Numerical Optimization of Partial Differential Equations

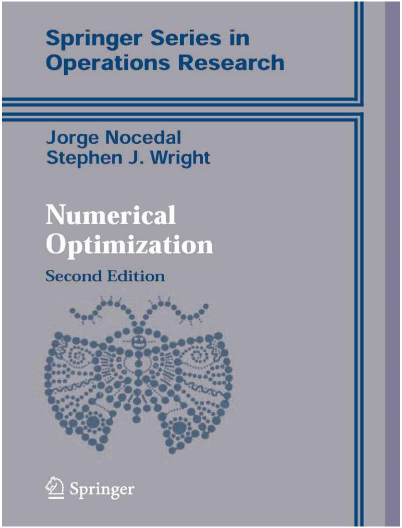
Part I: basic optimization concepts in $\mathbb{R}^n$

Bartosz Protas

Department of Mathematics & Statistics
McMaster University, Hamilton, Ontario, Canada
URL: http://www.math.mcmaster.ca/bprotas

Rencontres Normandes sur les aspects théoriques et numériques des EDP
5–9 November 2018, Rouen

A good reference for standard approaches



- Suppose $f : \mathbb{R}^N \rightarrow \mathbb{R}$, $N \geq 1$, is a twice continuously differentiable objective function

- Unconstrained Optimization Problems:

$$\min_{\mathbf{x}\in\mathbb{R}^N} f(\mathbf{x})$$

(for maximization problems, we can consider $min[-f(\mathbf{x})]$)

- A point $\tilde{\mathbf{x}}$ is a *global minimizer* if $f(\tilde{\mathbf{x}}) \leq f(\mathbf{x})$ for all $\mathbf{x}$
- A point $\tilde{\mathbf{x}}$ is a *local minimizer* if there exists a neighborhood $\mathcal{N}$ of $\tilde{\mathbf{x}}$ such that $f(\tilde{\mathbf{x}}) \leq f(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{N}$
  - A local minimizer is *strict* (or *strong*), if it is unique in $\mathcal{N}$

- Gradient of the objective function

$$\boldsymbol{\nabla} f(\mathbf{x}) := \left[\frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_N}\right]^T$$

- Hessian of the objective function

$$\left[\boldsymbol{\nabla}^2 f(\mathbf{x})\right]_{i,j} := \frac{\partial^2 f}{\partial x_j\, \partial x_i}, \qquad i,j = 1,\dots,N$$

Theorem (First-Order Necessary Condition)

If $\tilde{\mathbf{x}}$ is a local minimizer, then $\boldsymbol{\nabla} f(\tilde{\mathbf{x}}) = \mathbf{0}$.

Theorem (Second-Order Sufficient Conditions)

Suppose that $\boldsymbol{\nabla} f(\tilde{\mathbf{x}}) = \mathbf{0}$ and $\boldsymbol{\nabla}^2 f(\tilde{\mathbf{x}})$ is positive-definite. Then $\tilde{\mathbf{x}}$ is a strict local minimizers of f.

Unfortunately, analogous characterization of global minimizers is not possible

- How to find a local minimizer $\tilde{\mathbf{x}}$?

- Consider the following initial-value problem in $\mathbb{R}^N$, known as the *gradient flow*

$$\text{(GF)} \qquad \begin{cases} \dfrac{d\mathbf{x}(\tau)}{d\tau} = -\boldsymbol{\nabla} f(\mathbf{x}(\tau)), & \tau > 0, \\ \mathbf{x}(0) = \mathbf{x}_0, \end{cases}$$

  where
  - τ is a "pseudo-time" (a parametrization)
  - $\mathbf{x}_0$ is a suitable initial guess

- Then, $\lim_{\tau\to\infty} \mathbf{x}(\tau) = \tilde{\mathbf{x}}$

  In principle, the gradient flow may converge to a *saddle point* $\mathbf{x}_s$, where $\boldsymbol{\nabla} f(\mathbf{x}_s) = \mathbf{0}$ and the Hessian $\boldsymbol{\nabla}^2 f(\mathbf{x}_s)$ is *not* positive-definite, but in actual computations this is very unlikely.

- Discretize the gradient flow (GF) with *Euler's explicit method*

$$\text{(SD)} \qquad \begin{cases} \mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} - \Delta\tau\, \boldsymbol{\nabla} f(\mathbf{x}^{(n)}), & n = 1, 2, \dots, \\ \mathbf{x}^{(0)} = \mathbf{x}_0, \end{cases}$$

  where
  - $\mathbf{x}^{(n)} := \mathbf{x}(n\,\Delta\tau)$, such that $\lim_{n\to\infty} \mathbf{x}^{(n)} = \tilde{\mathbf{x}}$
  - $\Delta\tau$ is a *fixed* step size (since Euler's explicit scheme is only conditionally stable, $\Delta\tau$ must be sufficiently small)

- In principle, the gradient flow (GF) can be discretized with higher-order schemes, including implicit approaches, but they are not easy to apply to PDE optimization problems, hence will not be considered here.

Algorithm 1 Steepest Descent (SD)

1: $\mathbf{x}^{(0)} \leftarrow \mathbf{x}_0$ (initial guess)
2: $n \leftarrow 0$
3: **repeat**
4: compute the gradient $\boldsymbol{\nabla} f(\mathbf{x}^{(n)})$
5: update $\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} - \Delta\tau\, \boldsymbol{\nabla} f(\mathbf{x}^{(n)})$
6: $n \leftarrow n + 1$
7: **until** $\frac{|f(\mathbf{x}^{(n)}) - f(\mathbf{x}^{(n-1)})|}{|f(\mathbf{x}^{(n-1)})|} < \varepsilon_f$

Input:

$\mathbf{x}_0$ — initial guess
$\Delta\tau$ — fixed step size
ε_f — tolerance in the termination condition

Output:

an approximation of the minimizer $\tilde{\mathbf{x}}$

Computational Tests

- Rosenbrock's "banana" function

$$f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

- Global minimizer

$$x_1 = x_2 = 1, \qquad f(1,1) = 0$$

- The function is known for its poor conditioning
  - eigenvalues of the Hessian $\nabla^2 f$ at the minimum:

$$\lambda_1 \approx 0.4, \qquad \lambda_2 \approx 1001.6$$

- Choice of the step size $\Delta\tau$: steepest descent is not meant to approximate the gradient flow (GF) accurately, but to minimize $f(\mathbf{x})$ rapidly
- Sufficient decrease — *Armijo's condition*

$$f(\mathbf{x}^{(n)} + \tau\,\mathbf{p}^{(n)}) \le f(\mathbf{x}^{(n)}) - C\,\tau\,\boldsymbol{\nabla} f(\mathbf{x}^{(n)})^T\mathbf{p}^{(n)}$$

where $\mathbf{p}^{(n)}$ is a search direction and $C \in (0,1)$

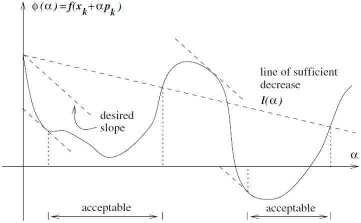


Figure credit: Nocedal & Wright (1999)

- *Wolfe's condition:* sufficient decrease and curvature

- Optimize the step size at every iteration by solving the line minimization (line-search) problem

$$\tau_n := \operatorname{argmin}_{\tau>0} f(\mathbf{x}^{(n)} - \tau \boldsymbol{\nabla} f(\mathbf{x}^{(n)}))$$

- *Brent's method for line minimization:* a combination of the golden-section search with parabolic interpolation (derivative-free)

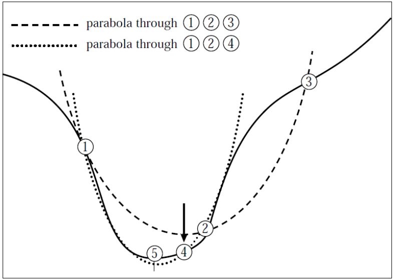


Figure credit: Numerical Recipes in C (1992)

- A robust implementation of Brent's method available in *Numerical Recipes in C (1992)*, see also the function `fminbnd` in MATLAB

Algorithm 2 Steepest Descent with Line Search (SDLS)

1: $\mathbf{x}^{(0)} \leftarrow \mathbf{x}_0$ (initial guess)
2: $n \leftarrow 0$
3: **repeat**
4: compute the gradient $\boldsymbol{\nabla} f(\mathbf{x}^{(n)})$
5: determine optimal step size $\tau_n = \operatorname{argmin}_{\tau>0} f(\mathbf{x}^{(n)} - \tau \boldsymbol{\nabla} f(\mathbf{x}^{(n)}))$
6: update $\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} - \tau_n \boldsymbol{\nabla} f(\mathbf{x}^{(n)})$
7: $n \leftarrow n+1$
8: **until** $\frac{|f(\mathbf{x}^{(n)})-f(\mathbf{x}^{(n-1)})|}{|f(\mathbf{x}^{(n-1)})|} < \varepsilon_f$

Input:

$\mathbf{x}_0$ — initial guess
ε_τ — tolerance in line search
ε_f — tolerance in the termination condition

Output:

an approximation of the minimizer $\tilde{\mathbf{x}}$

- Consider, for now, minimization of a quadratic form

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} - \mathbf{b}^T\mathbf{x},$$

  where $\mathbf{A} \in \mathbb{R}^{N\times N}$ is a symmetric, positive-definite matrix and $\mathbf{b} \in \mathbb{R}^N$

- Then,

$$\boldsymbol{\nabla} f(\mathbf{x}) = \mathbf{A}\mathbf{x} - \mathbf{b} =: \mathbf{r}$$

  such that minimizing $f(\mathbf{x})$ is equivalent to solving $\mathbf{A}\mathbf{x} = \mathbf{b}$

- A set of nonzero vectors $[\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_k]$ is said to be *conjugate* with respect to matrix $\mathbf{A}$ if

$$\mathbf{p}_i^T\mathbf{A}\mathbf{p}_j = 0, \qquad \forall i, j = 0, \dots, k, \ \ i \neq j$$

  (conjugacy implies linear independence)

- Conjugate Gradient (CG) method

$$
\begin{aligned}
\mathbf{x}^{(n+1)} &= \mathbf{x}^{(n)} + \tau_n \, \mathbf{p}^{(n)}, && n = 1, 2, \dots, \\
\mathbf{p}^{(n)} &= -\mathbf{r}^{(n)} + \beta^{(n)} \, \mathbf{p}_{n-1}, && (\mathbf{r}^{(n)} = \boldsymbol{\nabla} f(\mathbf{x}^{(n)}) = \mathbf{A}\mathbf{x}^{(n)} - \mathbf{b}), \\
\beta^{(n)} &= \frac{(\mathbf{r}^{(n)})^T \mathbf{A}\mathbf{p}^{(n-1)}}{(\mathbf{p}^{(n-1)})^T \mathbf{A}\mathbf{p}^{(n-1)}}, && (\text{"momentum"}), \\
\tau_n &= -\frac{(\mathbf{r}^{(n)})^T \mathbf{p}^{(n)}}{(\mathbf{p}^{(n)})^T \mathbf{A}\mathbf{p}^{(n)}}, && (\text{exact formula for optimal step size}), \\
\mathbf{x}_0 &= \mathbf{x}^0, \qquad \mathbf{p}^{(0)} = -\mathbf{r}^{(0)}
\end{aligned}
$$

- The directions $\mathbf{p}^{(0)}, \mathbf{p}^{(1)}, \dots, \mathbf{p}^{(n)}$ generated by the CG method are conjugate with respect to matrix $\mathbf{A}$
  - this gives rise to a number of interesting and useful properties

Theorem (properties of CG iterations)

The iterates generated by the CG method have the following properties

- $$\begin{aligned}\operatorname{span}\left\{\mathbf{p}^{(0)},\mathbf{p}^{(1)},\dots,\mathbf{p}^{(n)}\right\} &= \operatorname{span}\left\{\mathbf{r}^{(0)},\mathbf{r}^{(1)},\dots,\mathbf{r}^{(n)}\right\} \\ &= \operatorname{span}\left\{\mathbf{r}^{(0)},\mathbf{A}\mathbf{r}^{(0)},\dots,\mathbf{A}^{n}\mathbf{r}^{(0)}\right\}\end{aligned}$$
- *(the expanding subspace property)*
  $$(\mathbf{r}^{(n)})^T\mathbf{r}^{(k)} = (\mathbf{r}^{(n)})^T\mathbf{p}^{(k)} = 0, \qquad \forall i = 0,\dots,n-1$$
- $\mathbf{x}^{(n)}$ *is the minimizer of* $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} - \mathbf{b}^T\mathbf{x}$ *over the set*
  $$\left\{\mathbf{x}^0 + \operatorname{span}\left\{\mathbf{p}^{(0)},\mathbf{p}^{(1)},\dots,\mathbf{p}^{(n)}\right\}\right\}$$

- Thus, in the Conjugate Gradients method minimization of $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} - \mathbf{b}^T\mathbf{x}$ is performed by solving (exactly) $N = \dim(\mathbf{x})$ line-minimization problems along the conjugate directions $\left\{\mathbf{p}^{(0)}, \mathbf{p}^{(1)}, \dots, \mathbf{p}^{(n)}\right\}$
- As a result, convergence to $\tilde{\mathbf{x}}$ is achieved in at most N iterations
- What happens when $f(\mathbf{x})$ is a general convex function?

- The (linear) Conjugate Gradients method admits a generalization to the nonlinear setting by:
  - replacing the residual $\mathbf{r}^{(n)}$ with the gradient $\boldsymbol{\nabla} f(\mathbf{x}^{(n)})$
  - computing the step size via line search $\tau_n = \operatorname{argmin}_{\tau>0} f(\mathbf{x}^{(n)} - \tau\, \boldsymbol{\nabla} f(\mathbf{x}^{(n)}))$
  - using a more general expressions for the "momentum" term $\beta^{(n)}$ (such that the descent directions $\mathbf{p}^{(0)}, \mathbf{p}^{(1)}, \dots, \mathbf{p}^{(n)}$ will only be *approximately* conjugate)

- Nonlinear Conjugate Gradient (NCG) method

$$\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} + \tau_n \, \mathbf{p}^{(n)}, \qquad n = 1, 2, \dots,$$

$$\mathbf{p}^{(n)} = -\boldsymbol{\nabla} f(\mathbf{x}^{(n)}) + \beta^{(n)} \, \mathbf{p}_{n-1},$$

$$\beta^{(n)} = \begin{cases} \dfrac{\left(\boldsymbol{\nabla} f(\mathbf{x}^{(n)})\right)^T \boldsymbol{\nabla} f(\mathbf{x}^{(n)})}{(\boldsymbol{\nabla} f(\mathbf{x}^{(n-1)}))^T \boldsymbol{\nabla} f(\mathbf{x}^{(n-1)})} & \text{(Fletcher-Reeves)}, \\[2ex] \dfrac{\left(\boldsymbol{\nabla} f(\mathbf{x}^{(n)})\right)^T \left(\boldsymbol{\nabla} f(\mathbf{x}^{(n)}) - \boldsymbol{\nabla} f(\mathbf{x}^{(n-1)})\right)}{(\boldsymbol{\nabla} f(\mathbf{x}^{(n-1)}))^T \boldsymbol{\nabla} f(\mathbf{x}^{(n-1)})} & \text{(Polak-Ribière)}, \end{cases}$$

$$\tau_n = \operatorname{argmin}_{\tau > 0} f(\mathbf{x}^{(n)} - \tau \, \boldsymbol{\nabla} f(\mathbf{x}^{(n)})),$$

$$\mathbf{x}_0 = \mathbf{x}^0, \qquad \mathbf{p}^{(0)} = -\boldsymbol{\nabla} f(\mathbf{x}^{(0)})$$

- For quadratic functions $f(\mathbf{x})$, both the Fletcher-Reeves (FR) and the Polak-Ribière (PR) variant coincide with the the linear CG
- In general, the descent directions $\mathbf{p}^{(0)}, \mathbf{p}^{(1)}, \dots, \mathbf{p}^{(n)}$ are now only *approximately* conjugate

Algorithm 3 Polak-Ribière version of Conjugate Gradient (CG-PR)

$\mathbf{x}^{(0)} \leftarrow \mathbf{x}_0$ (initial guess)
$n \leftarrow 0$
repeat
  compute the gradient $\boldsymbol{\nabla} f(\mathbf{x}^{(n)})$
  calculate $\beta_n = \dfrac{\left(\boldsymbol{\nabla} f(\mathbf{x}^{(n)})\right)^T \left(\boldsymbol{\nabla} f(\mathbf{x}^{(n)}) - \boldsymbol{\nabla} f(\mathbf{x}^{(n-1)})\right)}{\left(\boldsymbol{\nabla} f(\mathbf{x}^{(n-1)})\right)^T \boldsymbol{\nabla} f(\mathbf{x}^{(n-1)})}$
  determine the descent direction $\mathbf{p}^{(n)} = -\boldsymbol{\nabla} f(\mathbf{x}^{(n)}) + \beta^{(n)} \mathbf{p}_{n-1}$
  determine optimal step size $\tau_n = \operatorname{argmin}_{\tau>0} f(\mathbf{x}^{(n)} + \tau \, \mathbf{p}^{(n)})$
  update $\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} + \tau_n \, \mathbf{p}^{(n)}$
  $n \leftarrow n + 1$
until $\dfrac{|f(\mathbf{x}^{(n)}) - f(\mathbf{x}^{(n-1)})|}{|f(\mathbf{x}^{(n-1)})|} < \varepsilon_f$

Input:

$\mathbf{x}_0$ — initial guess, ε_τ — tolerance in line search
ε_f — tolerance in the termination condition

Output:

an approximation of the minimizer $\tilde{\mathbf{x}}$

Convergence theory — quadratic case (I)

- Let $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} - \mathbf{b}^T\mathbf{x}$, where the matrix $\mathbf{A}$ has eigenvalues $0 < \lambda_1 \leq \cdots \leq \lambda_N$ and $\|\mathbf{x}\|_{\mathbf{A}} = \mathbf{x}^T\mathbf{A}\mathbf{x}$

Theorem (Linear Convergence of Steepest Descent)

For the Steepest-Descent approach we have the following estimate

$$\|\mathbf{x}^{(n+1)} - \tilde{\mathbf{x}}\|_{\mathbf{A}}^2 \leq \left(\frac{\lambda_N - \lambda_1}{\lambda_N + \lambda_1}\right)^2 \|\mathbf{x}^{(n)} - \tilde{\mathbf{x}}\|_{\mathbf{A}}^2$$

- The rate of convergence is controlled by the “spread” of the eigenvalues of $\mathbf{A}$

Convergence theory — quadratic case (II)

Theorem (Convergence of Linear Conjugate Gradients)

For the linear Conjugate Gradients approach we have the following estimate

$$\|\mathbf{x}^{(n+1)} - \tilde{\mathbf{x}}\|_{\mathbf{A}}^2 \le \left(\frac{\lambda_{N-n} - \lambda_1}{\lambda_{N-n} + \lambda_1}\right)^2 \|\mathbf{x}_0 - \tilde{\mathbf{x}}\|_{\mathbf{A}}^2$$

- The iterates take out one eigenvalue at a time
  - clustering of eigenvalues matters
- In the nonlinear setting, it is advantageous to periodically reset β_n to zero (helpful in practice and simplifies some convergence proofs)

- What about problems with equality constraints?
  - suppose $\mathbf{c} : \mathbb{R}^N \to \mathbb{R}^M$, where $1 \le M < N$
  - then, we have an *equality-constrained optimization problem*

$$\min_{\mathbf{x}\in\mathbb{R}^N} f(\mathbf{x})$$

$$\text{subject to: } \mathbf{c}(\mathbf{x}) = \mathbf{0}$$

- If the constraint equation can be “solved” and we can write $\mathbf{x} = \mathbf{y} + \mathbf{z}$, where $\mathbf{y} \in \mathbb{R}^{N-M}$ and $\mathbf{z} = \mathbf{g}(\mathbf{y}) \in \mathbb{R}^M$, then the problem is reduced to an unconstrained one with a *reduced* objective function

$$\min_{\mathbf{y}\in\mathbb{R}^{N-M}} f(\mathbf{y} + \mathbf{g}(\mathbf{y}))$$

- Consider *augmented* objective function $\mathbf{L} : \mathbb{R}^N \to \mathbb{R}$

$$\mathbf{L}(\mathbf{x}, \boldsymbol{\lambda}) := f(\mathbf{x}) - \boldsymbol{\lambda}^T \mathbf{c}(\mathbf{x}),$$

  where $\boldsymbol{\lambda} \in \mathbb{R}^M$ is the *Lagrange multiplier*

- Differentiating the augmented objective function with respect to $\mathbf{x}$

$$\boldsymbol{\nabla}_{\mathbf{x}} \mathbf{L}(\mathbf{x}, \boldsymbol{\lambda}) := \boldsymbol{\nabla} f(\mathbf{x}) - \boldsymbol{\lambda}^T \boldsymbol{\nabla} \mathbf{c}(\mathbf{x})$$

Theorem (First-Order Necessary Condition)

If $\tilde{\mathbf{x}}$ is a local minimizer of an equality-constrained optimization problem, then there exists $\boldsymbol{\lambda} \in \mathbb{R}^M$ such that the following equations are satisfied

$$\boldsymbol{\nabla} f(\tilde{\mathbf{x}}) - \boldsymbol{\lambda}^T \boldsymbol{\nabla} \mathbf{c}(\tilde{\mathbf{x}}) = \mathbf{0}, \qquad \mathbf{c}(\tilde{\mathbf{x}}) = \mathbf{0}$$

For *inequality*-constrained problems, the first-order necessary conditions become more complicated — the Karush-Kuhn-Tucker (KKT) conditions

- How to compute equality-constrained minimizers with a gradient method?
- At each $\mathbf{x} \in \mathbb{R}^N$ the linearized constraint function $\boldsymbol{\nabla}\mathbf{c}(\mathbf{x}) \in \mathbb{R}^{M\times N}$ defines a (kernel) subspace with dimension $\operatorname{rank}[\boldsymbol{\nabla}\mathbf{c}(\mathbf{x})]$

$$\mathcal{S}_{\mathbf{x}} := \{\mathbf{x}' \in \mathcal{R}^N, \quad \boldsymbol{\nabla}\mathbf{c}(\mathbf{x})\mathbf{x}' = \mathbf{0}\}$$

  - this is the subspace *tangent* to the constraint manifold at $\mathbf{x}$
  - we need to project the gradient $\boldsymbol{\nabla} f(\mathbf{x})$ onto $\mathcal{S}_{\mathbf{x}}$
- Assuming that $\operatorname{rank}[\boldsymbol{\nabla}\mathbf{c}(\mathbf{x})] = M$, the projection operator $\mathbf{P}_{\mathcal{S}_{\mathbf{x}}} \ : \ \mathbb{R}^N \to \mathcal{S}_{\mathbf{x}}$ is given by

$$\mathbf{P}_{\mathcal{S}_{\mathbf{x}}} := \mathbf{I} - \boldsymbol{\nabla}\mathbf{c}(\mathbf{x}) \left[(\boldsymbol{\nabla}\mathbf{c}(\mathbf{x}))^T \boldsymbol{\nabla}\mathbf{c}(\mathbf{x}) \right]^{-1} (\boldsymbol{\nabla}\mathbf{c}(\mathbf{x}))^T$$

- Replace $\boldsymbol{\nabla} f(\mathbf{x})$ with $\mathbf{P}_{\mathcal{S}_{\mathbf{x}}}\boldsymbol{\nabla} f(\mathbf{x})$ in the gradient method (SD or SDLS)
  - nonlinear constraints satisfied with an error $\mathcal{O}((\Delta\tau)^2)$ or $\mathcal{O}(\tau_n^2)$

Algorithm 4 Projected Steepest Descent (PSD)

1: $\mathbf{x}^{(0)} \leftarrow \mathbf{x}_0$ (initial guess)
2: $n \leftarrow 0$
3: **repeat**
4: compute the gradient $\boldsymbol{\nabla} f(\mathbf{x}^{(n)})$
5: compute linearization of the constraint $\boldsymbol{\nabla}\mathbf{c}(\mathbf{x}^{(n)})$
6: determine the projector $\mathbf{P}_{\mathcal{S}_{\mathbf{x}^{(n)}}}$
7: determine the projected gradient $\mathbf{P}_{\mathcal{S}_{\mathbf{x}^{(n)}}}\boldsymbol{\nabla} f(\mathbf{x}^{(n)})$
8: update $\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} - \Delta\tau\, \mathbf{P}_{\mathcal{S}_{\mathbf{x}^{(n)}}}\boldsymbol{\nabla} f(\mathbf{x}^{(n)})$
9: $n \leftarrow n + 1$
10: **until** $\frac{|f(\mathbf{x}^{(n)}) - f(\mathbf{x}^{(n-1)})|}{|f(\mathbf{x}^{(n-1)})|} < \varepsilon_f$

Input:
$\mathbf{x}_0$ — initial guess, $\Delta\tau$ — fixed step size
ε_f — tolerance in the termination condition

Output: an approximation of the minimizer $\tilde{\mathbf{x}}$

Computational Tests

- Rosenbrock's "banana" function

$$f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

- Global minimizer

$$x_1 = x_2 = 1, \qquad f(1,1) = 0$$

- The function is known for its poor conditioning
  - eigenvalues of the Hessian $\nabla^2 f$ at the minimum:

$$\lambda_1 \approx 0.4, \qquad \lambda_2 \approx 1001.6$$

- Constraint

$$c(x_1, x_2) = -0.05\, x_1^4 - x_2 + 2.651605 = 0$$