Optimization: Nonlinear Optimization without Constraints

Nonlinear optimization without constraints

Unconstrained minimization

$$\min_x f(x)$$

where $f(x)$ is nonlinear and non-convex.

Algorithms:

1. Newton and Quasi-Newton methods
2. Methods with direction determination and line optimization
3. Nelder-Mead Method

Newton's algorithm

2nd-order Taylor expansion

$$f(x) = f(x_0) + \nabla^T f(x_0) \cdot (x - x_0) + \frac{1}{2}(x - x_0)^T \cdot H(x_0) \cdot (x - x_0) + \mathcal{O}(\|x - x_0\|_2^3)$$

$\Rightarrow$ unconstrained quadratic problem

$$\min_{\tilde{x}} \frac{1}{2} \tilde{x}^T H(x_0) \tilde{x} + \nabla^T f(x_0) \tilde{x}$$

$$\tilde{x}_{\text{opt}} = (\, x - x_0 \,)_{\text{opt}} = -H^{-1}(x_0) \nabla f(x_0)$$

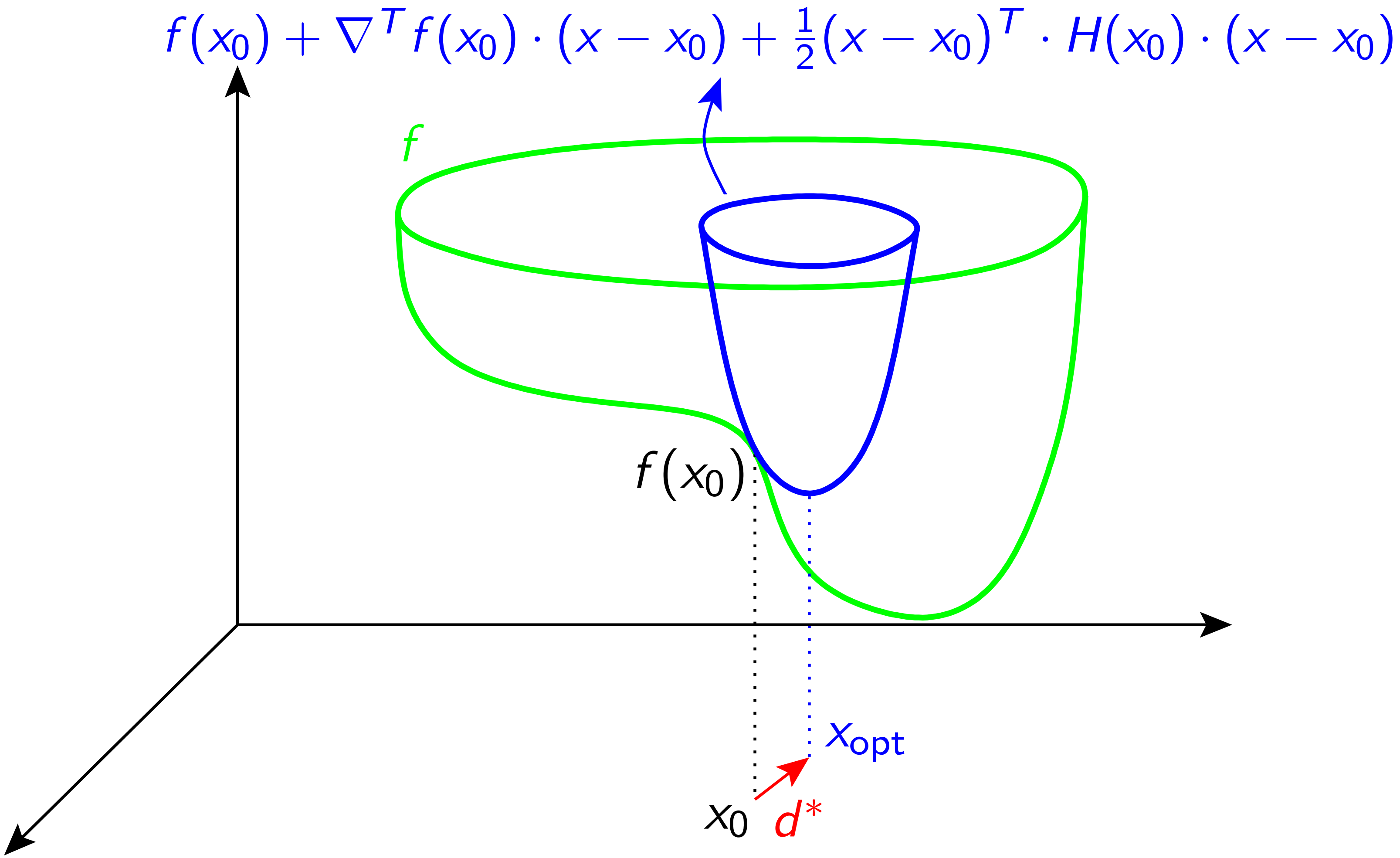
so

$$x_{\text{opt}} = x_0 - H^{-1}(x_0) \nabla f(x_0)$$

Newton's algorithm:

$$\rightarrow \; x_{k+1} = x_k - H^{-1}(x_k) \nabla f(x_k)$$

Newton's algorithm (continued)



$$d = x - x_0 \;\rightarrow\; \min_d \; \cancel{f(x_0)} + \nabla^T f(x_0) d + \frac{1}{2} d^T H(x_0) d$$

$$\rightarrow\; d^* = -H^{-1}(x_0)\nabla f(x_0) \;\rightarrow\; x_{\text{opt}} = x_0 + d^* = x_0 - H^{-1}(x_0)\nabla f(x_0)$$

Levenberg-Marquardt and quasi-Newton algorithms

Hessian matrix $H(x_k)$

- computation is time-consuming
- problems when close to singularity

Solution: choose approximate $\hat{H}_k$

- Levenberg-Marquardt algorithm
- Broyden-Fletcher-Goldfarb-Shanno quasi-Newton method
- Davidon-Fletcher-Powell quasi-Newton method

Modified Newton algorithm:

$$x_{k+1} = x_k - \hat{H}_k^{-1} \nabla f(x_k)$$

Modified Newton algorithm: $x_{k+1} = x_k - \hat{H}_k^{-1} \nabla f(x_k)$

- Levenberg-Marquardt algorithm:

$$\hat{H}_k = \lambda I + H(x_k)$$

- Broyden-Fletcher-Goldfarb-Shanno quasi-Newton method:

$$\hat{H}_k = \hat{H}_{k-1} + \frac{q_k q_k^T}{q_k^T s_k} - \frac{\hat{H}_{k-1}^T s_k s_k^T \hat{H}_{k-1}}{s_k^T \hat{H}_{k-1} s_k}$$

where
$$\begin{aligned} s_k &= x_k - x_{k-1} \\ q_k &= \nabla f(x_k) - \nabla f(x_{k-1}) \end{aligned}$$

- Davidon-Fletcher-Powell quasi-Newton method:

$$\hat{D}_k = \hat{D}_{k-1} + \frac{s_k s_k^T}{q_k^T s_k} - \frac{\hat{D}_{k-1} q_k q_k^T \hat{D}_{k-1}^T}{q_k^T \hat{D}_{k-1} q_k}$$

where
$$\begin{aligned} s_k &= x_k - x_{k-1} \\ q_k &= \nabla f(x_k) - \nabla f(x_{k-1}) \end{aligned}$$

$\rightarrow x_{k+1} = x_k - \hat{D}_k \nabla f(x_k) \rightarrow$ no inverse!

Nonlinear least squares problems

$$e(x) = \begin{bmatrix} e_1(x) & e_2(x) & \dots & e_N(x) \end{bmatrix}^T \quad (N \text{ components})$$

$$f(x) = \|e(x)\|_2^2 = e^T(x)\, e(x)$$

$$\Rightarrow \nabla f(x) = 2\, \nabla e(x)\, e(x)$$

$$\Rightarrow H(x) = 2\, \nabla e(x)\, \nabla^T e(x) + \sum_{i=1}^{N} 2\ \nabla^2 e_i(x) e_i(x)$$

with ∇e: Jacobian of e and $\nabla^2 e_i$: Hessian of e_i

$$e(x) \approx 0 \quad \Rightarrow \quad \hat{H}(x) = 2\, \nabla e(x)\, \nabla^T e(x)$$

Gauss-Newton method:

$$x_{k+1} = x_k - \left(\nabla e(x_k)\, \nabla^T e(x_k) \right)^{-1} \nabla e(x_k)\, e(x_k)$$

Levenberg-Marquardt method:

$$x_{k+1} = x_k - \left(\frac{\lambda I}{2} + \nabla e(x_k)\, \nabla^T e(x_k) \right)^{-1} \nabla e(x_k)\, e(x_k)$$

Direction determination & Line minimization

- Minimization along search direction
- Direction determination

n-dimensional minimization:

$$x^* = \arg\min_x f(x)$$

Choose a search direction d_k at x_k

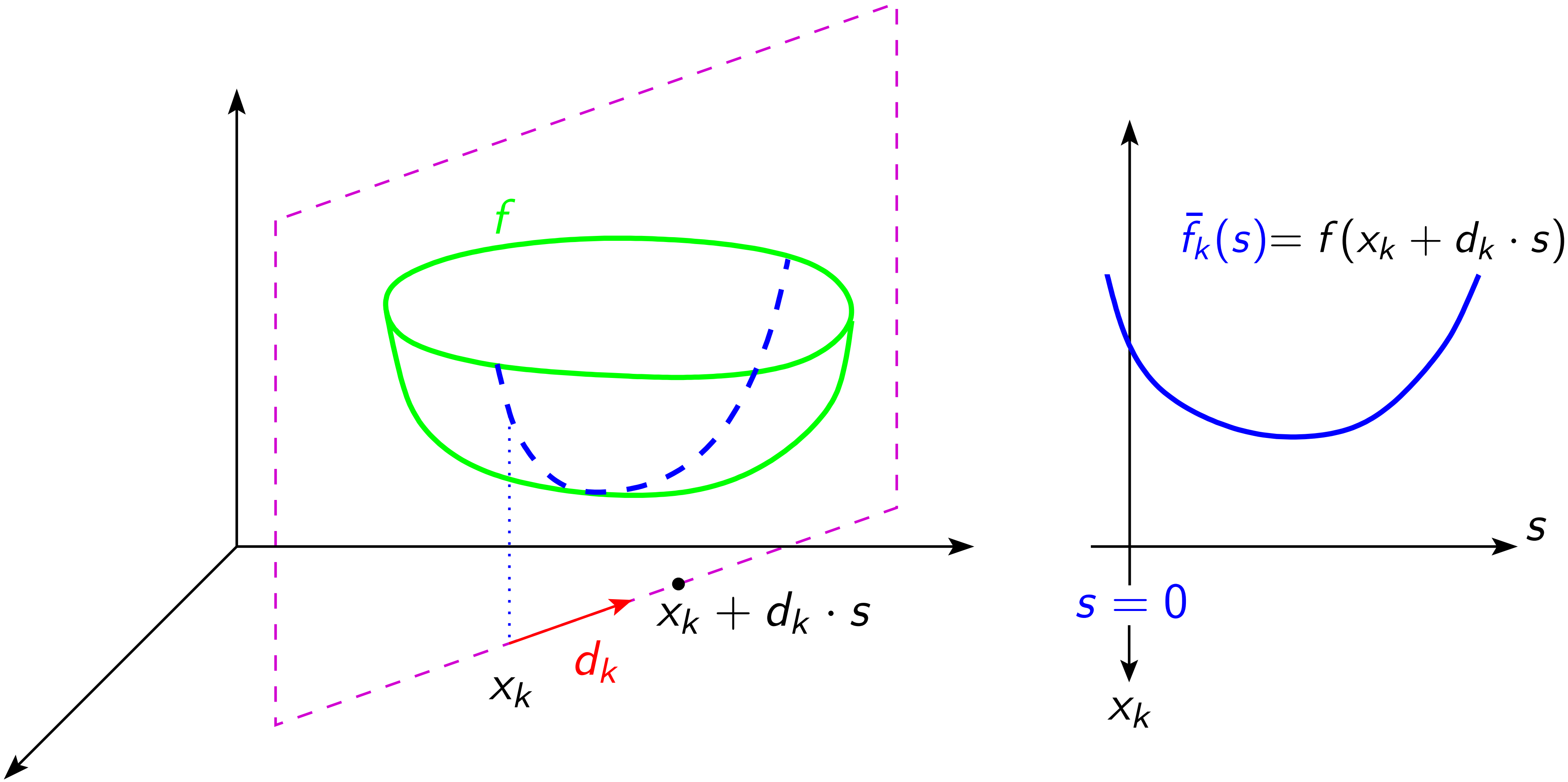
Minimize $f(x)$ over the line

$$x = x_k + d_k\, s \quad , \quad s \in \mathbb{R}$$

$\Rightarrow$ One-dimensional minimization:

$$x_{k+1} = x_k + d_k\, s_k^* \quad \text{with } s_k^* = \arg\min_s f(x_k + d_k\, s)$$

Direction determination & Line minimization (continued)



Line minimization

Initial point x_k

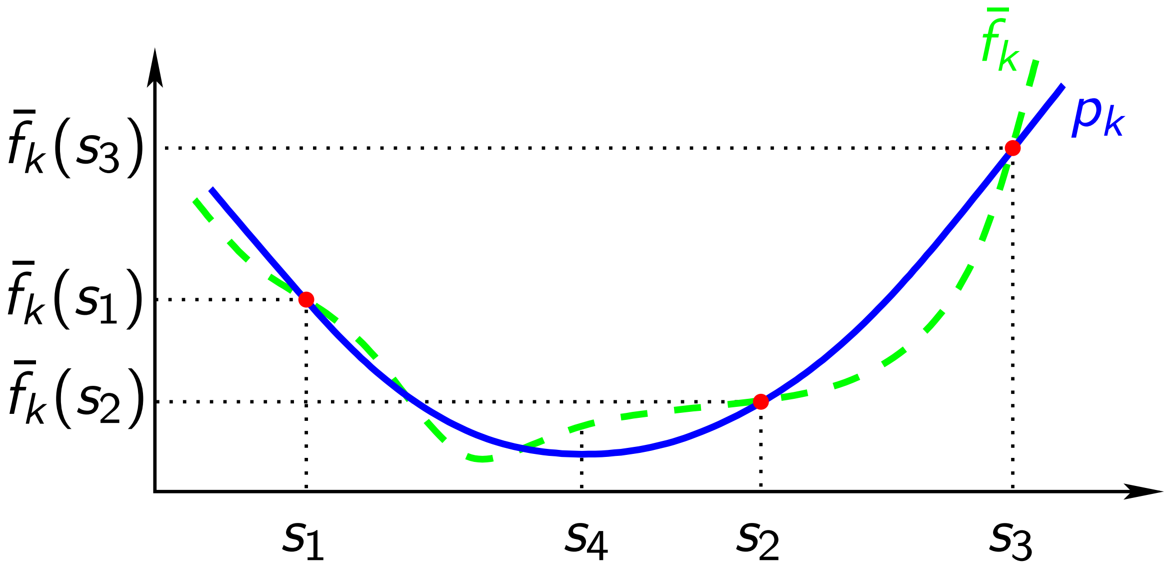
Search direction d_k

Line minimization

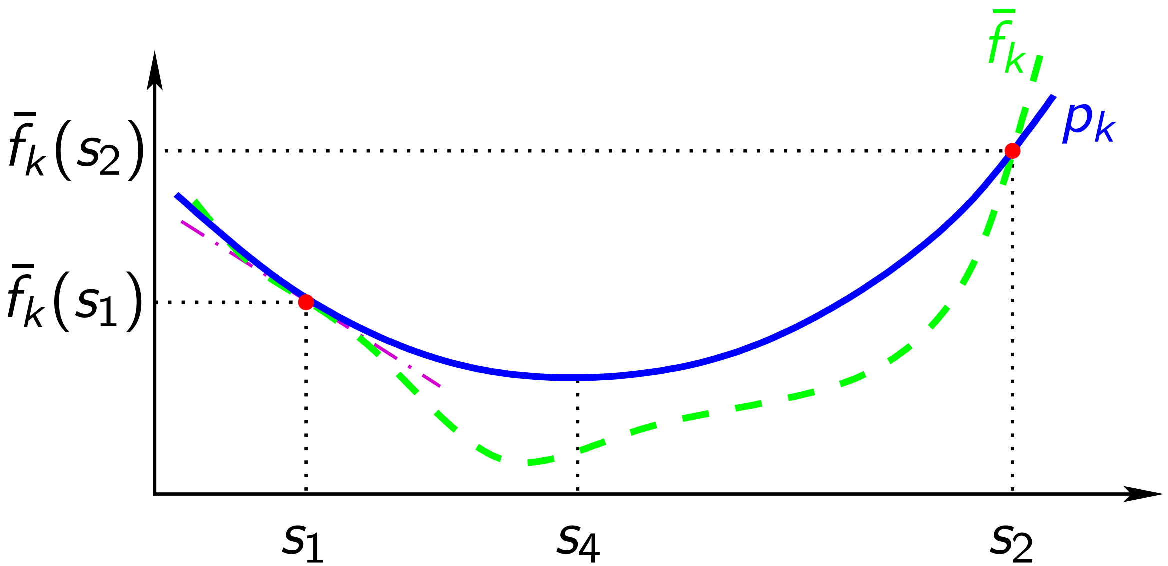
$$\min_s f(\, x_k + d_k\, s\,) = \min_s \bar{f}_k(\, s\,)$$

- Fixed / Variable step method
- Parabolic / Cubic interpolation
- Golden section / Fibonacci method

Parabolic interpolation



(a) using three function values



(b) using two function values and 1 derivative

Golden section method

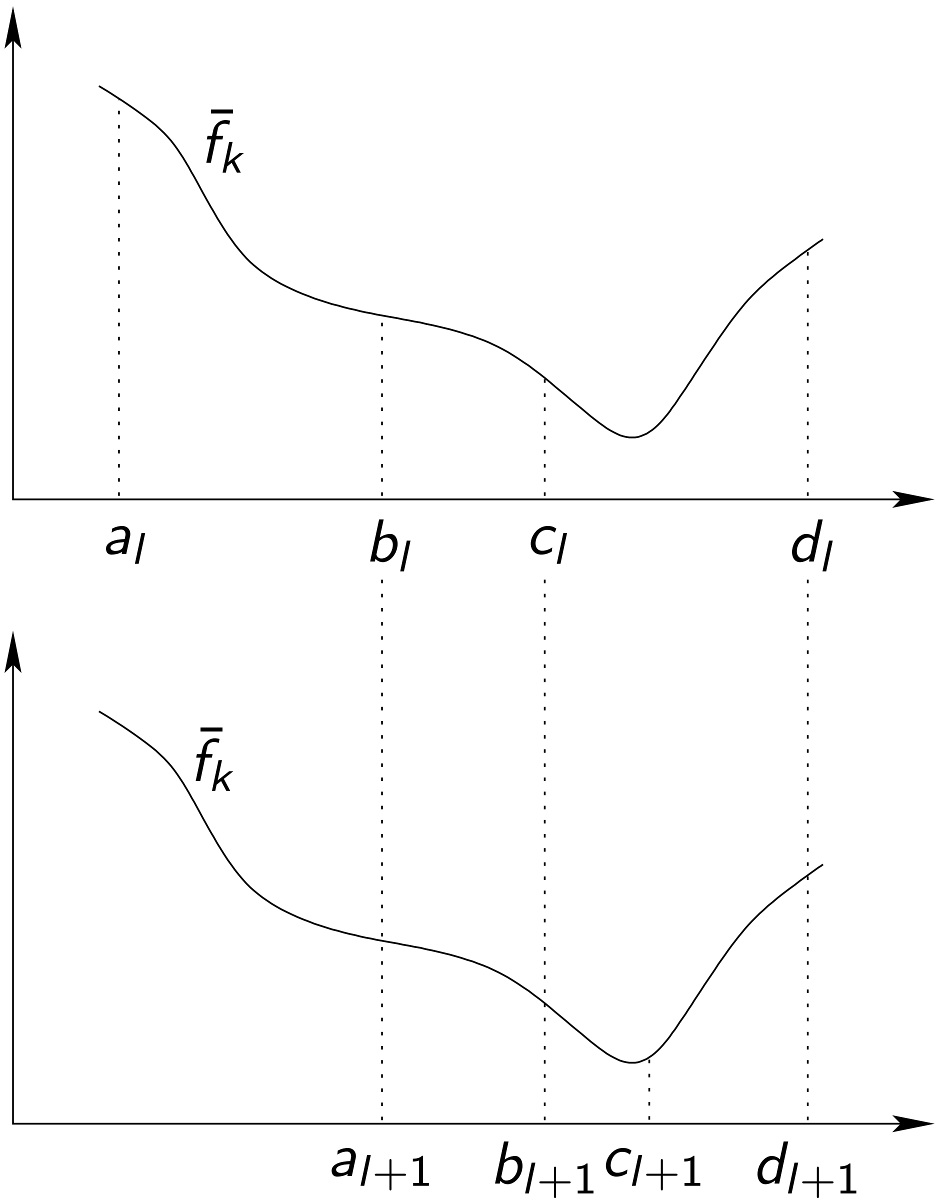
Suppose minimum in $[a_l, d_l]$, $\bar{f}_k$ unimodal in $[a_l, d_l]$

Construct:

$$b_l = \lambda\, a_l + (1-\lambda)\, d_l$$

$$c_l = (1-\lambda)\, a_l + \lambda\, d_l$$

Golden section method:
$\lambda = \frac{1}{2}(\sqrt{5}-1) \approx 0.6180$
$\rightarrow$ only one function evaluation per iteration



Fibonacci method:

Fibonacci sequence $\{\mu_k\} = 0,\ 1,\ 1,\ 2,\ 3,\ 5,\ 8,\ 13,\ 21, \ldots$

$$\mu_k = \mu_{k-1} + \mu_{k-2} \qquad \mu_0 = 0, \ \ \mu_1 = 1$$

Select n such that

$$\frac{1}{\mu_n}(b_0 - a_0) \leqslant \varepsilon$$

Next use

$$\lambda_l = \frac{\mu_{n-l}}{\mu_{n-l+1}}$$

Also allows to reuse points from one iteration to next

Fibonacci method gives optimal interval reduction for given number of function evaluations

Determination of search direction

- Gradient methods and conjugate-gradient methods
- Perpendicular search methods
  - Perpendicular method
  - Powell's perpendicular method

Gradient and conjugate-gradient methods

- Steepest descent:

$$d_k = -\nabla f(x_k)$$

- Conjugate gradient methods:

$$d_k = -\hat{H}_k^{-1} \nabla f(x_k)$$

  - Levenberg-Marquardt direction
  - Broyden-Fletcher-Goldfarb-Shanno direction
  - Davidon-Fletcher-Powell direction
  - Fletcher-Reeves direction:

$$d_k = -\nabla f(x_k) + \mu_k \, d_{k-1}$$

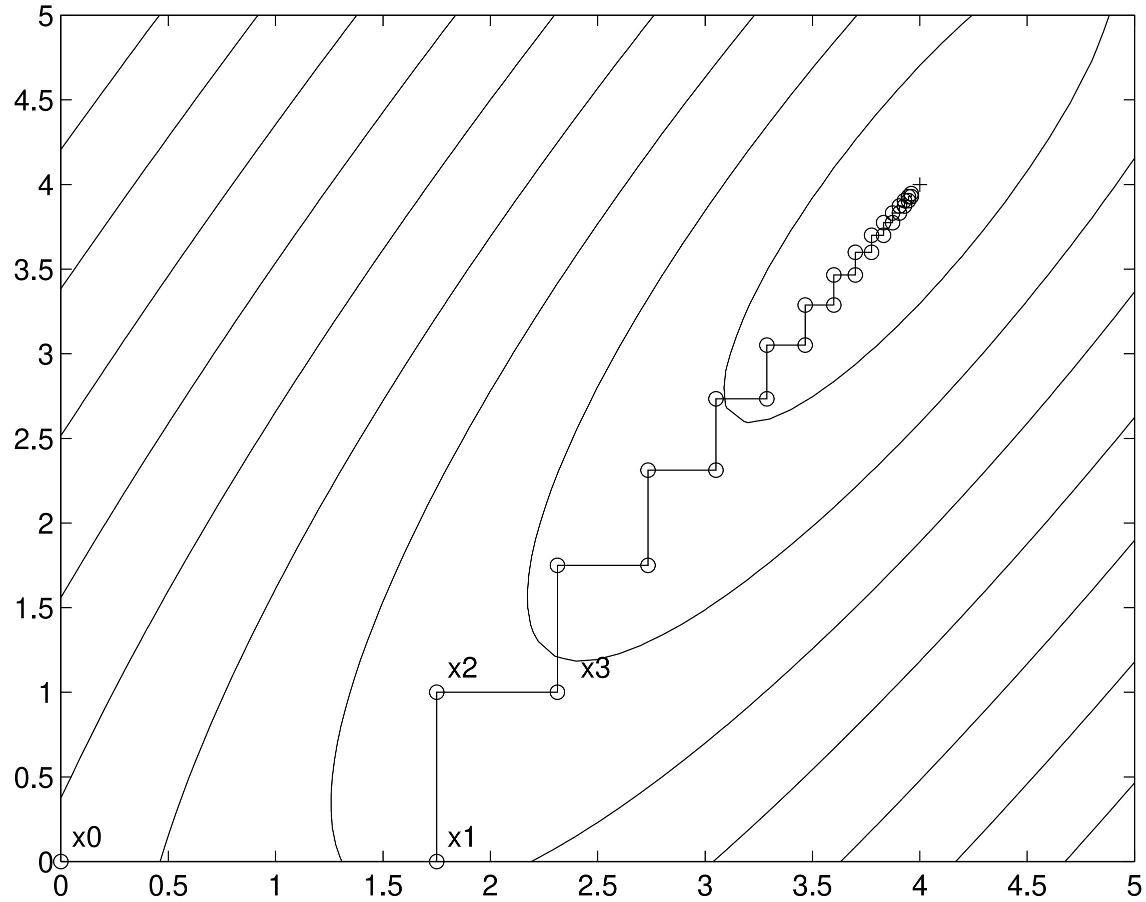
where

$$\mu_k = \frac{\nabla^T f(x_k) \nabla f(x_k)}{\nabla^T f(x_{k-1}) \nabla f(x_{k-1})}$$

Perpendicular search method

Perpendicular set of search directions:

$$d_0 = [\ 1 \quad 0 \quad 0 \quad \dots \quad 0\ ]^T$$
$$d_1 = [\ 0 \quad 1 \quad 0 \quad \dots \quad 0\ ]^T$$
$$\dots$$
$$d_{n-1} = [\ 0 \quad 0 \quad 0 \quad \dots \quad 1\ ]^T$$



Powell's perpendicular method

- Initial point: $\tilde{x}_0 := x_0$
- First set of search directions:

$$S_1 = (d_0, d_1, \dots, d_{n-1})$$

  results in $x_1, \dots, x_n$
- Perform search in direction $x_n - \tilde{x}_0 \longrightarrow \tilde{x}_n$
- New set of search directions: drop d_0 and add $x_n - \tilde{x}_0$:

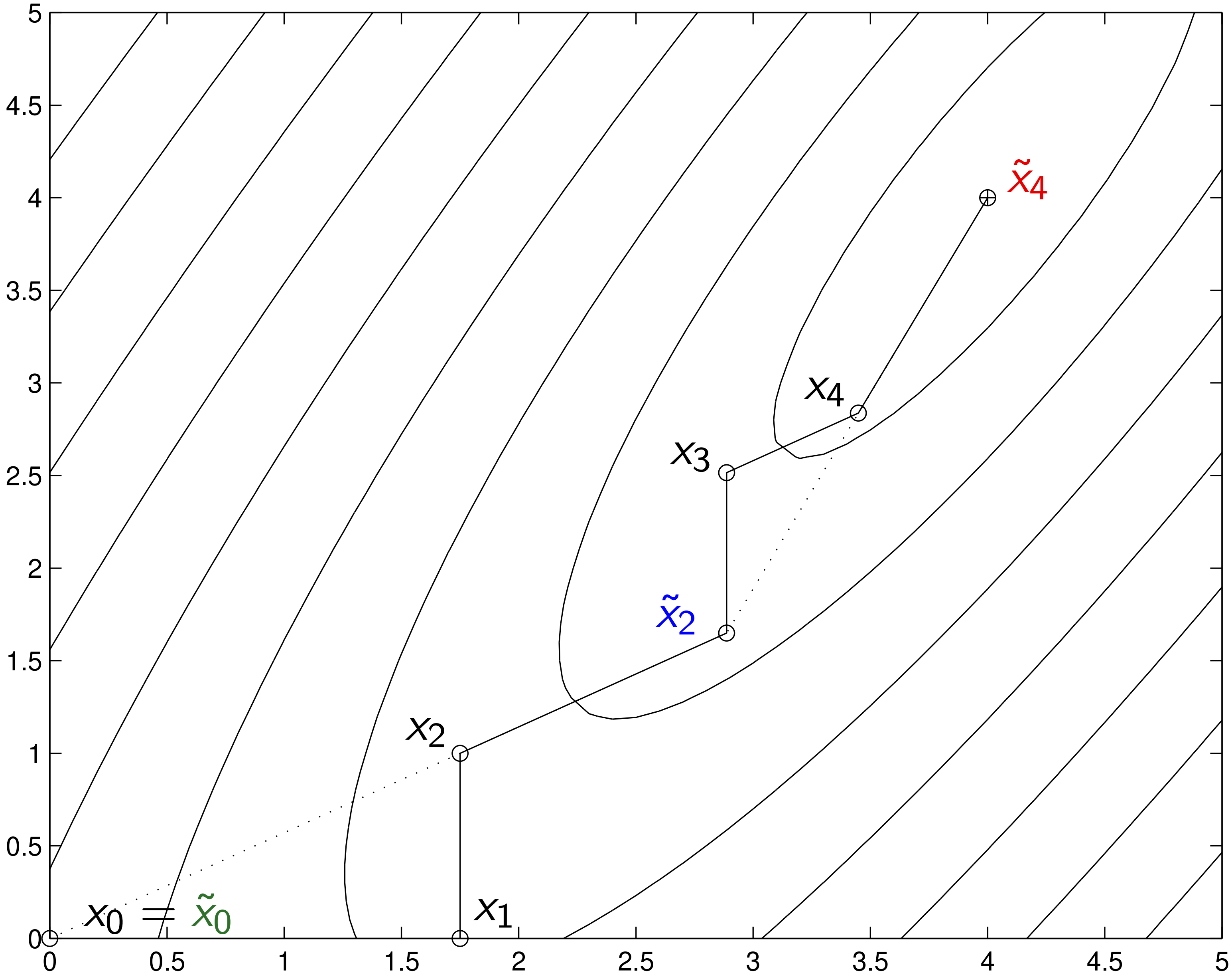
$$S_2 = (d_1, d_2, \dots, d_{n-1}, x_n - \tilde{x}_0)$$

  results in $x_{n+1}, \dots, x_{2n}$
- Perform search in direction $x_{2n} - \tilde{x}_n \longrightarrow \tilde{x}_{2n}$
- New set of search directions: drop d_1 and add $x_{2n} - \tilde{x}_n$:

$$S_3 = (d_2, d_3, \dots, d_{n-1}, x_n - \tilde{x}_0, x_{2n} - \tilde{x}_n)$$

- ...

Powell's perpendicular method (continued)



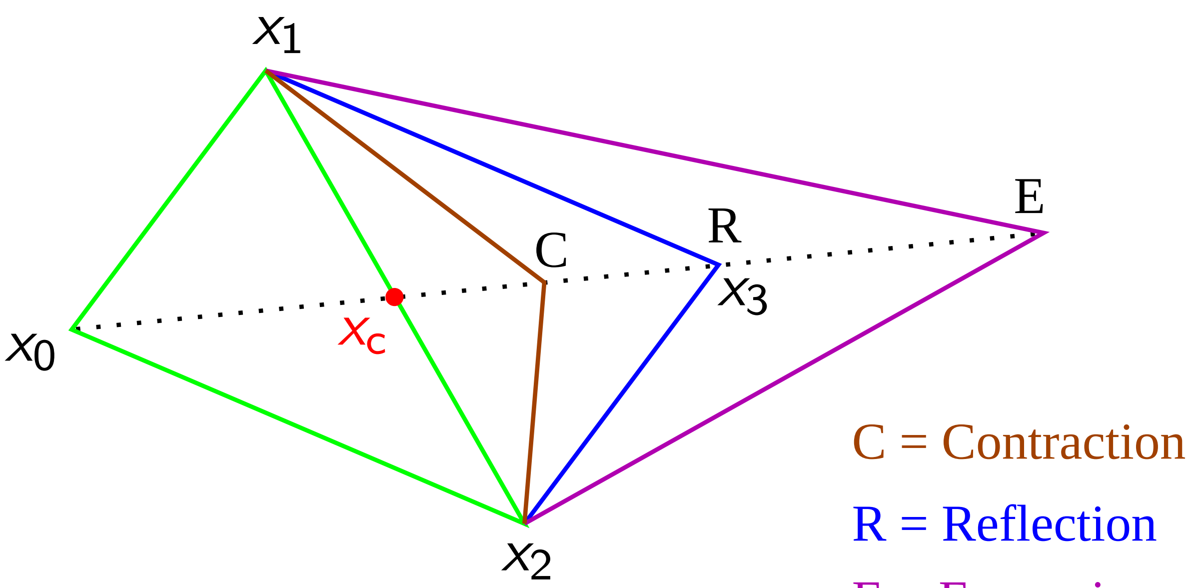
Nelder-Mead method

- Vertices of a simplex: (x_0, x_1, x_2)

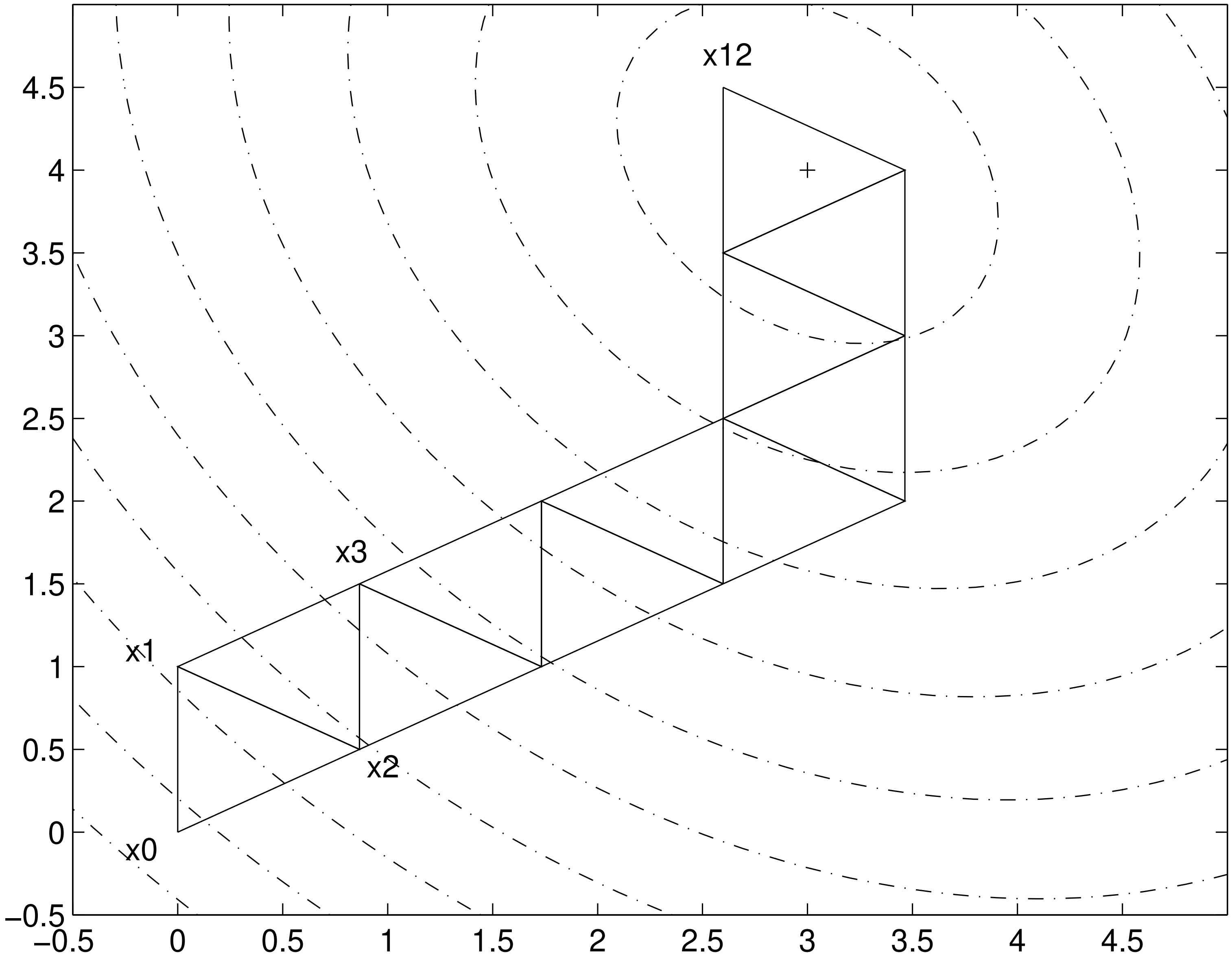
  Let $f(x_0) > f(x_1)$ and $f(x_0) > f(x_2)$

  $$x_3 = x_1 + x_2 - x_0$$

  $\rightarrow$ point reflection around $x_c = (x_1 + x_2)/2$



- Nelder-Mead method does not use gradient
- Method is less efficient than previous methods if more than 3–4 variables



Reflection in the Nelder-Mead algorithm

Summary

- Nonlinear programming without constraints: Standard form

$$\min_x f(x)$$

- Three main classes of algorithms:
  1. Newton and quasi-Newton methods
  2. Methods with direction determination and line optimization
  3. Nelder-Mead Method

Test: Classification of optimization problems I

- $$\max_{x \in \mathbb{R}^3} \exp(x_1 - x_2)$$
  $$\text{s.t. } |2x_1 + 3x_2 - 5x_3| \leqslant 1$$

  This problem is/can be recast as (check most restrictive answer):

  - ☐ linear programming problem
  - ☐ quadratic programming problem
  - ☐ nonlinear programming problem

Test: Classification of optimization problems II

- $$\max_{x \in \mathbb{R}^3} \; x_1 x_2 + x_2 x_3$$
  $$\text{s.t.} \;\; x_1^2 + x_2^2 + x_3^2 \leqslant 1$$

  This problem is/can be recast as (check most restrictive answer):

  - ☐ linear programming problem
  - ☐ quadratic programming problem
  - ☐ nonlinear programming problem