6 Nonlinear equations and Newton's method

goal: determine zero $\mathbf{x}^*$ of $\mathbf{f}(\mathbf{x}^*) = 0$
Since there are typically no exact solution formulas, the zero $\mathbf{x}^*$ is approximated by iterates $\mathbf{x}_n$ with $\lim_{n\to\infty} \mathbf{x}_n = \mathbf{x}^*$. The most common form is that of a *fixed point iteration*

$$\mathbf{x}_{n+1} = \Phi(\mathbf{x}_n) \tag{6.1}$$

with an initial guess $\mathbf{x}_0$ that is taken sufficiently close to $\mathbf{x}^*$. Thus, the iterative method is described by the function Φ.

Exercise 6.1 *Show: If $\mathbf{x}_n \to \mathbf{x}^*$ then $\mathbf{x}^*$ is a* fixed point *of Φ, i.e., $\mathbf{x}^* = \Phi(\mathbf{x}^*)$ (assumption: Φ is continuous at $\mathbf{x}^*$).* ∎

6.1 Newton's method in 1D

goal: Find zero x^* of $f(x^*) = 0$
Idea: *linearize* f at the current iterate x_n and find zero of the linearization.
procedure:

1. x_n = current iterate
2. $L(x) := f(x_n) + f'(x_n)(x - x_n)$ ⟦ linearization is the tangent at x_n, i.e., the Taylor expansion up to the linear term ⟧
3. x_{n+1} := zero of L, i.e.,
$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \tag{6.2}$$

We recognize that the 1D-Newton method (6.2) has the form $x_{n+1} = \Phi^{Newton}(x_n)$ of a fixed point iteration with Φ^{Newton} given by

$$\Phi^{Newton}(x) = x - \frac{f(x)}{f'(x)}. \tag{6.3}$$

finis 9.DS

Example 6.2 slide 1
$x^* = \sqrt{a}$ is the zero of $f(x) = x^2 - a$. With $f'(x) = 2x$, Newton's method is

$$x_{n+1} = \Phi^{Newton}(x_n) = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - \frac{x_n^2 - a}{2x_n}.$$

The rapid convergence of the method is visible in Fig. 6.1 for the choice $a = 2$ and initial value $x_0 = 2$. In fact, we observe so-called quadratic convergence *in that the error behaves like $|x^* - x_{n+1}| \approx C|x^* - x_n|^2$ for some $C > 0$.* ∎

	Newton iterates ($x_0 = 2$)	error
x_1	1.5	8.578643762690485_{-2}
x_2	**1.41**6666666666667	2.453104293571595_{-3}
x_3	**1.41421**5686274510	2.1239014147411694_{-6}
x_4	**1.41421356237**4690	1.5947243525715749_{-12}
exact:	1.414213562373095	

Figure 6.1: Newton's method for computing $\sqrt{2}$ (cf. Example 6.2)

6.2 convergence of fixed point iterations

The key property that ensures convergence of the fixed point iteration (6.1) is that Φ is a *contraction*:

Definition 6.3 *The function $\Phi : \mathbb{R}^d \to \mathbb{R}^d$ is a contraction (with respect to the norm $\|\cdot\|$) near the point $\mathbf{x}^*$ if there are $q \in (0,1)$ and $\varepsilon > 0$ such that*

$$\|\Phi(\mathbf{x}) - \Phi(\mathbf{y})\| \le q\|\mathbf{x} - \mathbf{y}\| \qquad \forall \mathbf{x}, \mathbf{y} \in B_\varepsilon(\mathbf{x}^*). \tag{6.4}$$

Exercise 6.4 *Consider the case $d = 1$. Show: If $\Phi \in C^1$ and $|\Phi'(x^*)| < 1$ near a point x^*, then Φ is a contraction near x^*.* ∎

The following result shows that the contraction property implies convergence of the fixed point iteration (6.1) if the initial value $\mathbf{x}_0$ is sufficiently close to the fixed point $\mathbf{x}^*$.

Theorem 6.5 *Let Φ be a contraction with contraction constant $q \in (0,1)$ near the fixed point $\mathbf{x}^* = \Phi(\mathbf{x}^*)$. Then there is $\varepsilon > 0$ such that for $\mathbf{x}_0 \in B_\varepsilon(\mathbf{x}^*)$ the iterates $\mathbf{x}_n$ given by (6.1) converge to $\mathbf{x}^*$. Moreover,*

$$\|\mathbf{x}^* - \mathbf{x}_{n+1}\| \le q\|\mathbf{x}^* - \mathbf{x}_n\| \qquad \forall n \in \mathbb{N}_0. \tag{6.5}$$

Proof: Let $\varepsilon > 0$ be given by Def. 6.3 and $x_n \in B_\varepsilon(x^*)$. Then:

$$\|x^* - x_{n+1}\| = \|x^* - \Phi(x_n)\| \overset{x^* \text{ fixed pt}}{=} \|\Phi(x^*) - \Phi(x_n)\| \overset{\text{contraction property}}{\le} q\|x^* - x_n\|.$$

Hence, if $x_0 \in B_\varepsilon(x^*)$, then by induction all iterates $x_n \in B_\varepsilon(x^*)$ and $\|x^* - x_n\| \to 0$. □

Exercise 6.4 gives an easy condition (in the scalar case $d = 1$) when the iteration (6.1) converges:

Exercise 6.6 *Let $d = 1$ and $\Phi \in C^1$ satisfy $|\Phi'(x^*)| < 1$ at the fixed point x^* of Φ. Then the iterates x_n given by (6.1) converge to x^* provided the initial value x_0 is sufficiently close to x^*.*
Remark: *The vector-valued analog is as follows: The derivative Φ' is a $d \times d$ matrix and if there is a norm $\|\cdot\|$ such that $\|\Phi'(\mathbf{x}^*)\| < 1$ at a fixed point $\mathbf{x}^*$ of Φ, then Φ is a contraction near $\mathbf{x}^*$.* ∎

Example 6.7 **slide 31**
We seek a solution of the nonlinear equation

$$2 - x^2 - e^x = 0. \tag{6.6}$$

n	$x_{n+1} = \Phi_1(x_n)$	$x_{n+1} = \Phi_2(x_n)$
0	0.592687716508341	0.559615787935423
1	0.437214425050104	0.522851128605001
2	0.672020792350124	0.546169619063046
3	0.204473907097276	0.531627015197373
4	0.879272743474883	0.540795632739194
5	stop: $(2 - e^{0.87} < 0)$	0.535053787215218
6		0.538664955236433
7		0.536399837485597
8		0.537823020842571
9		0.536929765486145

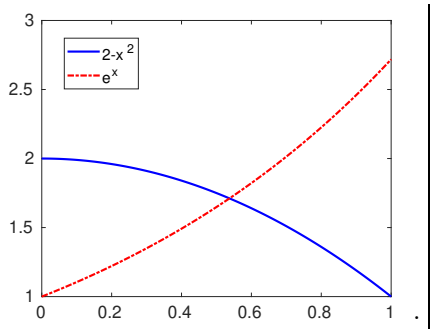

Table 6.1: Left: fixed point iteration of Example 6.7. Right: $x \mapsto e^x$ and $x \mapsto 2 - x^2$

Graphical considerations show that there is exactly one positive solution $x^* \approx 0.5$. For $x > 0$ equation (6.6) can be converted to a fixed point form in several ways:

$$x = \sqrt{2 - e^x} =: \Phi_1(x), \qquad x = \ln(2 - x^2) =: \Phi_2(x), \tag{6.7}$$

The fixed point iterations based on Φ_1 and Φ_2 behave differently when initialized with $x_0 = 0.5$ as is visible in Table 6.1: Whereas the iteration $x_{n+1} = \Phi_2(x_n)$ converges to the correct value $x^* = 0.5372744491738\ldots$ the iteration $x_{n+1} = \Phi_1(x_n)$ does not converge. The reason is that $|\Phi_1'(x^*)| \approx |-1.59| > 1$ whereas $|\Phi_2'(x^*)| \approx 0.31 < 1$. ∎

Theorem 6.5 shows that if Φ is a contraction, then one has *linear* convergence, i.e., the error decreases by a factor $q \in (0, 1)$ in each step. A special situation arises if $\Phi'(x^*) = 0$. Then faster convergence is possible:

Theorem 6.8 *Let $d = 1$ and $\Phi \in C^p(\mathbb{R}^d)$, $p \geq 2$. Assume $x^* = \Phi(x^*)$ and $0 = \Phi^j(x^*)$ for $j = 1, \ldots, p-1$. Then there are C, $\varepsilon > 0$ such that for $x_0 \in B_\varepsilon(x^*)$ the iterates x_n given by (6.1) converge to x^* and*

$$|x^* - x_{n+1}| \leq C|x^* - x_n|^p \qquad \forall n \in \mathbb{N}_0.$$

Proof: By Theorem 6.5 we already know that the iterates converge to x^* if ε is sufficiently small. For the estimate, we modify the proof of Theorem 6.5. By Taylor expansion around x^* we have

$$\begin{aligned} |x^* - x_{n+1}| = |\Phi(x^*) - \Phi(x_n)| = |\Phi(x^*) - \Phi(x_n)| &= \left| \frac{1}{(p-1)!} \int_{x^*}^{x_n} (x_n - t)^{p-1} \Phi^{(p)}(t)\, dt \right| \\ &\leq \frac{\|\Phi^{(p)}\|_{\infty, B_\varepsilon(x^*)}}{(p-1)!} |x^* - x_n|^p. \end{aligned}$$

□

In the setting of Theorem 6.8, we say that the iteration converges with *order* p. In particular, for $p = 2$ the method converges *quadratically*. Example 6.2 shows that the Newton method applied to the problem $f(x) = x^2 - a = 0$ convergence quadratically. This is typical of the Newton method:

Corollary 6.9 *Let $d = 1$ and $f \in C^2$. Assume $f(x^*) = 0$ and $f'(x^*) \neq 0$. Then Newton's method converges quadratically. That is, there are constants $C, \varepsilon > 0$ such that if $|x^* - x_0| \leq \varepsilon$ then the sequence $(x_n)_n$ converges to x^* and*

$$|x^* - x_{n+1}| \leq C|x^* - x_n|^2 \qquad \forall n.$$

Proof: One computes (exercise!) $\frac{d\Phi^{Newton}}{dx}(x^*) = 0$. Hence, Theorem 6.8 implies (at least) quadratic convergence. □

The quadratric convergence asserted in Cor. 6.9 requires $f'(x^*) \neq 0$. This is not an artefact of the proof:

Exercise 6.10 *Apply Newton's method to find the zero of $f(x) = x^2$. Show that Newton's method converges only linearly.* ∎

6.3 Newton's method in higher dimensions

Idee: as in 1D: linearize (= Taylor expansion up to linear terms) and find zero of linearization procedure:

- in $\mathbb{R}^n$: $\mathbf{x}_n$ = current iterate
- linearization $L(\mathbf{x}) := \mathbf{f}(\mathbf{x}_n) + \mathbf{f}'(\mathbf{x}_n)(\mathbf{x} - \mathbf{x}_n)$ = linearization of $\mathbf{f}$ at $\mathbf{x}_n$, where

$$\mathbf{f}'(\mathbf{x}) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(\mathbf{x}) & \frac{\partial f_1}{\partial x_2}(\mathbf{x}) & \cdots & \frac{\partial f_1}{\partial x_n}(\mathbf{x}) \\ \frac{\partial f_2}{\partial x_1}(\mathbf{x}) & \frac{\partial f_2}{\partial x_2}(\mathbf{x}) & \cdots & \frac{\partial f_2}{\partial x_n}(\mathbf{x}) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1}(\mathbf{x}) & \frac{\partial f_n}{\partial x_2}(\mathbf{x}) & \cdots & \frac{\partial f_n}{\partial x_n}(\mathbf{x}) \end{pmatrix}$$

- determine $\mathbf{x}_{n+1}$ as the zero of L, i.e.,

$$\mathbf{x}_{n+1} := \mathbf{x}_n - \Big(\mathbf{f}'(\mathbf{x}_n)\Big)^{-1}\mathbf{f}(\mathbf{x}_n).$$

That is, the iteration function Φ is

$$\Phi^{\text{Newton}}(\mathbf{x}) = \mathbf{x} - \Big(\mathbf{f}'(\mathbf{x})\Big)^{-1}\mathbf{f}(\mathbf{x}) \tag{6.8}$$

The convergence of the method is analogous to the 1D situation:

Theorem 6.11 *Let $\mathbf{f} \in C^2(B_\delta(\mathbf{x}^*))$ for some $\delta > 0$. Assume $\mathbf{f}(x^*) = 0$ and $\mathbf{f}'(x^*)$ is an invertible matrix. Then there exist $\varepsilon > 0$ and $C > 0$ such that if $\mathbf{x}_0 \in B_\varepsilon(\mathbf{x}^*)$, then all iterates $\mathbf{x}_n$ are in $B_\varepsilon(\mathbf{x}^*)$, one has convergence $\mathbf{x}_n \to \mathbf{x}^*$, and*

$$\|\mathbf{x}^* - \mathbf{x}_{n+1}\| \leq C\|\mathbf{x}^* - \mathbf{x}_n\|^2 \qquad \forall n.$$

Theorem 6.11 states *quadratic* convergence of Newton's method (provided the starting value is sufficiently close to $\mathbf{x}^*$) provided $\mathbf{f}'(\mathbf{x}^*)$ is invertible.

Remark 6.12 *In practice the Newton step is* not *realized by computing the inverse* $(\mathbf{f}')^{-1}$ *but by solving a linear system:*

1. *compute* $\mathbf{f}'(\mathbf{x}_n)$ *and the residual* $\mathbf{f}(\mathbf{x}_n)$
2. *compute the correction by solving the linear system* $\mathbf{f}'(\mathbf{x}_n)\delta = \mathbf{f}(\mathbf{x}_n)$
3. *perform the update* $\mathbf{x}_{n+1} := \mathbf{x}_n - \delta$ ∎

Remark 6.13 *The residual* $\mathbf{f}(\mathbf{x}_n)$ *is some measure for the error* $\mathbf{x}^* - \mathbf{x}_n$*. If* $\mathbf{f}'(\mathbf{x}^*)$ *is invertible, then for* $\mathbf{x}_n$ *sufficiently close to* $\mathbf{x}^*$*, Taylor expansion indicates*

$$\mathbf{f}(\mathbf{x}_n) = \mathbf{f}(\mathbf{x}_n) - \mathbf{f}(\mathbf{x}^*) \approx \mathbf{f}'(\mathbf{x}^*)(\mathbf{x}_n - \mathbf{x}^*)$$

so that we can expect

$$\|(\mathbf{f}'(\mathbf{x}^*))^{-1}\mathbf{f}(\mathbf{x}_n)\| \approx \|\mathbf{x}^* - \mathbf{x}_n\|. \tag{6.9}$$

The residual $\mathbf{f}(\mathbf{x}_n)$ *still is a measure for the error, however, only up to constant depending on* $\mathbf{f}'(\mathbf{x}^*)$*:*

$$\|\mathbf{f}(\mathbf{x}_n)\| \leq \|\mathbf{f}'(\mathbf{x}^*)\|\|\mathbf{x}^* - \mathbf{x}_n\| + O(\|\mathbf{x}^* - \mathbf{x}_n\|^2), \tag{6.10}$$

$$\|\mathbf{x}^* - \mathbf{x}_n\| \leq \|(\mathbf{f}'(\mathbf{x}^*))^{-1}\|\|\mathbf{f}(\mathbf{x}_n)\| + O(\|\mathbf{x}^* - \mathbf{x}_n\|^2). \tag{6.11}$$

∎

6.4 implementation aspects of Newton methods

stopping criteria

1. $\mathbf{x}_n$ close to $\mathbf{x}^*$ $\Rightarrow$ quadratic convergence $\Rightarrow$ $\|\mathbf{x}_{n+1} - \mathbf{x}_n\|$ is a good estimate for $\|\mathbf{x}_n - \mathbf{x}^*\|$:

$$\|\mathbf{x}_n - \mathbf{x}^*\| \;\leq\; \|\mathbf{x}_n - \mathbf{x}_{n+1}\| + \underbrace{\|\mathbf{x}_{n+1} - \mathbf{x}^*\|}_{\substack{\leq\ c\,\|x_n - x^*\|^2 \\ \ll\ \|x_n - x^*\|}}$$

   $\Rightarrow$ If each Newton step is cheap, then the stopping criterion is

$$\|\mathbf{x}_{n+1} - \mathbf{x}_n\| \;\leq\; \text{given tolerance}$$

2. If Newton steps are expensive (e.g., for large systems of equations) then one can approximate $\|\mathbf{x}_{n+1} - \mathbf{x}_n\|$ as follows:

$$\|\mathbf{x}_{n+1} - \mathbf{x}_n\| = \|\Big(\mathbf{f}'(\mathbf{x}_n)\Big)^{-1}\mathbf{f}(\mathbf{x}_n)\| \approx \|\Big(\mathbf{f}'(\mathbf{x}_{n-1})\Big)^{-1}\mathbf{f}(\mathbf{x}_n)\|$$

   This expression is computable since $\mathbf{f}'(\mathbf{x}_{n-1})$ has been determined for the computation of $\mathbf{x}_n$. If an *LU*-factorization of $\mathbf{f}'(\mathbf{x}_{n-1})$ is available, then the computation of $\mathbf{f}'^{-1}(\mathbf{x}_{n-1})\mathbf{f}(x_n)$ is comparatively cheap.

computing $\mathbf{f}'(\mathbf{x}_n)$:

1. problem: often $\mathbf{f}'$ is not explicitly available but only $\mathbf{f}$ (e.g., if $\mathbf{f}$ is available as a C-code). Then $\mathbf{f}'(\mathbf{x}_n)$ can be approximated by difference quotients.

2. problem: Computing $\mathbf{f}'(\mathbf{x}_n)$ can be expensive (for example: for large d the $d \times d$-matrix $\mathbf{f}'$ has many entries) Then one often uses the *simplified Newton method*

$$\mathbf{x}_{n+1} \;=\; \mathbf{x}_n - \Big(\mathbf{f}'(\mathbf{x}_0)\Big)^{-1}\mathbf{f}(\mathbf{x}_n)$$

   Since one uses the same, fixed derivative (at the point $\mathbf{x}_0$), the method is only linearly convergent.

Exercise 6.14 *Let $\mathbf{B} \in \mathbb{R}^{d\times d}$ be invertible, $\widetilde{\mathbf{f}}(x) := \mathbf{B}\mathbf{f}(x)$. Then: $\mathbf{f}(x^*) = 0$ if and only if $\widetilde{\mathbf{f}}(\mathbf{x}^*) = 0$, and the Newton iterates for computing the zeros of $\mathbf{f}$ and of $\widetilde{\mathbf{f}}$ coincide.* ∎

6.5 damped and globalized Newton methods

Problem: Newton's method converges only *locally*, i.e., if $\mathbf{x}_0$ is sufficiently close to the zero $\mathbf{x}^*$.
goal: methods that cope (reasonably well) with poor initial values $\mathbf{x}_0$.

6.5.1 damped Newton method

Problem: quite often, the Newton steps $\mathbf{x}_{n+1} - \mathbf{x}_n$ are too large for convergence.
slide 32
The way to cope with this problem is the *damped Newton method* where, for chosen $\lambda_n \in (0, 1]$, the update is

$$\mathbf{x}_{n+1} := \mathbf{x}_n - \lambda_n(\mathbf{f}'(\mathbf{x}_n))^{-1}\mathbf{f}(\mathbf{x}_n) \tag{6.12}$$

For suitably small λ_n, this method converges for a larger regime of initial values $\mathbf{x}_0$. However, the convergence is only linear. One is therefore interested in methods where the parameters λ_n are selected adaptively and in particular $\lambda_n = 1$ for the iterates sufficiently close to $\mathbf{x}^*$ so as to obtain the quadratic convergence of the Newton method. An algorithm that realizes this is given in Alg. 6.16. finis 10.DS

6.5.2 a digression: descent methods

Let $g : \mathbb{R}^d \to \mathbb{R}$ be a given function. Minima of g can be sought with *descent methods*, which are iterative methods that determine the next iterate $\mathbf{x}_{n+1}$ from a current iterate $\mathbf{x}_n$ as follows:

1. select a *search direction* $\mathbf{d}_n$

2. select a *step length* λ_n such that for $\mathbf{x}_{n+1} := \mathbf{x}_n + \lambda_n\mathbf{d}_n$ one has $g(\mathbf{x}_{n+1}) < g(\mathbf{x}_n)$.

The search direction $\mathbf{d}_n$ is called a *descent direction* if the 1D function $\widetilde{g}(t) := g(\mathbf{x}_n + t\mathbf{d}_n)$ satisfies $\widetilde{g}'(0) < 0$, i.e., is decreasing for small $t > 0$. Put differently, $\mathbf{d}_n$ needs to satisfy

$$\nabla g(\mathbf{x}_n) \cdot \mathbf{d}_n < 0.$$

The method of *steepest descent* corresponds to the choice $\mathbf{d}_n = -\nabla g(\mathbf{x}_n)$.
The second ingredient of a descent method is the choice of the step length λ_n. The "greedy" approach would be to select λ_n such that

$$\min_{t>0} \widetilde{g}(t) = \widetilde{g}(\lambda_n).$$

Since this "line search" is still quite expensive, several other options are common that realize the idea of selecting a step size with "sufficient" descent. We mention the so-called *Armijo*-rule: Given $\sigma \in (0,1)$ and $q \in (0,1)$ one select the *largest* step length of the form q^k, $k = 0, 1, \ldots$, such that

$$\widetilde{g}(q^k) < \widetilde{g}(0) + \sigma \widetilde{g}'(0) q^k,$$

or, written in terms of g

$$g(\mathbf{x}_n + q^k \mathbf{d}_n) < g(\mathbf{x}_n) + \sigma(\nabla g(\mathbf{x}) \cdot \mathbf{d}_n) q^k. \tag{6.13}$$

This can be realized by trying the cases $k = 0$, 1, etc. in turn until (6.13) is satisfied. This step length choice can be interpreted as trying to make fairly large steps with a reasonable reduction of the functional g.

6.5.3 globalized Newton method as a descent method

observe: zeros of $\mathbf{f}$ are minima of $\mathbf{x} \mapsto \|\mathbf{f}(\mathbf{x})\|_2^2 = \mathbf{f}(\mathbf{x})^\top \mathbf{f}(\mathbf{x})$.
idea: View the damped Newton method as a descent method with search direction $\mathbf{d}_n := -(\mathbf{f}'(\mathbf{x}_n))^{-1}\mathbf{f}(\mathbf{x}_n)$ and step length parameter λ_n.
For this idea to work, we need to know that the so-called *Newton direction*

$$\mathbf{d}_n := -(\mathbf{f}(\mathbf{x}_n))^{-1}\mathbf{f}(\mathbf{x}_n) \tag{6.14}$$

is a descent direction for $g(\mathbf{x}) := \|\mathbf{f}(\mathbf{x})\|_2^2$.

Lemma 6.15 *Let $\mathbf{f} \in C^2(\mathbb{R}^d)$. Then: For given $\mathbf{x}$ and $\mathbf{d} := -(\mathbf{f}(\mathbf{x}))^{-1}\mathbf{f}(\mathbf{x})$ the function $\widetilde{g}(\lambda) := g(\mathbf{x} + \lambda\mathbf{d})$ has the Taylor expansion $\widetilde{g}(\lambda) = g(\mathbf{x}) - 2\lambda g(\mathbf{x}) + O(\lambda^2)$ for small λ.*

Proof: For notational simplicity we consider the case $d = 1$. Then $g(x) = f^2(x)$ and $\widetilde{g}(\lambda) = f^2(x + \lambda\mathfrak{d}(x))$ with $\mathfrak{d}(x) = -(f'(x))^{-1}f(x)$. Then by Taylor, we have for small λ

$$\begin{aligned}\widetilde{g}(\lambda) &= \widetilde{g}(0) + \lambda g'(0) + O(\lambda^2) = f^2(x) + \lambda 2 f(x) f'(x) \mathfrak{d}(x) + O(\lambda^2) \\ &= f^2(x) - \lambda 2 f(x) f'(x) \frac{f(x)}{f'(x)} + O(\lambda^2) = f^2(x) - 2\lambda f^2(x) + O(\lambda^2).\end{aligned}$$

□

Lemma 6.15 shows that the Newton direction is a descent method and that, for λ sufficiently small, we may achieve a descent

$$g(\mathbf{x}_n + \lambda_n \mathbf{d}_n) - g(\mathbf{x}_n) \approx 2\lambda_n g(\mathbf{x}_n) \tag{6.15}$$

$\Rightarrow$ sensible goals for selecting λ are:

- if $\mathbf{x}_n$ is close to $\mathbf{x}^*$ then select $\lambda = 1$ (so that actual Newton steps with quadratic convergence are performed!). We note that the quadratic convergence implies a descent of almost $\|\mathbf{f}(\mathbf{x}_n)\|^2$: for $\mathbf{x}_n$ near $\mathbf{x}^*$ we have

$$\|\mathbf{f}(\mathbf{x}_{n+1})\|^2 \overset{(6.10)}{\leq} C_1\|\mathbf{x}^* - \mathbf{x}_{n+1}\|_2^2 \overset{\text{quad. conv.}}{\leq} C_2\|\mathbf{x}^* - \mathbf{x}_n\|_2^4 \overset{(6.11)}{\leq} C_3\|\mathbf{f}(\mathbf{x}_n)\|_2^4.$$

  In other words: for actual Newton steps, we expect $\|\mathbf{f}(\mathbf{x}_n)\|_2^2 - \|\mathbf{f}(\mathbf{x}_{n+1})\|_2^2 \approx \|\mathbf{f}(\mathbf{x}_n)\|_2^2$.

- If $\mathbf{x}_n$ is far from $\mathbf{x}^*$, then select λ small but s.t. the descent is $\|\mathbf{f}(\mathbf{x}_n)\|_2^2 - \|\mathbf{f}\big(\mathbf{x}_n + \lambda\,\mathbf{d}(x_n)\big)\|_2^2$ is large. By (6.15), a descent $\|\mathbf{f}(\mathbf{x}_n + \lambda\mathbf{d}_n)\|_2^2 - \|\mathbf{f}(\mathbf{x}_n)\|_2^2 \approx 2\lambda_n\|\mathbf{f}(\mathbf{x}_n)\|_2^2$ is possible for small λ_n

We wish to require the descent to be compatible with Newton steps. Therefore, we require a descent of $\approx \lambda_n\|\mathbf{f}(\mathbf{x}_n)\|_2^2$ rather than the "greedy" $2\lambda_n\|\mathbf{f}(\mathbf{x}_n)\|_2^2$. This is what we enforce in the following algorithm:

Algorithm 6.16 *Input: initial value* $\mathbf{x}_0$, *parameter* $\mu, q \in (0,1)$

$\lambda_0 := 1$
$n := 0$
while *(stopping criterion not satisfied)* ***do***
  $\mathbf{d}_n := -\Big(\mathbf{f}'(\mathbf{x}_n)\Big)^{-1}\mathbf{f}(\mathbf{x}_n)$
  while $\Big(\|\mathbf{f}(\mathbf{x}_n)\|_2^2 - \|\mathbf{f}(\mathbf{x}_n + \lambda_n\,\mathbf{d}_n)\|_2^2 < \mu\,\lambda_n\|\mathbf{f}(\mathbf{x}_n)\|_2^2\Big)$ ***do*** *% reduce λ until sufficient amount of descent*
    $\lambda_n := \lambda_n \cdot q$
  end while
  $\mathbf{x}_{n+1} := \mathbf{x}_n + \lambda_n\,\mathbf{d}_n$
  $\lambda_{n+1} := \min\Big(1, \frac{\lambda_n}{q}\Big)$ *% try a little large λ next time*
end while

Remark 6.17 *The $\|\cdot\|_2$-norm was selected for convenience of exposition. Especially for large systems, other norms may be more appropriate.* ∎

6.6 Quasi-Newton methods (CSE)

Problem: often, the computation of $\mathbf{f}'$ is expensive.
simple solution: simplified Newton method where $\mathbf{f}'(\mathbf{x}_n)$ is replaced with $\mathbf{f}'(\mathbf{x}_0)$. Downside: *linear convergence*
goal: methods that converge superlinearly but are cheaper than full Newton method

6.6.1 Broyden method

Setting: $\mathbf{f} \in C^1(\mathbb{R}^d;\mathbb{R}^d)$, $\mathbf{f}(x^*) = 0$, $\mathbf{f}'(x^*)$ invertible
Broyden methods are iterative methods of the form $\mathbf{x}_{n+1} = \mathbf{x}_n - \mathbf{H}_n^{-1}\mathbf{f}(\mathbf{x}_n)$ with suitable matrices $\mathbf{H}_n$.
idea of Broyden's method

- after computing $\mathbf{x}_{n+1}$ compute the next $\mathbf{H}_{n+1}$ from $\mathbf{H}_n$
- $\mathbf{H}_{n+1}$ is some kind of "approximation" to $\mathbf{f}'(\mathbf{x}_{n+1})$

Taylor yields $-\mathbf{f}(\mathbf{x}_{n+1}) + \mathbf{f}(\mathbf{x}_n) = \mathbf{f}'(\mathbf{x}_{n+1})(\mathbf{x}_n - \mathbf{x}_{n+1}) + O(\|\mathbf{x}_{n+1} - \mathbf{x}_n\|^2)$ so that we expect $\mathbf{f}'(\mathbf{x}_{n+1})(\mathbf{x}_{n+1} - \mathbf{x}_n) \approx \mathbf{f}(\mathbf{x}_{n+1}) - \mathbf{f}(\mathbf{x}_n)$ Hence, a resonable condition on $\mathbf{H}_{n+1}$ is the "secant condition"

$$\mathbf{H}_{n+1}(\mathbf{x}_{n+1} - \mathbf{x}_n) \overset{!}{=} \mathbf{f}(\mathbf{x}_{n+1}) - \mathbf{f}(\mathbf{x}_n) \tag{6.16}$$

Condition (6.16) does not fix $\mathbf{H}_{n+1}$ (unless $d = 1$). A reasonable further condition is that $\mathbf{H}_{n+1}$ does not deviate much from $\mathbf{H}_n$, i.e., that $\mathbf{H}_{n+1} - \mathbf{H}_n$ be small. This leads to the problem:

$$\text{Find } \mathbf{H}_{n+1} \text{ satisfying (6.16) s.t. } \|\mathbf{H}_{n+1} - \mathbf{H}_n\|_F = \min\{\|\mathbf{A} - \mathbf{H}_n\|_F \mid \mathbf{A}(\mathbf{x}_{n+1} - \mathbf{x}_n) = \mathbf{f}(\mathbf{x}_{n+1}) - \mathbf{f}(\mathbf{x}_n)\} \tag{6.17}$$

This constrained minimization problem has a unique solution:

$$\mathbf{H}_{n+1} = \mathbf{H}_n + \frac{1}{\|\mathbf{s}\|_2^2}(\mathbf{y} - \mathbf{H}_n\mathbf{s})\,\mathbf{s}^\top, \quad \mathbf{s} = \mathbf{x}_{n+1} - \mathbf{x}_n, \quad \mathbf{y} = \mathbf{f}(\mathbf{x}_{n+1}) - \mathbf{f}(\mathbf{x}_n). \tag{6.18}$$

The reason is the following, more general result:

Lemma 6.18 *Let $\mathbf{B} \in \mathbb{R}^{d\times d}$, $\mathbf{s}$, $\mathbf{y} \in \mathbb{R}^d$ with $\mathbf{s} \neq 0$. Then the matrix $\mathbf{B}_+ \in \mathbb{R}^{d\times d}$ given by*

$$\mathbf{B}_+ = \mathbf{B} + \frac{1}{\|\mathbf{s}\|_2^2}(\mathbf{y} - \mathbf{B}\mathbf{s})\mathbf{s}^\top \tag{6.19}$$

solves the following constrained minimization problem:

$$\textit{Find the minimizer } \mathbf{A} \textit{ of } \|\mathbf{A} - \mathbf{B}\|_F \textit{ under the constraint } \mathbf{A}\mathbf{s} = \mathbf{y} \tag{6.20}$$

Furthermore, the minimizer is unique.

Proof: We will only show that the given $\mathbf{B}_+$ solves the minimization problem. By construction, $\mathbf{B}_+\mathbf{s} = \mathbf{y}$. For arbitrary $\mathbf{A}$ with $\mathbf{A}\mathbf{s} = \mathbf{y}$, we compute

$$\|\mathbf{B}_+ - \mathbf{B}\|_F = \|\frac{1}{\|\mathbf{s}\|_2^2}(\mathbf{y} - \mathbf{B}\mathbf{s})\mathbf{s}^\top\|_F = \|\frac{1}{\|\mathbf{s}\|_2^2}(\mathbf{A}\mathbf{s} - \mathbf{B}\mathbf{s})\mathbf{s}^\top\|_F = \|(\mathbf{A} - \mathbf{B})\frac{\mathbf{s}^\top}{\|\mathbf{s}\|_2^2}\|_F$$
$$\overset{\|\mathbf{G}\mathbf{H}\|_F \le \|\mathbf{G}\|_F\|\mathbf{H}\|_2}{\le} \|\mathbf{A} - \mathbf{B}\|_F \underbrace{\|\frac{\mathbf{s}\mathbf{s}^\top}{\|\mathbf{s}\|_2^2}\|_2}_{=1 \text{ since } \mathbf{s}\mathbf{s}^\top \text{ is sym. with } d-1 \text{ EVs 0 and one EV 1}}$$

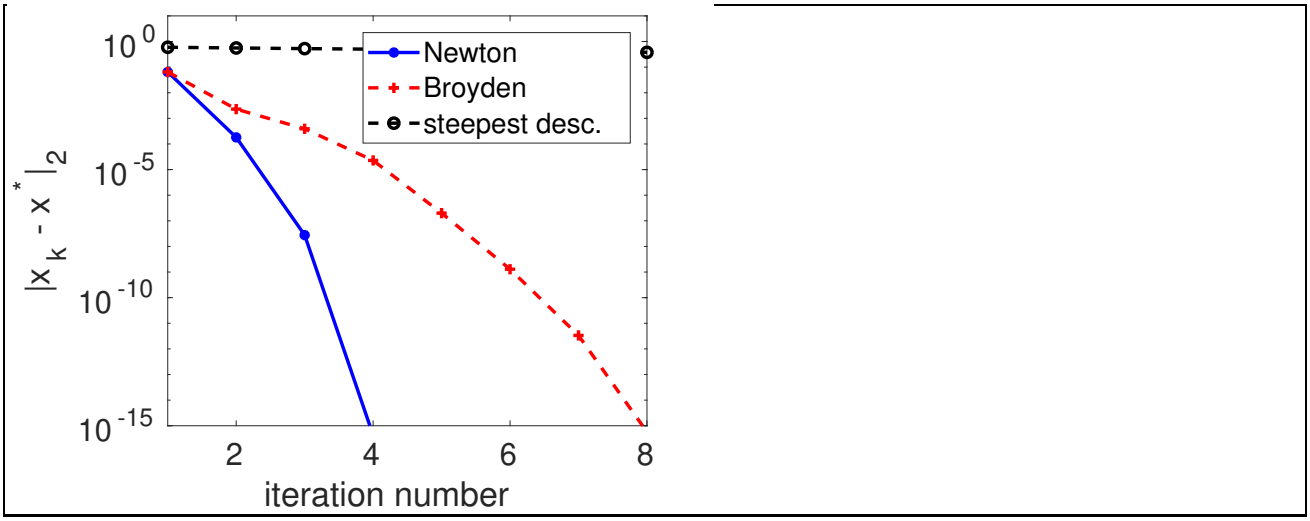

Figure 6.2: Comparison of Newton method, Broyden method, and gradient method (See Example 6.19).

□

The update formula (6.18) yields the following *Broyden method*:

1. given $\mathbf{H}_n$ compute $\mathbf{x}_{n+1} = \mathbf{x}_n - \mathbf{H}_n^{-1}\mathbf{f}(\mathbf{x}_n)$
2. compute $\mathbf{H}_{n+1}$ via (6.18).

Important features of this method are:

1. The method converges (locally) superlinearly, i.e., for some sequence $\varepsilon_n \to 0$ there holds

$$\|\mathbf{x}_{n+1} - \mathbf{x}_n\| \leq \varepsilon_n \|\mathbf{x}_n - \mathbf{x}_{n-1}\|$$

2. The Broyden updates are rank-1 updates. For rank-1 updates of matrices, the inverses can be computed fairly cheaply with the *Sherman-Morrison-Woodbury formula*, which asserts (exercise!) that for arbitrary invertible $\mathbf{A} \in \mathbb{R}^{d\times d}$ and vectors $\mathbf{u}$, $\mathbf{v}$ (with $\mathbf{v}^\top \mathbf{A}^{-1}\mathbf{u} \neq -1$) there holds

$$(\mathbf{A} + \mathbf{u}\mathbf{v}^\top)^{-1} = \mathbf{A}^{-1} - \frac{1}{1 + \mathbf{v}^\top \mathbf{A}^{-1}\mathbf{u}} \mathbf{A}^{-1}\mathbf{u}\mathbf{v}^\top \mathbf{A}^{-1}. \tag{6.21}$$

Example 6.19 **slide 32a**
We seek the zero $\mathbf{x}^* = (0,1)^T$ *of*

$$F(\mathbf{x}) = \begin{pmatrix} (x_1 + 3)(x_2^3 - 7) + 18 \\ \sin(x_2 e^{x_1} - 1) \end{pmatrix} = 0$$

with initial value $\mathbf{x}_0 = (-0.5, 1.4)^T$. *The classical Broyden method is started with* $\mathbf{H}_0 = F'(\mathbf{x}_0)$. *One observes in Fig. 6.2 in particular superlinear convergence of the Broyden method. For comparison purposes also the the gradient method for* $f(x) := \|F(x)\|_2^2$ *with* $\sigma = 0.9$ *and* $q = 0.5$ *(see Sec. 6.7.1) is shown.*

Remark 6.20 *There are many important variations of the Broyden method. Consider for example the case that Newton's method is applied to find the minimum of a function f (see Section 6.7.1). Then the Hessian of f is symmetric and — at least in the vicinity of the sought minimum — positive definite. One would like to make Broyden-like updates that preserve symmetric and positive definiteness. Such methods exist: see PSB ("Powell symmetric Broyden"), DFP ("Davidson-Fletcher-Powell"), BFGS ("Broyden-Fletcher-Goldfarb-Shanno").* ∎

Remark 6.21 *Just like globalized Newton methods, Broyden and Broyden-like methods are in practice combined with algorithms that select the step length. the* ∎

6.7 unconstrained minimization problems (CSE)

goal: minimize a function $f : \mathbb{R}^d \to \mathbb{R}$
This problem can be approached in several ways, for example:

1. The minimizer satisfies $\nabla f(\mathbf{x}^*) = 0$ so that a (globalized) Newton method could be used. We note that then the Hessian of f is required.

2. Descent method: These methods identify a descent direction for f (e.g., $\nabla f(\mathbf{x}_n)$) and then make a step that reduces f. These methods typically require only ∇f and are discussed in Sec. 6.7.1.

3. Trust region methods: these methods approximate f locally by a quadratic function that is minimized in a region where the quadratic approximation is deemed reliable. This is sketched in Sec. 6.7.3.

6.7.1 gradient methods

The simplest minimization strategy is the following iteration, starting with an initial point $\mathbf{x}_0$:

1. select a *search direction* $\mathbf{d}_n$ with $\nabla f(\mathbf{x}_n) \cdot \mathbf{d}_n < 0$

2. select a *step length* λ_n such that $f(\mathbf{x}_n + \lambda_n \mathbf{d}_n) < f(\mathbf{x}_n)$

Concerning the search direction $\mathbf{d}_n$, the simplest one is the negative gradient: $\mathbf{d}_n = -\nabla f(\mathbf{x}_n)$. This is called the *steepest descent direction.*
There are many choices for the step length λ_n. The "greedy" approach is to take λ_n as the minimizer of 1D optimization problem:

$$\text{minimize } t \mapsto \varphi(t) := f(\mathbf{x}_n + t\mathbf{d}_n). \tag{6.22}$$

Since this minimization problem is typically still difficult to solve various simplified versions are employed. A typical condition imposed on the step length λ_n is that each step make sufficient descent, namely,

$$f(\mathbf{x}_n + \lambda_n \mathbf{d}_n) < f(\mathbf{x}_n) - \lambda_n \sigma \nabla f(\mathbf{x}_n) \cdot \mathbf{d}_n \tag{6.23}$$

for some user chosen parameter σ. That is, the reduction in f should be proportional to the step size as well as the directional derivative $\nabla f(\mathbf{x}_n) \cdot \mathbf{d}_n$. On popular technique to ensure this is the *Armijo*-rule: Given $q \in (0, 1)$, one selects λ_n as the largest number of the form $\lambda_n = q^k$, $k \in \mathbb{N}_0$, such that the condition (6.23) is satisfied.

6.7.2 gradient method with quadratic cost function

We consider the special case of a quadratric function f:

$$f(\mathbf{x}) = \gamma + \mathbf{c}^\top \mathbf{x} + \frac{1}{2}\mathbf{x}^\top \mathbf{Q}\mathbf{x} \tag{6.24}$$

where $\gamma \in \mathbb{R}$, $\mathbf{c} \in \mathbb{R}^d$, $\mathbf{Q}$ is SPD. ⟦ note: in the vicinity of a minimum of f, one expects f to be close to a quadratic polynomial of this form by Taylor ⟧. We employ as the search direction $\mathbf{d}_n := -\nabla f(\mathbf{x}_n)$. Rather than using the Armijo rule, we use the minimum rule since the minimum can be computed: The minimum of $\varphi : t \mapsto f(\mathbf{x}_n + t\mathbf{d}_n)$ is explicitly given by

$$t = -\frac{f(\mathbf{x}_n) \cdot \mathbf{d}_n}{\mathbf{d}_n^\top \mathbf{Q}\mathbf{d}_n}$$

since

$$\begin{aligned}
\varphi(t) &= f(\mathbf{x}_n + t\mathbf{d}_n) = f(\mathbf{x}_n) + t\nabla f(\mathbf{x}_n) \cdot \mathbf{d}_n + \frac{1}{2}t^2 \mathbf{d}_n^\top \mathbf{Q}\mathbf{d}_n, \\
\varphi'(t) &= \nabla f(\mathbf{x}_n) \cdot \mathbf{d}_n + t\mathbf{d}_n^\top \mathbf{Q}\mathbf{d}_n;
\end{aligned}$$

therefore, one step of the gradient method is

$$\mathbf{x}_{n+1} = \mathbf{x}_n + t\mathbf{d}_n = \mathbf{x}_n - \frac{\nabla f(\mathbf{x}_n) \cdot \mathbf{d}_n}{\mathbf{d}_n^\top \mathbf{Q}\mathbf{d}_n}\mathbf{d}_n$$

The convergence can be estimate:

Lemma 6.22 *Let f be given by (6.24) with an SPD matrix $\mathbf{Q}$. Consider steepest descent, i.e., $\mathbf{d}_n := -\nabla f(\mathbf{x}_n)$. Then:*

$$\begin{aligned}
f(\mathbf{x}_{n+1}) - f(\mathbf{x}^*) &\le \left(\frac{\lambda_{max} - \lambda_{min}}{\lambda_{max} + \lambda_{min}}\right)^2 (f(\mathbf{x}_n) - f(\mathbf{x}^*)) = \left(\frac{\kappa - 1}{\kappa + 1}\right)^2 (f(\mathbf{x}_n) - f(\mathbf{x}^*)), \\
\|\mathbf{x}_{n+1} - \mathbf{x}^*\|_{\mathbf{Q}}^2 &\le \left(\frac{\lambda_{max} - \lambda_{min}}{\lambda_{max} + \lambda_{min}}\right)^2 \|\mathbf{x}_n - \mathbf{x}^*\|_{\mathbf{Q}}^2, = \left(\frac{\kappa - 1}{\kappa + 1}\right)^2 \|\mathbf{x}_n - \mathbf{x}^*\|_{\mathbf{Q}}^2,
\end{aligned}$$

where $\|\mathbf{z}\|_Q^2 = \mathbf{z}^\top \mathbf{Q}\mathbf{z}$ and $\kappa = \lambda_{max}/\lambda_{min}$ is the condition number of $\mathbf{Q}$.

Proof: Literature. □

Lemma 6.22 shows that the steepest descent method degrades if $\mathbf{Q}$ has widely differing eigenvalues (i.e., large condition number κ). This problem can be solved or at least mitigated by selecting the search directions in a different way. In fact, if one takes an SPD matrix $\mathbf{H}$ (as a "preconditioner") and considers as the search direction

$$\mathbf{d}_n = -\mathbf{H}\nabla f(\mathbf{x}_n)$$

then, one can show that

$$f(\mathbf{x}_{n+1}) - f(\mathbf{x}^*) \le \left(\frac{\lambda_{max}(\mathbf{H}^{-1}\mathbf{Q}) - \lambda_{min}(\mathbf{H}^{-1}\mathbf{Q})}{\lambda_{max}(\mathbf{H}^{-1}\mathbf{Q}) + \lambda_{min}(\mathbf{H}^{-1}\mathbf{Q})}\right)^2 (f(\mathbf{x}_n) - f(\mathbf{x}^*)),$$

so that the contraction factor can be much smaller than in the unpreconditioned case. The extreme case $\mathbf{H} = \mathbf{Q}$ leads to convergence in one step.

Remark 6.23 *The minimization of the quadratic function f can be done explicitly with solution $x^* = -\mathbf{Q}^{-1}\mathbf{c}$ so that a (steepest) descent method seems useless. Nevertheless, the discussion of quadratic functions f is of interest as it indicates weaknesses of the steepest descent methods for general f: one should expect slow convergence if, for example, the Hessian of f has a large condition number.* ∎

Returning to the quadratic problem, it is of interest to note that the minimum can also be found as the zero of the function $\mathbf{x} \mapsto \nabla f(\mathbf{x})$. This is a linear function. The Hessian of f is $\mathbf{H} = \mathbf{Q}$. Applying the Newton method yields convergence in one step. The Newton step is

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \mathbf{H}^{-1}\nabla f(\mathbf{x}_n).$$

This is precisely the preconditioned gradient method with the above identified optimal preconditioner $\mathbf{H} = \mathbf{Q}$.

6.7.3 trust region methods

starting point: many minimization techniques are based on "sequential quadratic programming", i.e., the function f is approximated locally by a quadratric "model" of the form

$$q_k(x) = f(x_k) + g_k \cdot (x - x_k) + \frac{1}{2}(x - x_k)^T B_k (x - x_k), \tag{6.25}$$

that is then minimized instead. Examples are:

- $g_k = \nabla f(x_k)$ and $B_k = \mathbf{H}(x_k)$, where $\mathbf{H}(x_k)$ is the Hessian of f at x_n: $\rightarrow$ Newton's method if $\mathbf{H}(x_k)$ SPD
- $g_k = \nabla f(x_k)$ and $B_k = \mathrm{Id}$: $\rightarrow$ gradient method (with step length $t_k = 1$)

Problems:

- the quadratic model is only valid in a small region near x_k. Too large steps of the minimization algorithm may lead to leaving the region of validity of the model.
- If B_k is not SPD, then the minimization problem is not meaningful.

In *trust region methods* the model q_k is not minimized over $\mathbb{R}^d$ but merely on a ball $B_{\Delta_k}(x_k)$ for given Δ_k:

$$\text{Minimize} \quad q_k(x) \quad \text{under the constraint } \|x_{k+1} - x_k\| \leq \Delta_k. \tag{6.26}$$

- (6.26) has a solution
- key ingredient of the algorithm is the steering of the Δ_k.
- in order to assess whether the quadratic model is "good", one defines

$$\rho_k := \frac{f(x_k) - f(x_{k+1})}{q_k(x_k) - q_k(x_{k+1})}. \tag{6.27}$$

⟦ = ratio of actual descent and descent predicted by the model ⟧
⟦ denominator is always non-negative ⟧ If the model is "good", then $\rho_k \approx 1$ will be close to 1. In particular, for $\rho_k \leq 0$ no descent is achieved (since the denominator is positive!).
In trust region methods, the search directions and the step lengths are not selected separately. Rather, they are selected in some sense simultaneously.

Algorithm 6.24 (Trust region method) *%input* $\widehat{\Delta}$, $\Delta_0 \in (0, \widehat{\Delta})$, $\eta \in [0, 1/4)$

for $k = 0, 1, \ldots$ **do** {
minimize q_k *with minimizer* $\widehat{x}_{k+1}$
$\rho_k = (f(\widehat{x}_{k+1}) - f(x_k))/(q_k(\widehat{x}_{k+1}) - q_k(x_k))$
if $\rho_k < 1/4$ **then** $\Delta_{k+1} := \frac{1}{4}\Delta_k$ *% Model "bad" → reduce trust region*
else if *(*$\rho_k > 3/4$ **and** $\|\widehat{x}_{k+1} - x_k\| = \Delta_k$*)* **then** $\Delta_{k+1} = \min(2\Delta_k, \widehat{\Delta})$
% model "good", minimizer at boundary → trust region apparently too small
else $\Delta_{k+1} = \Delta_k$
if $\rho_k > \eta$ **then** $x_{k+1} := \widehat{x}_{k+1}$ *% model OK, → accept step*
else $x_{k+1} := x_k$ *% model not OK → reject the step*
}

Remark 6.25 *The actual realization of a trust region method is non-trivial as the constrained minimization problem of finding* $\widehat{x}_{k+1}$ *has to be (approximately) solved. For actual realizations of trust region methods: see literature.* ∎