Trust region optimisation

Filip Hroch

May '22

- Which is an extreme way for a point mass in physics?
- How to build warehouse?
- The shortest (or fastest) path from Brno to Wischau?
- How to be drunk with both minimal time and budget?

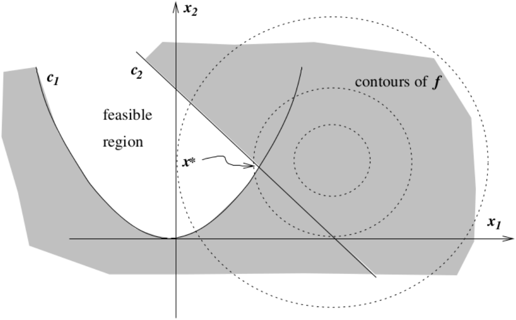
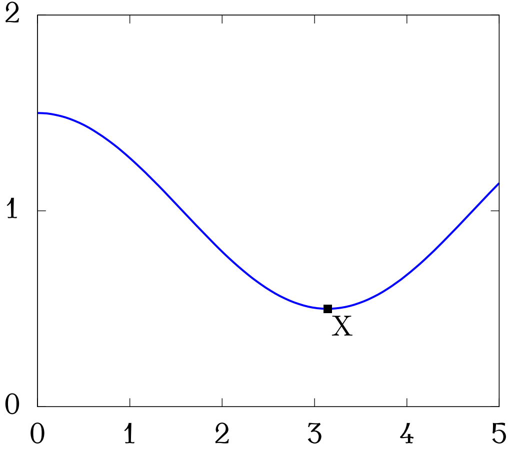
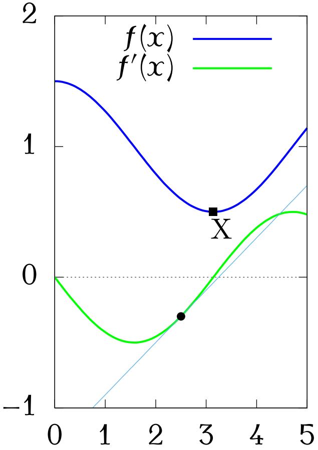


Fig. by Nocedal & Wright

$$\arg\min f(x), \quad x \in \mathbb{R}^n + \text{ initial estimate}$$





Taylor in vicinity of the optimum

$$f(x+p) = f(x) + \frac{\mathrm{d}f(x)}{\mathrm{d}x} p + \frac{1}{2} \frac{\mathrm{d}^2 f(x)}{dx^2} p^2 + \dots$$

The condition

$$\left.\frac{\mathrm{d}f(x)}{\mathrm{d}x}\right|_p + \frac{\mathrm{d}^2 f(x)}{\mathrm{d}x^2} \cdot p = 0$$

Optimum of n-dimensional function ($p \in \mathbb{R}^n$):

$$f(x+p) = f(x) + \nabla f(x)^T p + \frac{1}{2} p^T \nabla^2 f(x+p)\, p + \dots$$

Conditions:

$$\text{Jacobian} \nabla f(x) = 0 \text{ does vanish}$$

$$\text{Hessian} \nabla^2 f(x) \text{ is positive definite}$$

Traps:

- Non-positive definite Hessian
- Not so good the function approximation
- The initial point overpass

A model function of k-th step valid in trust region Δ_k:

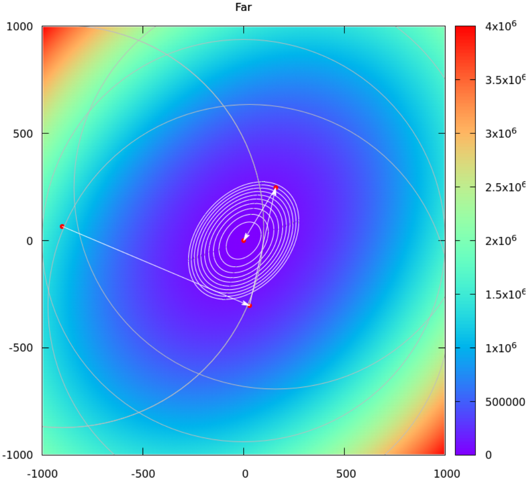
$$m(x+p) = f_k + (\nabla f_k)^T p + \frac{1}{2} p^T B_k p, \quad \|p_k\| < \Delta_k$$

Hessian (approximation): $B_k = \nabla^2 f_k$, or by a quasi-Newton method (SR1)

The approximation validity

$$\varrho_k = \frac{f_k - f_{k+1}}{m(0) - m(p)}$$

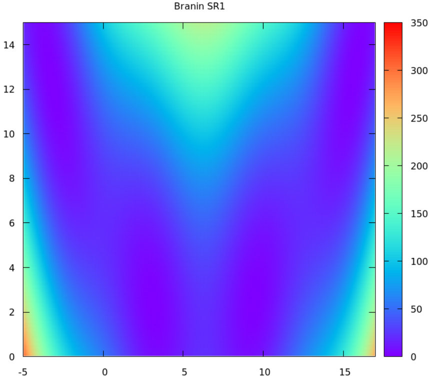
for $k = 1, \ldots$
    Obtain new estimate of p_k
    if $\varrho_k < 1/4$
        $\Delta_{k+1} = 1/4 \Delta_k$
    else
        if $\varrho_k > 3/4$ and $\|p_k\| = \Delta$
            $\Delta_{k+1} = \min(2\Delta_k, \widehat{\Delta})$
        else
            $\Delta_{k+1} = \Delta_k$
        end if
    end if
    if $\varrho_k > 1/4$
        $x_{k+1} = x_k + p_k$
    else
        $x_{k+1} = x_k$
    end if
end for



Graphics inspired by `https://optimization.mccormick.northwestern.edu/index.php/Trust-region_methods`

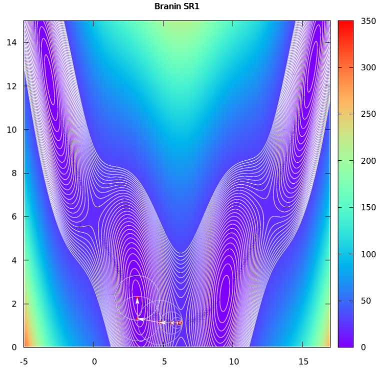
Branin function

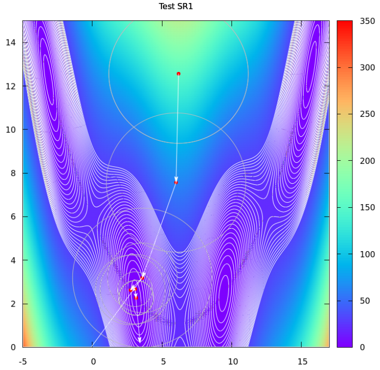
$$f(x, y) = a(y - bx^2 + cx - r)^2 + s(1 - t)\cos(x) + s$$



$$a = 1, b = 5.1/(4\pi^2), c = 5/\pi, r = 6, s = 10, t = 1/(8\pi)$$

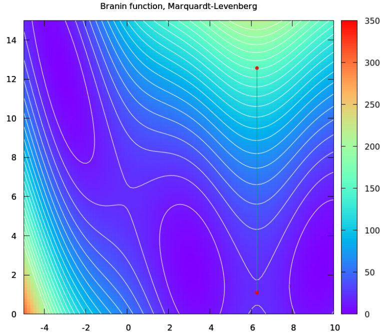
Trust region downhill





Marquardt-Levenberg method

- Implemented by Minpack `https://en.wikipedia.org/wiki/MINPACK`
- Python wrapper `curve_fit`
- Least-squares



$$f(x) = \frac{1}{2}\sum r_j^2, \quad J = \nabla r$$

$$(J^T J + \lambda I)p = -J^T r, \quad \|p\| \leq \Delta$$

`scipy.optimize`

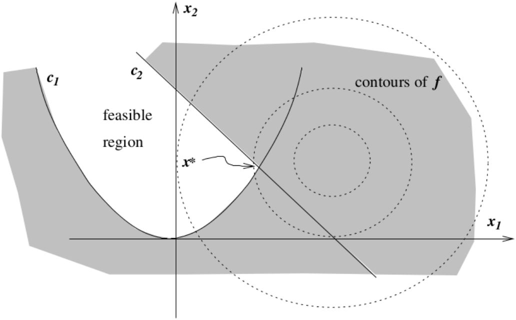
- large systems,
- general minimisation,
- conjugate-gradient

Optimisation with constrains

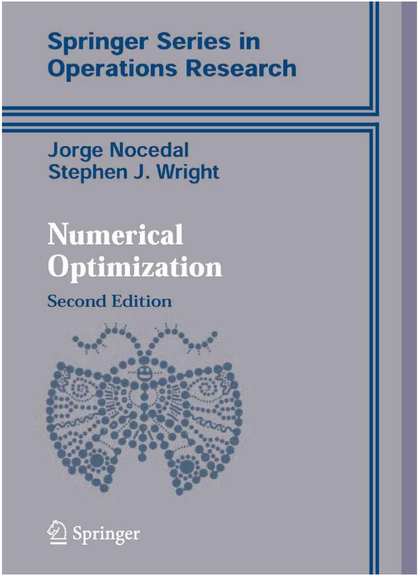
$$\arg\min_{x \in \mathbb{R}^n} f(x), \qquad \begin{array}{l} c_i(x) = 0, i \in \mathcal{E}, \\ c_i(x) \geq 0, i \in \mathcal{I}. \end{array}$$

The solution:

- Lagrange multiplicators
- augmented Hessian
- natural continuation of trust region methods



The book


Springer Series in
Operations Research
Jorge Nocedal
Stephen J. Wright
Numerical
Optimization
Second Edition
Springer

The end