Multiresolution in image restoration

3- Acceleration

Elisa Riccietti, Nelly Pustelnik

Acceleration techniques

- Second order methods: more expensive iterations but faster convergence rates
- Momentum methods: decreases number of iterations
- Structure exploiting methods: multilevel and block methods

Newton's direction

Let us consider the quadratic model of f in $x^{[k]}$, deriving from the second order Taylor's polynomial:

$$T_2(x^{[k]}, d) := f(x^{[k]}) + \nabla f(x^{[k]})^\top d + \frac{1}{2} d^\top H_f(x^{[k]}) d,$$

and let us assume $H_f(x^{[k]})$ **to be positive definite**.

Newton's direction d is the minimizer of $T_2(x^{[k]}, d)$, i.e., being $T_2(x^{[k]}, d)$ a positive definite quadratic function, it is the solution of Newton's system

$$H_f(x^{[k]}) d = -\nabla f(x^{[k]}). \tag{1}$$

Newton's algorithm

Given $x^{[0]}$, $\varepsilon > 0$, set $k = 0$.

While $\|\nabla f(x^{[k]})\| > \varepsilon$

1. Compute the search direction solving $H_f(x^{[k]})d^{[k]} = -\nabla f(x^{[k]})$.
2. Choose γ_k
3. Set $x^{[k+1]} = x^{[k]} + \gamma_k d^{[k]}$
4. Set $k = k + 1$

Newton's direction

The analytic expression of Newton's direction is then

$$d^{[k]} = -H_f(x^{[k]})^{-1}\nabla f(x^{[k]}).$$

$d^{[k]}$ is a descent direction:

$$\nabla f(x^{[k]})^\top d^{[k]} = \nabla f(x^{[k]})^\top \Big(- H_f(x^{[k]})^{-1}\nabla f(x^{[k]})\Big) = -\nabla f(x^{[k]})^\top H_f(x^{[k]})^{-1}\nabla f(x^{[k]}) < 0,$$

because $H_f(x^{[k]})^{-1}$ is positive definite.

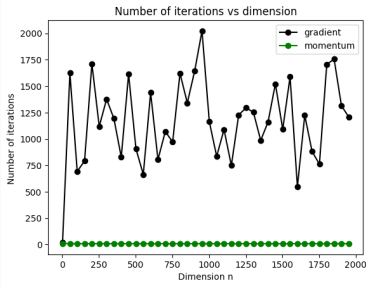
- fast local convergence (quadratic)
- expensive, requires:
  - computation of $H_f(x^{[k]})$ at each step
  - solution of the linear system (1) that may be expensive if the size of the problem is large.
  - Note that in practice $H_f(x^{[k]})$ is never explicitly inverted to compute such a direction, Newton's system is rather solved, because linear equation solution is faster and produces less round-off error than inversion and multiplication.

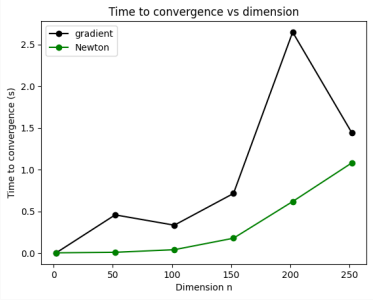
Local quadratic convergence

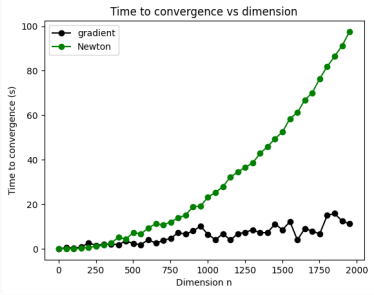
Local convergence of Newton's method. Let $x^\star$ be a minimizer for f, Ω be a neighbourhood of $x^\star$, $f \in \mathcal{C}^2(\Omega)$, $H_f(x^\star)$ positive definite and $H_f(x)$ Lipschitz continuous in Ω with Lipschitz constant L.
It exists $\rho > 0$ such that if $x^{[0]} \in \mathcal{B}_\rho(x^\star)$ then the sequence $\{x^{[k]}\}$ built by Newton's method is well-defined, converges to $x^\star$ quadratically and $\|\nabla f(x^{[k]})\|$ converges to 0 quadratically.

For the proof see optimization.tex, Theorem 2.3.1.

Newton vs gradient descent







Improvements over Newton's method

Inexact Newton's methods Solve the Newton's system inexactly by an iterative procedure

$$H_f(x^{[k]})d^{[k]} = -\nabla f(x^{[k]})$$

until

$$\|H_f(x^{[k]})d^{[k]} + \nabla f(x^{[k]})\| \leq \eta \|\nabla f(x^{[k]})\|$$

Quasi Newton's methods replace the Hessian by an approximation

$$B^{[k]}d^{[k]} = -\nabla f(x^{[k]}).$$

or even directly approximate the inverse:

$$d^{[k]} = -B^{[k]}\nabla f(x^{[k]}).$$

Sketched Newton method: restrict the Newton step to a lower-dimensional subspace. Let $S \in \mathbb{R}^{n \times s}$, $s \ll n$, be a sketching matrix, and assume that

$$p^{[k]} = S z^{[k]}.$$

Substituting this expression into the Newton equation and projecting onto the subspace generated by S gives

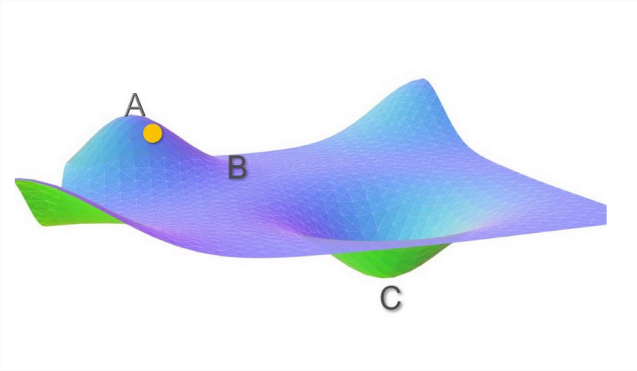
$$\boxed{S^\top H_f(x^{[k]}) S \, z^{[k]} = -S^\top \nabla f(x^{[k]}).}$$

Hence, instead of solving an $n \times n$ system, we solve an $s \times s$ system and update $x^{[k+1]} = x^{[k]} + S z^{[k]}$.

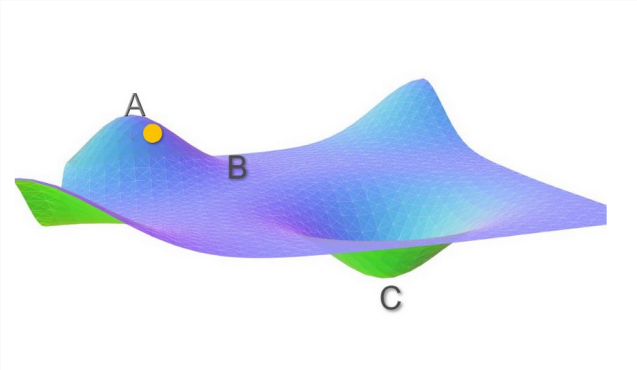
While Newton's method searches for the direction in $\mathbb{R}^n$, sketched Newton searches for an approximate direction in the lower-dimensional subspace $\mathrm{range}(S) \subset \mathbb{R}^n$.

The sketch therefore trades some accuracy in the Newton direction for a potentially substantial reduction in computational cost.

Momentum: keep memory of the past



Momentum: keep memory of the past

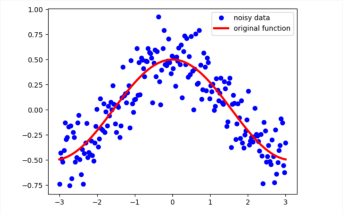


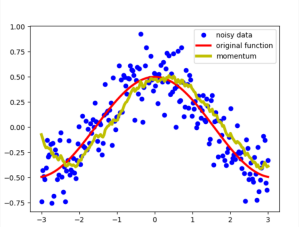
- Define the step as the average of past gradients instead of the gradient at the current iteration.
- Cannot consider all the gradients with equal weightage.
- Need to use some sort of weighted average

Exponential Moving Average (EMA)
Consider a noisy sequence $y(t)$. The EMA $s(t)$ for a series $y(t)$ may be calculated recursively as:

$$s(t) = \begin{cases} y(1) & \text{if } t = 1, \\ \beta s(t-1) + (1-\beta) y(t) & \text{if } t > 1. \end{cases}$$

where $\beta \in [0, 1]$ represents the degree of weighting increase. A lower β discounts older observations faster.





EMA for gradients

- New update:

$$s^{[k]} = \beta s^{[k-1]} + (1-\beta)\nabla f(x^{[k]})$$

  where $s^{[k]} = x^{[k+1]} - x^{[k]}$

- Often $\beta \to \beta_k$ and $(1-\beta) \to \gamma_k$

$$x^{[k+1]} = \underbrace{x^{[k]} - \gamma_k \nabla f(x^{[k]})}_{\text{standard gradient step}} + \underbrace{\beta_k (x^{[k]} - x^{[k-1]})}_{\text{momentum term}},$$

- Special cases:
  - $\beta_k = 0$ for all $k \in \mathbb{N}$: classical GD
  - $\gamma_k = \gamma$ and $\beta_k = \beta$: *heavy ball method*.

Heavy ball momentum

- By expanding the update:

$$x^{[k+1]} = x^{[k]} - \gamma \sum_{j=1}^{k} \beta^{k-j} \nabla f(x^{[k]})$$

each step is an exponentially decaying average of past gradients.

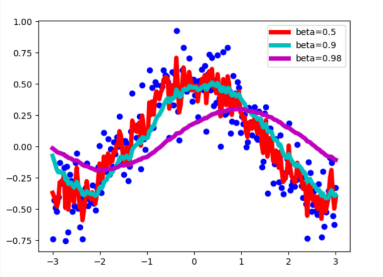
Heavy ball momentum

- By expanding the update:

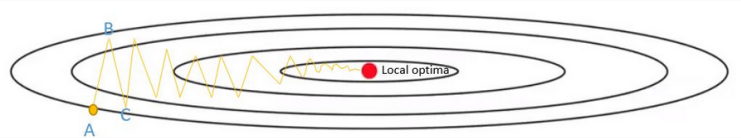
$$x^{[k+1]} = x^{[k]} - \gamma \sum_{j=1}^{k} \beta^{k-j} \nabla f(x^{[k]})$$

  each step is an exponentially decaying average of past gradients.

- Let's analyse the contribution of β. Assume $\gamma = 1$. At $k = 3$; $\nabla f(x^{[3]})$ will contribute 100% of its value.
  - $\beta = 0.1$: $\nabla f(x^{[2]})$ 10% and $\nabla f(x^{[1]})$ 1%:
  - $\beta = 0.9$: $\nabla f(x^{[2]})$ 90% and $\nabla f(x^{[1]})$ 81%.
  - Higher β: contribution from earlier gradients decreases slowly, accommodates more gradients from the past. Usually $\beta \sim 0.9$



How does momentum help?



- step size too small: small steps, convergence takes a lot of time even when the gradient is high.
- step size too high: the sequence oscillates around the minima

How does momentum fix this?

1. *All the past gradients have the same sign*: the summation term will become large and we will take large steps
2. *Different signs*: the summation term will become small and the steps will be small, damping the oscillations.

Convergence of GD with heavy-ball momentum for strongly convex quadratics

Consider the quadratic optimization problem

$$\min_{x \in \mathbb{R}^n} f(x) = \frac{1}{2} x^T Q x - b^T x,$$

where

$$Q = Q^T, \qquad \mu I \preceq Q \preceq L I.$$

The unique minimizer satisfies

$$Q x^\star = b.$$

The Heavy-Ball iteration converges linearly to $x^\star$ if

$$0 \leq \beta < 1 \qquad \text{and} \qquad 0 < \gamma < \frac{2(1+\beta)}{L}.$$

More precisely, under these conditions,

$$x^{[k]} \longrightarrow x^\star$$

at a linear rate.

Proof (I)

The Heavy-Ball iteration is

$$\begin{aligned} x^{[k+1]} &= x^{[k]} - \gamma \nabla f(x^{[k]}) + \beta(x^{[k]} - x^{[k-1]}) \\ &= x^{[k]} - \gamma(Qx^{[k]} - b) + \beta(x^{[k]} - x^{[k-1]}). \end{aligned}$$

Define the error

$$e^{[k]} = x^{[k]} - x^\star.$$

Since $Qx^\star = b$, we obtain

$$\begin{aligned} e^{[k+1]} &= x^{[k+1]} - x^\star \\ &= x^{[k]} - \gamma(Qx^{[k]} - b) + \beta(x^{[k]} - x^{[k-1]}) - x^\star \\ &= (I - \gamma Q + \beta I)e^{[k]} - \beta e^{[k-1]} \\ &= ((1+\beta)I - \gamma Q)e^{[k]} - \beta e^{[k-1]}. \end{aligned}$$

Proof (II)

Since Q is symmetric,

$$Q = U\Lambda U^T,$$

where

$$\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n),$$

and U is orthogonal.

Define

$$z^{[k]} = U^T e^{[k]}.$$

Multiplying the error equation by U^T gives

$$z^{[k+1]} = ((1+\beta)I - \gamma\Lambda)z^{[k]} - \beta z^{[k-1]}.$$

Therefore each coordinate evolves independently:

$$z_i^{[k+1]} = (1 + \beta - \gamma\lambda_i)z_i^{[k]} - \beta z_i^{[k-1]}.$$

Proof (III)

Introducing

$$M(\lambda_i) = \begin{bmatrix} 1+\beta-\gamma\lambda_i & -\beta \\ 1 & 0 \end{bmatrix},$$

we can rewrite

$$\begin{bmatrix} z_i^{[k+1]} \\ z_i^{[k]} \end{bmatrix} = M(\lambda_i) \begin{bmatrix} z_i^{[k]} \\ z_i^{[k-1]} \end{bmatrix} = \cdots = M(\lambda_i)^k \begin{bmatrix} z_i^{[1]} \\ z_i^{[0]} \end{bmatrix}$$

If $\rho(M(\lambda_i)) < 1$ then there exist constants $\tilde{C}_i$ and ρ_i such that

$$\|M(\lambda_i)^k\| \leq \tilde{C}_i \rho_i^k$$

Hence

$$|z_i^{[k]}| \leq \left\| \begin{bmatrix} z_i^{[k+1]} \\ z_i^{[k]} \end{bmatrix} \right\| \leq \tilde{C}_i \rho_i^k \left\| \begin{bmatrix} z_i^{[1]} \\ z_i^{[0]} \end{bmatrix} \right\| := C_i \rho_i^k$$

Proof (IV)

Therefore,

$$\|z^{[k]}\|^2 = \sum_{i=1}^{N} |z_i^{[k]}|^2 \leq \rho^{2k} \sum_{i=1}^{N} C_i^2.$$

Setting

$$C = \left(\sum_{i=1}^{N} C_i^2 \right)^{1/2},$$

we obtain

$$\|z^{[k]}\| \leq C\rho^k.$$

Because U is orthogonal,

$$\|e^{[k]}\| = \|z^{[k]}\|,$$

and therefore

$$\boxed{\|x^{[k]} - x^\star\| \leq C\rho^k.}$$

Stability condition

Studying the eigenvalues of M we obtain the sufficient and necessary conditions for having

$$\rho(M(\lambda)) < 1$$

for every eigenvalue:

$$0 \leq \beta < 1,$$

and

$$0 < \gamma < \frac{2(1+\beta)}{L}.$$

Optimal parameters

Polyak showed that the choice

$$\boxed{\gamma = \frac{4}{(\sqrt{L} + \sqrt{\mu})^2},}$$

and

$$\boxed{\beta = \left(\frac{\sqrt{L} - \sqrt{\mu}}{\sqrt{L} + \sqrt{\mu}} \right)^2}$$

minimizes the worst-case convergence factor.

The corresponding spectral radius is

$$\boxed{\rho = \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1},} \text{ where } \kappa = \frac{L}{\mu}.$$

Remark

The proof relies crucially on the Hessian being constant. For a general smooth strongly convex function, the error recursion becomes nonlinear:

$$e^{[k+1]} = e^{[k]} - \gamma\big(\nabla f(x^{[k]}) - \nabla f(x^\star)\big) + \beta(e^{[k]} - e^{[k-1]}),$$

and the above spectral analysis no longer applies. In fact, unlike gradient descent, the Heavy-Ball method with constant momentum is not globally convergent under only smoothness and strong convexity assumptions.

Comparison with GD

The convergence factor is $\rho_{\mathrm{GD}}(\gamma) = \max_{\lambda_i \in [\mu, L]} |1 - \gamma \lambda_i|$.

The optimal values of stepsize and convergence factor are

$$\boxed{\gamma^\star_{\mathrm{GD}} = \frac{2}{L+\mu}}, \quad \boxed{\rho^\star_{\mathrm{GD}} = \frac{L-\mu}{L+\mu} = \frac{\kappa - 1}{\kappa + 1}}.$$

Thus,

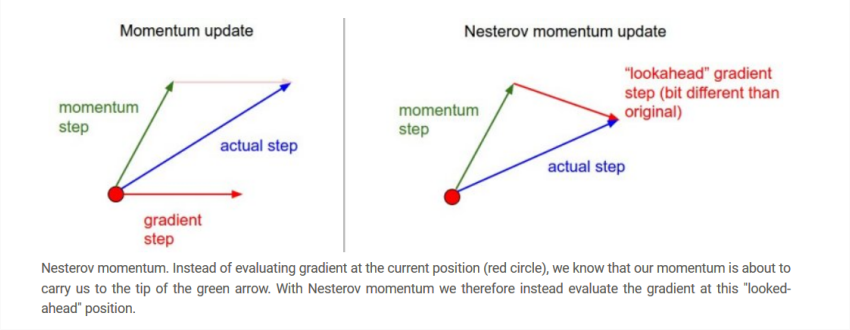
$$\|x^{[k]} - x^\star\| = \mathcal{O}\left(\left(\frac{\kappa - 1}{\kappa + 1}\right)^k\right).$$

Comparison with GD

	Gradient Descent	Heavy-Ball
Iteration	$x^{[k+1]} = x^{[k]} - \gamma \nabla f(x^{[k]})$	$x^{[k+1]} = x^{[k]} - \gamma \nabla f(x^{[k]}) + \beta(x^{[k]} - x^{[k-1]})$
Convergence conditions	$0 < \gamma < \frac{2}{L}$	$0 \leq \beta < 1, \quad 0 < \gamma < \frac{2(1+\beta)}{L}$
Optimal stepsize	$\frac{2}{L+\mu}$	$\frac{4}{(\sqrt{L}+\sqrt{\mu})^2}$
Optimal momentum	$-$	$\left(\frac{\sqrt{L}-\sqrt{\mu}}{\sqrt{L}+\sqrt{\mu}}\right)^2$
Convergence factor	$\frac{\kappa-1}{\kappa+1}$	$\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}$
Iteration complexity	$\mathcal{O}(\kappa \log(1/\varepsilon))$	$\mathcal{O}(\sqrt{\kappa} \log(1/\varepsilon))$

Nesterov accelerated gradient

- Treats the future approximate position $\tilde{x}^{[k]} = x^{[k]} + \beta_k(x^{[k]} - x^{[k-1]})$ as a "lookahead"
- Computes the gradient at $\tilde{x}^{[k]}$ instead of the old $x^{[k]}$



Nesterov momentum. Instead of evaluating gradient at the current position (red circle), we know that our momentum is about to carry us to the tip of the green arrow. With Nesterov momentum we therefore instead evaluate the gradient at this "looked-ahead" position.

- The Nesterov update is then $x^{[k+1]} = \tilde{x}_k^{[k]} - \gamma_k g(\tilde{x}^{[k]})$

Momentum vs Nesterov momentum

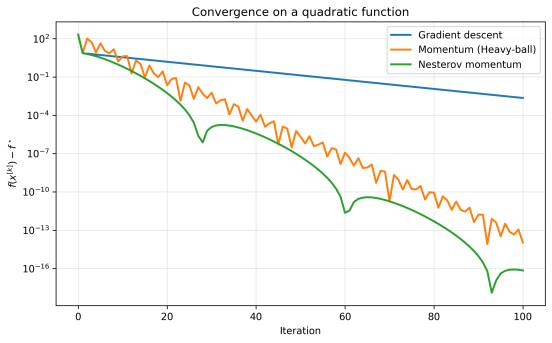
- In momentum: first gradient descent step and then momentum term
- In Nesterov: first momentum then gradient descent (with the gradient evaluated at $\tilde{x}^{[k]}$, not at $x^{[k]}$).
- Better convergence for Nesterov: distance to the optimal value decaying with a rate $O(1/k^2)$ with standard assumptions and linear convergence of the iterates for strongly convex functions, not necessarily quadratics

Method	**Assumptions**	**Objective value rate**
Gradient Descent	Convex + L-smooth	$\mathcal{O}(1/k)$
Heavy-Ball	Additional assumptions needed	No general $\mathcal{O}(1/k)$ result
Nesterov	Convex + L-smooth	$\mathcal{O}(1/k^2)$

Table 1: Comparison of convergence rates.

Accelerated gradient method: momentum

$$f : \mathbb{R}^2 \to \mathbb{R}, \qquad f(x) = \frac{1}{2} x^\top A x, \qquad A = \begin{pmatrix} 1 & 0 \\ 0 & 25 \end{pmatrix}.$$



Convergence of Nesterov - convex case

Assume that:

- $f : \mathbb{R}^N \to \mathbb{R}$ is convex;
- f is differentiable and ∇f is L-Lipschitz continuous;
- $x^\star \in \operatorname{argmin} f$ exists.

Nesterov's accelerated gradient method with

$$\gamma = \frac{1}{L}, \qquad \beta_k = \frac{t_k - 1}{t_{k+1}}, \qquad t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2}, \quad t_0 = 1$$

satisfies

$$f(x^{[k]}) - f^\star \leq \frac{2L\|x^{[0]} - x^\star\|^2}{(k+1)^2}.$$

Therefore,

$$\boxed{f(x^{[k]}) - f^\star = \mathcal{O}\left(\frac{1}{k^2}\right).}$$

Convergence of Nesterov - strongly convex case

Assume that:

- $f : \mathbb{R}^N \to \mathbb{R}$ is μ-strongly convex;
- f is differentiable and ∇f is L-Lipschitz continuous;
- $x^\star \in \operatorname{argmin} f$ exists.

Then Nesterov's accelerated gradient method, with $\gamma = \frac{1}{L}$ and $\beta = \frac{\sqrt{L}-\sqrt{\mu}}{\sqrt{L}+\sqrt{\mu}}$, satisfies

$$f(x^{[k]}) - f^\star \leq C \left(1 - \sqrt{\frac{\mu}{L}}\right)^k,$$

for some constant $C > 0$ and $\kappa = \frac{L}{\mu}$.

Convergence of Nesterov - strongly convex case

Therefore,

$$\boxed{f(x^{[k]}) - f^\star = \mathcal{O}\left(\left(1 - \frac{1}{\sqrt{\kappa}}\right)^k\right).}$$

Using strong convexity,

$$f(x) - f^\star \geq \frac{\mu}{2}\|x - x^\star\|^2,$$

we also obtain

$$\boxed{\|x^{[k]} - x^\star\| = \mathcal{O}\left(\left(1 - \frac{1}{\sqrt{\kappa}}\right)^{k/2}\right).}$$

Remark

Nesterov's convergence holds for any strongly convex function, not only for quadratics!

Coordinate Descent

Consider

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

Idea: instead of updating all the coordinates simultaneously, update one coordinate at a time.

Starting from $x^{[0]}$, at iteration k choose a coordinate $i_k \in \{1, \dots, N\}$ and solve

$$x_{i_k}^{[k+1]} \in \arg\min_{t \in \mathbb{R}} f\big(x_1^{[k]}, \dots, x_{i_k-1}^{[k]}, t, x_{i_k+1}^{[k]}, \dots, x_N^{[k]}\big).$$

The other coordinates remain unchanged:

$$x_i^{[k+1]} = x_i^{[k]}, \qquad i \neq i_k.$$

one-dimensional optimization problem at each iteration

Coordinate Descent: cyclic version

A common choice is to update the coordinates cyclically:

$$i_k = 1 + (k \bmod N).$$

Thus,

$$\begin{aligned}
x_1^{[k+1]} &\in \arg\min_t f(t, x_2^{[k]}, \ldots, x_N^{[k]}), \\
x_2^{[k+1]} &\in \arg\min_t f(x_1^{[k+1]}, t, x_3^{[k]}, \ldots, x_N^{[k]}), \\
&\vdots \\
x_N^{[k+1]} &\in \arg\min_t f(x_1^{[k+1]}, \ldots, x_{N-1}^{[k+1]}, t).
\end{aligned}$$

Other coordinate-selection strategies

- **randomized:** choose i_k randomly;
- **Gauss–Southwell:** choose a coordinate based on the magnitude of its partial derivative:

$$i_k \in \arg\max_i \left| \frac{\partial f}{\partial x_i}(x^{[k]}) \right|$$

Cost of coordinate selection strategies

- Coordinate descent reduces the cost of an update by modifying only one coordinate.
- Randomized selection is particularly cheap
- Gauss–Southwell is not necessarily cheaper per iteration than standard gradient descent
- Gauss–Southwell may achieve better progress per update, but selecting the coordinate can require additional computation.
- There are important settings where the full gradient can be maintained or computed efficiently, so finding the largest component is much less problematic than it might seem. Example: linear least-squares problems

Coordinate Descent with a gradient step

If the one-dimensional minimization is not solved exactly, we can perform a gradient step along one coordinate.

Let e_i denote the i-th canonical basis vector. Then

$$x^{[k+1]} = x^{[k]} - \gamma_k \frac{\partial f}{\partial x_{i_k}}\big(x^{[k]}\big) e_{i_k}.$$

Equivalently,

$$x_i^{[k+1]} = \begin{cases} x_i^{[k]} - \gamma_k \dfrac{\partial f}{\partial x_i}\big(x^{[k]}\big), & i = i_k, \\ x_i^{[k]}, & i \neq i_k. \end{cases}$$

Comparison with gradient descent:

Gradient Descent	Coordinate Descent
update all coordinates	update one coordinate
costlier iteration	cheaper iteration
full gradient	one partial derivative

Block Coordinate Descent

Partition the variables into p blocks:

$$x = \begin{pmatrix} x_1 \\ \vdots \\ x_p \end{pmatrix}, \qquad x_j \in \mathbb{R}^{n_j}, \qquad \sum_{j=1}^{p} n_j = N.$$

At iteration k, choose a block j_k and update only that block:

$$x_{j_k}^{[k+1]} \in \arg\min_{z \in \mathbb{R}^{n_{j_k}}} f\big(x_1^{[k]}, \ldots, x_{j_k-1}^{[k]}, z, x_{j_k+1}^{[k]}, \ldots, x_p^{[k]}\big).$$

The remaining blocks satisfy

$$x_j^{[k+1]} = x_j^{[k]}, \qquad j \neq j_k.$$

Coordinate Descent = Block Coordinate Descent with $n_j = 1$.

Block Coordinate Descent: gradient update

Instead of minimizing exactly over the selected block, one can perform a gradient step:

$$x_{j_k}^{[k+1]} = x_{j_k}^{[k]} - \gamma_k \nabla_{j_k} f\big(x^{[k]}\big),$$

while

$$x_j^{[k+1]} = x_j^{[k]}, \qquad j \neq j_k.$$

Equivalently,

$$x^{[k+1]} = x^{[k]} - \gamma_k P_{j_k} \nabla f(x^{[k]}),$$

where P_{j_k} projects onto the selected block.

Exercise

Consider

$$f(x) = \|Ax - z\|^2 + \lambda \|Lx\|^2$$

with $A \in \mathbb{R}^{M \times N}$ and $L \in \mathbb{R}^{N \times N}$, write the iterations of gradient descent, gradient with momentum and block coordinate descent. What are the rate of convergence of the gradient method and momentum for this problem?