x,t) := x_1 + x_2 t$, $t \in \mathbb{R}$, defines a line.
- determine $x = (x_1 \; x_2)^T$ as solution of (LLS)

$$\min_{x \in \mathbb{R}^2} \sum_{i=1}^{4} \|\Phi(x,t_i) - y_i\|^2.$$

$$\Phi(x,t_i) - y_i = 0,\ i = \overline{1,4} \quad \Leftrightarrow \quad \begin{cases} x_1 & -x_2 & = & 3 \\ x_1 & & = & 2 \\ x_1 & +x_2 & = & 0 \\ x_1 & +2x_2 & = & 4. \end{cases}$$

A simple LLS example ...

Let J matrix of system; x^* LLS solution iff $J^T J x^* = J^T y$.

$$\begin{pmatrix} 4 & 2 \\ 2 & 6 \end{pmatrix} \begin{pmatrix} x_1^* \\ x_2^* \end{pmatrix} = \begin{pmatrix} 9 \\ 5 \end{pmatrix},$$

$\Leftrightarrow \quad x^* = (2.2, 0.1)$ and $\Phi(x^*, t) = 2.2 + 0.1t$.

Nonlinear Least-Squares (NLS)

- $r : \mathbb{R}^n \to \mathbb{R}^m$ with $m \geq n$; r smooth.

  $$\min_{x \in \mathbb{R}^n} f(x) := \frac{1}{2} \sum_{j=1}^{m} [r_j(x)]^2 = \frac{1}{2} \|r(x)\|^2. \qquad \text{(NLS)}$$

- $r(x^*) = 0$: zero-residual pb.; $r(x^*) \neq 0$: nonzero-residual pb.

Nonlinear Least-Squares (NLS)

- $r : \mathbb{R}^n \to \mathbb{R}^m$ with $m \geq n$; r smooth.

  $$\min_{x \in \mathbb{R}^n} f(x) := \frac{1}{2} \sum_{j=1}^m [r_j(x)]^2 = \frac{1}{2} \|r(x)\|^2. \qquad \text{(NLS)}$$

- $r(x^*) = 0$: zero-residual pb.; $r(x^*) \neq 0$: nonzero-residual pb.

- $\nabla f(x) = J(x)^T r(x)$, where $J(x)$ Jacobian of r at x:

  (NLS) and chain rule $\Rightarrow$ for $i \in \{1, \ldots, n\}$,

  $$\frac{\partial f}{\partial x_i}(x) = \sum_{j=1}^m r_j(x) \frac{\partial r_j}{\partial x_i}(x) = r(x)^T \begin{pmatrix} \frac{\partial r_1}{\partial x_i}(x) \\ \ldots \\ \frac{\partial r_m}{\partial x_i}(x) \end{pmatrix}.$$

  The formula follows by using that the vector $(\partial r_1/\partial x_i \ldots \partial r_m/\partial x_i)^T$ is the ith column of $J(x)$.

Nonlinear Least-Squares (NLS)

- $r : \mathbb{R}^n \to \mathbb{R}^m$ with $m \geq n$; r smooth.

  $\min_{x\in\mathbb{R}^n} f(x) := \frac{1}{2}\sum_{j=1}^m [r_j(x)]^2 = \frac{1}{2}\|r(x)\|^2.$ (NLS)

- $r(x^*) = 0$: zero-residual pb.; $r(x^*) \neq 0$: nonzero-residual pb.

- $\nabla f(x) = J(x)^T r(x)$, where $J(x)$ Jacobian of r at x:

  (NLS) and chain rule $\Rightarrow$ for $i \in \{1, \ldots, n\}$,

$$\frac{\partial f}{\partial x_i}(x) = \sum_{j=1}^m r_j(x)\frac{\partial r_j}{\partial x_i}(x) = r(x)^T \begin{pmatrix} \frac{\partial r_1}{\partial x_i}(x) \\ \ldots \\ \frac{\partial r_m}{\partial x_i}(x) \end{pmatrix}.$$

  The formula follows by using that the vector $(\partial r_1/\partial x_i \ldots \partial r_m/\partial x_i)^T$ is the ith column of $J(x)$.

- $\nabla^2 f(x) = J(x)^T J(x) + \sum_{j=1}^m r_j(x)\nabla^2 r_j(x)$.

Gauss-Newton method for nonlinear least-squares

$\nabla^2 f(x) = J(x)^T J(x) + \sum_{j=1}^m r_j(x) \nabla^2 r_j(x)$.

- $r_j(x^*) \approx 0$ or $\nabla^2 r_j(x^*)$ small $\Longrightarrow$ $r_j(x) \nabla^2 r_j(x)$ small when x close to x^* $\Longrightarrow$ $\nabla^2 f(x) \approx J(x)^T J(x) := \widetilde{\nabla^2} f(x)$.

Gauss-Newton method for nonlinear least-squares

$\nabla^2 f(x) = J(x)^T J(x) + \sum_{j=1}^m r_j(x) \nabla^2 r_j(x)$.

- $r_j(x^*) \approx 0$ or $\nabla^2 r_j(x^*)$ small $\Longrightarrow$ $r_j(x) \nabla^2 r_j(x)$ small when x close to x^* $\Longrightarrow$ $\nabla^2 f(x) \approx J(x)^T J(x) := \widetilde{\nabla^2} f(x)$.
- Gauss-Newton (GN) direction:

$$\widetilde{\nabla^2} f(x^k) s^k = -\nabla f(x^k) \Longleftrightarrow J(x^k)^T J(x^k) s^k = -J(x^k)^T r(x^k),$$

Gauss-Newton method for nonlinear least-squares

$\nabla^2 f(x) = J(x)^T J(x) + \sum_{j=1}^m r_j(x)\nabla^2 r_j(x)$.

- $r_j(x^*) \approx 0$ or $\nabla^2 r_j(x^*)$ small $\Longrightarrow$ $r_j(x)\nabla^2 r_j(x)$ small when x close to x^* $\Longrightarrow$ $\nabla^2 f(x) \approx J(x)^T J(x) := \widetilde{\nabla^2} f(x)$.
- Gauss-Newton (GN) direction:

$$\widetilde{\nabla^2} f(x^k)s^k = -\nabla f(x^k) \Longleftrightarrow J(x^k)^T J(x^k)s^k = -J(x^k)^T r(x^k),$$

and so s^k solves the (LLS):

$$\min_{s\in\mathbb{R}^n} \frac{1}{2}\|J(x^k)s + r(x^k)\|^2$$

$$= \frac{1}{2}s^T J(x^k)^T J(x^k)s + s^T\nabla f(x^k) + \frac{1}{2}\|r(x^k)\|^2 := m_k(x^k + s).$$

$\longrightarrow$ f approximated by local convex quadratic model as $J(x^k)^T J(x^k)$ positive semi-definite for each k. Note that $\nabla_s m_k(x^k + s^k) = J(x^k)^T J(x^k)s^k + J(x^k)^T r(x^k) = 0$.

Gauss-Newton method for nonlinear least-squares

- GN direction: $J(x^k)^T J(x^k) s^k = -J(x^k)^T r(x^k)$
- s^k descent provided $J(x^k)$ full column rank! since if $J(x^k)$ full column rank $\Rightarrow J(x^k)^T J(x^k)$ positive definite.

Gauss-Newton method for nonlinear least-squares

- GN direction: $J(x^k)^T J(x^k) s^k = -J(x^k)^T r(x^k)$
- s^k descent provided $J(x^k)$ full column rank! since if $J(x^k)$ full column rank $\Rightarrow J(x^k)^T J(x^k)$ positive definite.

<u>Gauss-Newton (GN) method for nonlinear least-squares</u>: (with linesearch)

Choose $\epsilon > 0$ and $x^0 \in \mathbb{R}^n$.
While $\|\nabla f(x^k)\| > \epsilon$, REPEAT:
- solve the linear system $\widetilde{\nabla^2 f}(x^k) s^k = -\nabla f(x^k)$.
- set $x^{k+1} = x^k + \alpha^k s^k$, with $\alpha^k \in (0,1]$; $k := k+1$.

END.

- for example, calculate α^k by bArmijo linesearch, with $\alpha_{(0)} = 1$ and $\beta \leq 0.5$.
- GN method is a GLM if $J(x^k)$ is full column rank.

Convergence properties of Gauss-Newton method

- $\nabla f(x) = 0$ may not imply $r(x) = 0$

- (global convergence) $J(x^k)$ uniformly full-rank for all x^k (and ∇f Lips cont; see Th 4) $\implies$
$\|\nabla f(x^k)\| = \|J(x^k)^T r(x^k)\| \to 0$, $k \to \infty$.

- (local convergence) if $r(x^*) = 0$, $J(x^*)$ full-rank, $\alpha^k = 1$ for all k (+ conds in Th 7) $\implies$ $x^k \to x^*$ quadratically.

Convergence properties of Gauss-Newton method

- $\nabla f(x) = 0$ may not imply $r(x) = 0$

- (global convergence) $J(x^k)$ uniformly full-rank for all x^k (and ∇f Lips cont; see Th 4) $\implies$ $\|\nabla f(x^k)\| = \|J(x^k)^T r(x^k)\| \to 0$, $k \to \infty$.

- (local convergence) if $r(x^*) = 0$, $J(x^*)$ full-rank, $\alpha^k = 1$ for all k (+ conds in Th 7) $\implies$ $x^k \to x^*$ quadratically.

Gauss-Newton vs. Newton method:

- computational cost per iteration: N > GN.
- N direction may be ascent.
- only linear rate for GN when $r(x^*) \neq 0$.
- N & GN mthds unreliable without a linesearch (or other safeguards). Use bArmijo linesearch for example.

Gauss-Newton vs. Newton: an example

- $r : \mathbb{R} \to \mathbb{R}^2$; $r(x) := (x + 1 \quad 0.1x^2 + x - 1)^T$
- $r(x^*) = (1,\ -1)^T \neq 0 \ \longrightarrow$ nonzero residuals problem: only linear convergence asymptotically for GN.

	1	2	3	4	5	6
N	1.0	0.14	0.003	$1.5 \cdot 10^{-6}$	$4.3 \cdot 10^{-13}$	$3.1 \cdot 10^{-26}$
GN	1.0	0.13	0.014	0.0014	0.00014	0.000014