Second order methods for the solution of large-scale machine learning problems

Elisa Riccietti

Joint work with: H. Calandra (TOTAL), S. Gratton (INPT-IRIT), X. Vasseur (ISAE-SUPAERO)

PRIMO Talks - 15 October 2020

Context: continuous optimization problems in learning

$$\min_x f(x) \quad \Rightarrow \quad \min_{x\in\mathbb{R}^n} f(x) = \frac{1}{2}\|F(x)\|^2 = \frac{1}{2}\sum_{i=1}^m F_i(x)^2$$

Large scale problems

$$F : \mathbb{R}^n \to \mathbb{R}^m$$

- F has a large number of components: large m (ex: classification of large datasets)
- F has a large number of unknowns: large n (ex: deep learning)

Common objective

Exploit objective function approximations to reduce computational cost of the solution

Context: continuous optimization problems in learning

$$\min_x f(x) \quad \Rightarrow \quad \min_{x\in\mathbb{R}^n} f(x) = \frac{1}{2}\|F(x)\|^2 = \frac{1}{2}\sum_{i=1}^m F_i(x)^2$$

Large scale problems

$$F : \mathbb{R}^n \to \mathbb{R}^m$$

- F has a large number of components: large m (ex: classification of large datasets) $\Rightarrow$ **subsampled methods**

  Bellavia, S. and Gratton, S. and Riccietti, E.. *A Levenberg-Marquardt method for large nonlinear least-squares problems with noisy functions and gradients*. **Numer. Math. (2018)**.

- F has a large number of unknowns: large n (ex: deep learning) $\Rightarrow$ **multilevel methods**

Common objective

Exploit objective function approximations to reduce computational cost of the solution

Outline

- Part I:
  - high-order optimization methods
  - their multilevel extension
- Part II:
  - second order multilevel training methods for artificial neural networks
  - Application to the solution of PDEs

High-order optimization methods

The optimization methods

We consider large-scale nonlinear unconstrained optimization problems:

$$\min_x f(x)$$

The optimization methods

We consider large-scale nonlinear unconstrained optimization problems:

$$\min_x f(x)$$

Classical iterative optimization methods:

$$f(x_k + s) \simeq T_{2,k}(x_k, s)$$

with $T_{2,k}(x_k, s)$ Taylor model of order 2.
At each iteration we compute a step s_k to update the iterate:

$$\min_s m_k(x_k, s) = T_{2,k}(x_k, s) + r(\lambda_k), \qquad \lambda_k > 0$$

$r(\lambda_k)$ regularization term.

Classical examples

- Trust region (TR) method:

$$m_k(x_k, s) = f(x_k) + s^T \nabla f(x_k) + \frac{1}{2} s^T \nabla^2 f(x_k) s + \frac{\lambda_k}{2} \|s\|^2$$

- Adaptive Cubic Regularization (ARC) method:

$$m_k(x_k, s) = f(x_k) + s^T \nabla f(x_k) + \frac{1}{2} s^T \nabla^2 f(x_k) s + \frac{\lambda_k}{3} \|s\|^3$$

Cubic regularization of Newton method and its global performance, Y. Nesterov and B. Polyak, 2006

Adaptive cubic regularization methods for unconstrained optimization, C. Cartis, N. Gould, Ph. Toint, 2009

Classical examples

- Trust region (TR) method: Complexity: $O(\epsilon^{-2})$

$$m_k(x_k, s) = f(x_k) + s^T \nabla f(x_k) + \frac{1}{2} s^T \nabla^2 f(x_k) s + \frac{\lambda_k}{2} \|s\|^2$$

- Adaptive Cubic Regularization (ARC) method: Complexity: $O(\epsilon^{-3/2})$

$$m_k(x_k, s) = f(x_k) + s^T \nabla f(x_k) + \frac{1}{2} s^T \nabla^2 f(x_k) s + \frac{\lambda_k}{3} \|s\|^3$$

Cubic regularization of Newton method and its global performance, Y. Nesterov and B. Polyak, 2006

Adaptive cubic regularization methods for unconstrained optimization, C. Cartis, N. Gould, Ph. Toint, 2009

Worst case complexity

Given $\epsilon > 0$, compute the number of iterations required to achieve an iterate x_k such that $\|\nabla f(x_k)\| \leq \epsilon$: $\quad k = O(\epsilon^{?})$

Family of higher-order methods generalizing ARC

Model of order $q \Rightarrow$ Complexity: $O(\epsilon^{-(q+1)/q})$

$$m_{q,k}(x_k, s) = T_{q,k}(x_k, s) + \frac{\lambda_k}{q+1}\|s\|^{q+1}, \qquad \lambda_k > 0$$

$$T_{q,k}(x_k, s) = \sum_{i=1}^{q} \frac{1}{i!}\nabla^i f(x_k)(\overbrace{s, \ldots, s}^{i \text{ times}})$$

Unifying framework for global convergence and worst-case complexity is presented $\Rightarrow$ ARC $q = 2$.

Worst-case evaluation complexity for unconstrained nonlinear optimization using high-order regularized models, E. G. Birgin, J. L. Gardenghi, J. M. Martínez, S. A. Santos and Ph. L. Toint, 2017

ARq$(x_0, \lambda_0, \epsilon)$

Given $0 < \eta_1 < 1$, set $k = 0$
while $\|\nabla_x f(x_k)\| > \epsilon$ **do**
- **Initialization:** Define $m_{q,k}(x_k, s) = T_{q,k}(x_k, s) + \frac{\lambda_k}{q+1}\|s\|^{q+1}$
- **Model minimization:** Find a step s_k that sufficiently reduces the model $m_{q,k}$
- **Acceptance of the trial point:** Compute

$$\rho_k = \frac{f(x_k) - f(x_k + s_k)}{T_{q,k}(x_k, 0) - T_{q,k}(x_k, s_k)}$$

if $\rho_k \geq \eta_1$ **then**
$x_{k+1} = x_k + s_k$, decrease λ_k,
else
$x_{k+1} = x_k$, increase λ_k.
end if
$k = k + 1$
end while

Bottleneck: Subproblem solution

Solving

$$\min_s T_{q,k}(x_k, s) + \frac{\lambda_k}{q+1}\|s\|^{q+1}$$

represents greatest cost per iteration, which depends on the size of the problem.

Our proposition: family of multilevel methods using high-order models

Calandra, H. and Gratton, S. and Riccietti, E. and Vasseur, X.. *On high-order multilevel optimization strategies*. **Submitted to SIAM J. Optim. (2019)**.

Multilevel strategy

Hierarchy of problems

- $\{f^\ell(x^\ell)\}$, $x^\ell \in \mathbb{R}^{n_\ell}$
- $n_{\ell-1} < n_\ell \Rightarrow f^{\ell-1}$ is cheaper to optimize compared with f^ℓ
- $\mu^{\ell-1}$ model for $f^{\ell-1}$

x_k^ℓ

Multilevel strategy

Hierarchy of problems

- $\{f^\ell(x^\ell)\}$, $x^\ell \in \mathbb{R}^{n_\ell}$
- $n_{\ell-1} < n_\ell \Rightarrow f^{\ell-1}$ is cheaper to optimize compared with f^ℓ
- $\mu^{\ell-1}$ model for $f^{\ell-1}$

$$
\begin{array}{c}
x_k^\ell \\
R^\ell \Big\downarrow \\
x_{0,k}^{\ell-1} := R^\ell x_k^\ell
\end{array}
$$

Multilevel strategy

Hierarchy of problems

- $\{f^\ell(x^\ell)\}$, $x^\ell \in \mathbb{R}^{n_\ell}$
- $n_{\ell-1} < n_\ell \Rightarrow f^{\ell-1}$ is cheaper to optimize compared with f^ℓ
- $\mu^{\ell-1}$ model for $f^{\ell-1}$

$$
\begin{tikzcd}
x_k^\ell \arrow[d, "R^\ell"'] & \\
x_{0,k}^{\ell-1} := R^\ell x_k^\ell \arrow[r, "\min_x \mu^{\ell-1}(x)"] & x_{*,k}^{\ell-1}
\end{tikzcd}
$$

Multilevel strategy

Hierarchy of problems

- $\{f^\ell(x^\ell)\}$, $x^\ell \in \mathbb{R}^{n_\ell}$
- $n_{\ell-1} < n_\ell \Rightarrow f^{\ell-1}$ is cheaper to optimize compared with f^ℓ
- $\mu^{\ell-1}$ model for $f^{\ell-1}$

$$
\begin{tikzcd}
x_k^\ell \arrow[d, "R^\ell"'] & & x_{k+1}^\ell = x_k^\ell + s_k^\ell \\
x_{0,k}^{\ell-1} := R^\ell x_k^\ell \arrow[rr, "\min_x \mu^{\ell-1}(x)"] & & x_{*,k}^{\ell-1} \arrow[u, "s_k^\ell = P^\ell(x_{*,k}^{\ell-1} - x_{0,k}^{\ell-1})"']
\end{tikzcd}
$$

The procedure is recursive: more levels can be used

MARq$(x_0, \lambda_0, \epsilon)$

Given $0 < \eta_1 < 1$, set $k = 0$
while $\|\nabla_x f(x_k)\| > \epsilon$ **do**
- **Initialization:** Define the model $m_{q,k}(x_k, s) = T_{q,k}(x_k, s) + \frac{\lambda_k}{q+1}\|s\|^{q+1}$ and the lower level model
- **Model minimization:** Choose wether to use Taylor model or to recursively minimize the lower level model to get s_k
- **Acceptance of the trial point:** Compute

$$\rho_k = \frac{f(x_k) - f(x_k + s_k)}{T_{q,k}(x_k, 0) - T_{q,k}(x_k, s_k)}$$

  if $\rho_k \geq \eta_1$ **then**
    $x_{k+1} = x_k + s_k$, decrease λ_k,
  else
    $x_{k+1} = x_k$, increase λ_k.
  end if
  $k = k + 1$
end while

Lower level model

When to use the lower level model?

- Choose lower level model $\mu^{\ell-1}$ if
  - if $\|\nabla \mu_{q,k}^{\ell-1}(x_{0,k}^{\ell-1})\| = \|R^\ell \nabla f^\ell(x_k^\ell)\| \geq \kappa \|\nabla f^\ell(x_k^\ell)\|$, $\kappa > 0$
  - if $\|\nabla \mu_{q,k}^{\ell-1}(x_{0,k}^{\ell-1})\| > \epsilon^\ell$
- Minimize regularized Taylor model otherwise.

How to define the lower level model?

Modify $f^{\ell-1}$ to ensure coherence among levels

Coherence between levels, $q = 1$

Let $x_{0,k}^{\ell-1} = Rx_k^\ell$. Model with first order correction:

$$\mu_{1,k}^{\ell-1}(x_{0,k}^{\ell-1}, s^{\ell-1}) = f^{\ell-1}(x_{0,k}^{\ell-1} + s^{\ell-1}) + (R^\ell \nabla f^\ell(x_k^l) - \nabla f^{\ell-1}(x_k^{\ell-1}))^T s^{\ell-1}$$

This ensures that

$$\nabla \mu_{1,k}^{\ell-1}(x_{0,k}^{\ell-1}) = R^\ell \nabla f^\ell(x_k^\ell)$$

$\rightarrow$ first-order behaviours of f^ℓ and $\mu^{\ell-1}$ are coherent in a neighbourhood of the current approximation. If $s^\ell = P^\ell s^{\ell-1}$

$$\nabla f^\ell(x_k^\ell)^T s^\ell = \nabla f^\ell(x_k^\ell)^T P^\ell s^{\ell-1} = \nabla \mu_{1,k}^{\ell-1}(x_{0,k}^{\ell-1})^T s^{\ell-1}.$$

Coherence between levels, $q = 2$

Let $x_{0,k}^{\ell-1} = Rx_k^\ell$. We define $\mu_{2,k}^{\ell-1}$ as

$$\begin{aligned}\mu_{2,k}^{\ell-1}(x_{0,k}^{\ell-1}, s^{\ell-1}) = {} & f^{\ell-1}(x_{0,k}^{\ell-1} + s^{\ell-1}) \\ & + (R^\ell \nabla f^\ell(x_k^l) - \nabla f^{\ell-1}(x_k^{\ell-1}))^T s^{\ell-1} \\ & + \frac{1}{2}(s^{\ell-1})^T((R^\ell)^T \nabla^2 f^\ell(x_k^l) P^\ell - \nabla^2 f^{\ell-1}(x_k^{\ell-1}))s^{\ell-1}\end{aligned}$$

$\rightarrow$ We can generalize this up to order q to have the behaviours of f^ℓ and $\mu_{q,k}^{\ell-1}$ to be coherent up to order q in a neighbourhood of the current approximation.

Coherence up to order q

We define

$$\mu_{q,k}^{\ell-1}(x_{0,k}^{\ell-1}, s^{\ell-1}) = f^{\ell-1}(x_{0,k}^{\ell-1} + s^{\ell-1}) + \sum_{i=1}^{q} \frac{1}{i!}[\mathcal{R}(\nabla^i f^\ell(x_k)) - \nabla^i f^{\ell-1}(x_{0,k}^{\ell-1})] \underbrace{(s^{\ell-1}, \dots, s^{\ell-1})}_{i \text{ times}},$$

where $\mathcal{R}(\nabla^i f^\ell(x_k^\ell))$ is such that for all $i = 1, \dots, q$ and $s_1^{\ell-1}, \dots, s_i^{\ell-1} \in \mathbb{R}^{n_{l-1}}$

$$[\mathcal{R}(\nabla^i f^\ell(x_k^\ell))](s_1^{\ell-1}, \dots, s_i^{\ell-1}) := \nabla^i f^\ell(x_k^\ell, P^\ell s_1^{\ell-1}, \dots, P^\ell s_i^{\ell-1}),$$

where $\nabla^i f^\ell$ denotes the i-th order tensor of f^ℓ.

Theoretical results: Assumptions

Assumption 1

Let us assume that for all ℓ the q-th derivative tensors of f^ℓ are Lipschitz continuous.

Assumption 2

There exist strictly positive scalars $\kappa_{EB}, \rho > 0$ such that

$$\mathrm{dist}(x, \mathcal{X}) \leq \kappa_{EB} \|\nabla_x f(x)\|, \quad \forall x \in \mathcal{N}(\mathcal{X}, \rho),$$

where $\mathcal{X}$ is the set of second-order critical points of f, $\mathrm{dist}(x, \mathcal{X})$ denotes the distance of x to $\mathcal{X}$ and $\mathcal{N}(\mathcal{X}, \rho) = \{x \mid \mathrm{dist}(x, \mathcal{X}) \leq \rho\}$.

On the Quadratic Convergence of the Cubic Regularization Method under a Local Error Bound Condition, M.C. Yue, Z. Zhou, and A.M.C. So, 2018

Theoretical results: global convergence

Theorem

Let Assumption 1 hold. Then, the sequence of iterates generated by the algorithm converges globally to a first-order stationary point:

$$\lim_{k \to \infty} \|\nabla f(x_k)\| = 0$$

Theoretical results: global convergence

Theorem

Let Assumption 1 hold. Then, the sequence of iterates generated by the algorithm converges globally to a first-order stationary point:

$$\lim_{k \to \infty} \|\nabla f(x_k)\| = 0$$

- E. G. Birgin, J. L. Gardenghi, J. M. Martínez, S. A. Santos and Ph. L. Toint, 2017: generalized to multilevel framework
- S. Gratton, A. Sartenaer and Ph. L. Toint, 2008: extended to higher-order models and simplified

Theoretical results: complexity

Theorem

Let Assumption 1 hold. Let f_{low} be a lower bound on f. Then, the method requires at most

$$K_3 \frac{(f(x_{k_1}) - f_{low})}{\epsilon^{\frac{q+1}{q}}} \left(1 + \frac{|\log \gamma_1|}{\log \gamma_3}\right) + \frac{1}{\log \gamma_3} \log\left(\frac{\lambda_{\max}}{\lambda_0}\right)$$

iterations to achieve an iterate x_k such that $\|\nabla f(x_k)\| \leq \epsilon$, where

$$K_3 := \frac{q+1}{\eta_1 \lambda_{\min}} L^{1/q}.$$

E. G. Birgin, J. L. Gardenghi, J. M. Martínez, S. A. Santos and Ph. L. Toint, 2017: $k = O(\epsilon^{-\frac{q+1}{q}})$ Complexity of standard method is maintained

Theoretical result: local convergence

Theorem

Let Assumptions 1 and 2 hold. Assume that $\mathcal{L}(f(x_k))$ is bounded for some $k \geq 0$ and that it exists an accumulation point x^* such that $x^* \in \mathcal{X}$. Then, the whole sequence $\{x_k\}$ converges to x^* and it exist strictly positive constants $c \in \mathbb{R}$ and $\bar{k} \in \mathbb{N}$ such that:

$$\frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|^q} \leq c, \quad \forall k \geq \bar{k}.$$

Theoretical result: local convergence

Theorem

Let Assumptions 1 and 2 hold. Assume that $\mathcal{L}(f(x_k))$ is bounded for some $k \geq 0$ and that it exists an accumulation point x^* such that $x^* \in \mathcal{X}$. Then, the whole sequence $\{x_k\}$ converges to x^* and it exist strictly positive constants $c \in \mathbb{R}$ and $\bar{k} \in \mathbb{N}$ such that:

$$\frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|^q} \leq c, \quad \forall k \geq \bar{k}.$$

- E. G. Birgin, J. L. Gardenghi, J. M. Martínez, S. A. Santos and Ph. L. Toint, 2017: local convergence not proved
- S. Gratton, A. Sartenaer and Ph. L. Toint, 2008: local convergence not proved
- M.C. Yue, Z. Zhou, and A.M.C. So, 2018: generalized to $q > 2$

Numerical results on the solution of PDEs

$$\begin{cases} -\Delta u(z) + e^{u(z)} = g(z) & \text{in } \Omega \subset \mathbb{R}^d, \\ u(z) = 0 & \text{on } \partial\Omega, \end{cases}$$

The following nonlinear minimization problem is then solved:

$$\min_{u \in \mathbb{R}^{n^d}} \frac{1}{2} u^T A u + \|e^{u/2}\|^2 - g^T u,$$

which is equivalent to the system $Au + e^u = g$.

- Coarse approximations: coarser discretization of the problem (2^d times lower dimension).

4 levels methods of order $q = 2$, $d = 2$

$$\min_{u \in \mathbb{R}^{n^2}} \frac{1}{2} u^T A u + \|e^{u/2}\|^2 - g^T u,$$

	$d = 2,\ q = 2$	$n = 256$		$n = 512$	
		AR2	MAR2	AR2	MAR2
$\bar{u}_1$	it_T/it_f	11/11	7/2	23/23	15/4
	save		**2.2**		**4.1**
$\bar{u}_2$	it_T/it_f	27/27	13/4	56/56	22/6
	save		**3.9**		**6.1**

$\bar{u}_i$: strating point
it_T/it_f: total iterations/fine iterations
save: save in CPU time

4 levels methods of order $q = 3$, $d = 1$

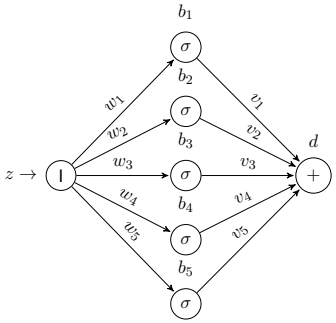
$$\min_{u \in \mathbb{R}^n} \frac{1}{2} u^T A u + \|e^{u/2}\|^2 - g^T u,$$

	$d = 1,\ q = 3$	$n = 1024$		$n = 4096$	
		AR3	MAR3	AR3	MAR3
$\bar{u}_1$	it_T/it_f	7/7	9/2	18/18	15/2
	save		**2.5**		**4.3**
$\bar{u}_2$	it_T/it_f	23/23	14/1	34/34	20/5
	save		**4.1**		**4.4**

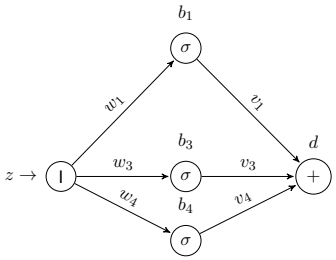
$\bar{u}_i$: strating point
it_T/it_f: total iterations/fine iterations
save: save in CPU time

Multilevel training methods

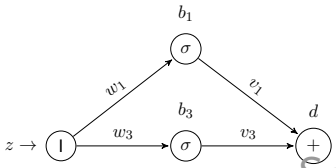
How to exploit multilevel method for training of ANNs?



$$R_1 \Downarrow P_1 \Uparrow$$

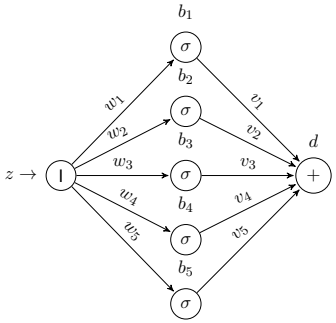


$$R_2 \Downarrow P_2 \Uparrow$$

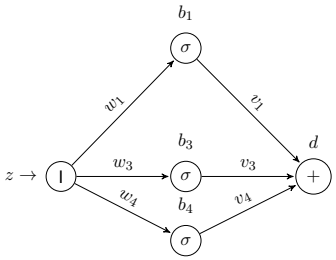


Large-scale problem

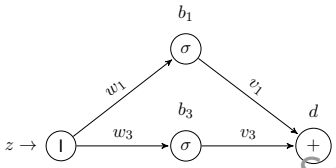
How to exploit multilevel method for training of ANNs?



$R_1 \Downarrow P_1 \Uparrow$



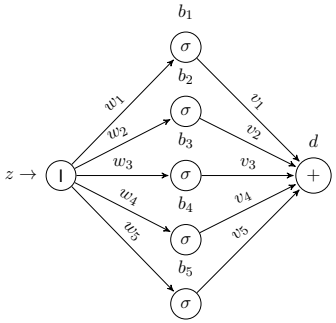
$R_2 \Downarrow P_2 \Uparrow$



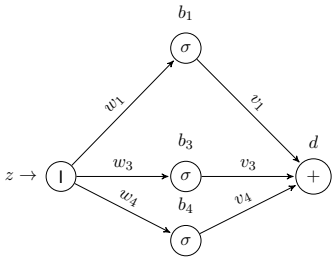
Large-scale problem

- How to build the hierarchy of problems? The variables to be optimized are the network's weights:
  NO evident geometrical structure to exploit!

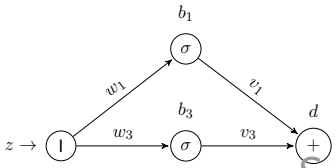
How to exploit multilevel method for training of ANNs?



$R_1 \Downarrow P_1 \Uparrow$



$R_2 \Downarrow P_2 \Uparrow$



Large-scale problem

- How to build the hierarchy of problems? The variables to be optimized are the network's weights:
  NO evident geometrical structure to exploit!
- The network possesses a purely algebraic structure: can we exploit it?

How do we select the hierarchy of variables?

Algebraic multigrid (AMG): C/F splitting

Ruge and Stueben C/F splitting for $Ax = b$

- Two variables i, j are said to be *coupled* if $a_{i,j} \neq 0$.
- We say that a variable i is strongly coupled to another variable j, if $-a_{i,j} \geq \epsilon \max_{a_{i,k}<0} |a_{i,k}|$ for a fixed $0 < \epsilon < 1$, usually $\epsilon = 0.25$.

Prolongation-Restriction operators

$P = [I; \Delta]$, $R = P^T$.

Which matrix should we use?

Assume to use a second-order model ($B_k \sim \nabla^2 f(x_k)$):

$$m_k(x_k, s) = f(x_k) + s^T \nabla f(x_k) + \frac{1}{2} s^T B_k s + \frac{\lambda_k}{3} \|s\|^3$$

At each iteration we have to solve a linear system of the form:

$$(B_k + \tilde{\lambda}_k I)s = -\nabla f(x_k), \quad \tilde{\lambda}_k > 0.$$

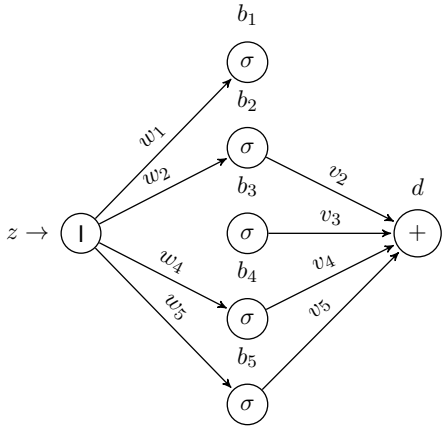
As in AMG for linear systems, we use information contained in matrix B_k.

Which matrix should we use?

Remark

Variables are coupled!

$\{w_i, b_i, v_i\}$



We do not use the full matrix B_k and we define A as:

$$B_k = \begin{bmatrix} f_{v,v} & .. & .. \\ .. & f_{w,w} & .. \\ .. & .. & f_{b,b} \end{bmatrix} \rightarrow A = \frac{f_{v,v}}{\|f_{v,v}\|_\infty} + \frac{f_{w,w}}{\|f_{w,w}\|_\infty} + \frac{f_{b,b}}{\|f_{b,b}\|_\infty}$$

We define the coarse/fine splitting based on the auxiliary matrix A.

Calandra, H. and Gratton, S. and Riccietti, E. and Vasseur, X.. *On a multilevel Levenberg-Marquardt method for the training of artificial neural networks and its application to the solution of partial differential equations*. **Optim. Methods Softw. (2020)**.

Application: solution of PDEs with ANNs

- Overcoming the curse of dimensionality in the numerical approximation of high-dimensional semilinear parabolic partial differential equations (2020).
- The Deep Ritz method: A deep learning-based numerical algorithm for solving variational problems (2018)
- A proof that deep artificial neural networks overcome the curse of dimensionality in the numerical approximation of Kolmogorov partial differential equations with constant diffusion and nonlinear drift coefficients (2018).
- Analysis of the generalization error: Empirical risk minimization over deep artificial neural networks overcomes the curse of dimensionality in the numerical approximation of Black-Scholes partial differential equations (2019).
- Solving stochastic differential equations and Kolmogorov equations by means of deep learning (2018).
- Deep Neural Networks motivated by Partial Differential Equations (2019).

Why try to solve PDEs with ANNs?

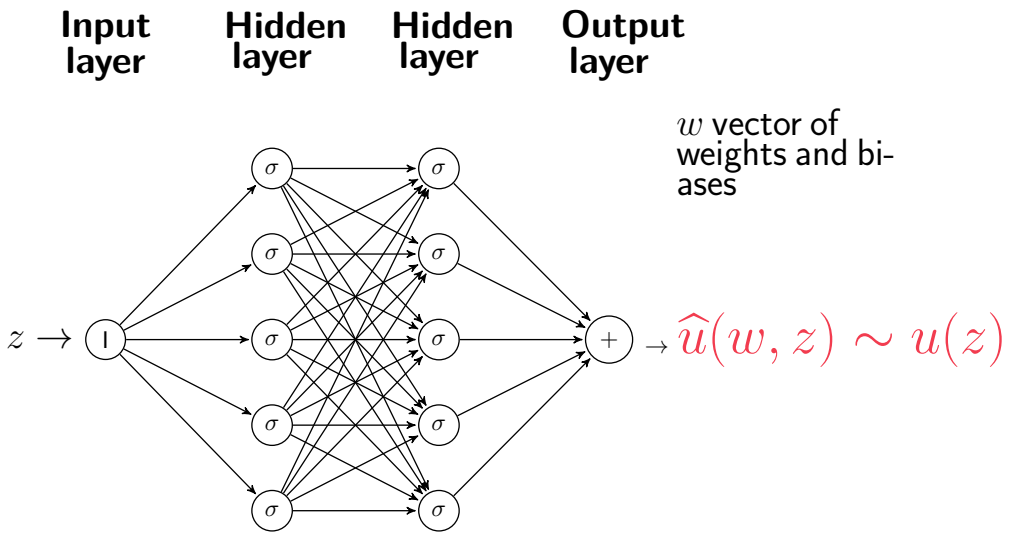
Compared with classical approaches (FDM, FEM), approaches using ANNs present the following advantages.

Advantages of ANNs over classical approaches

- Natural approach for nonlinear equations
- Provides analytical expression of the approximate solution which is continuously differentiable
- The solution is meshless, well suited for problems with complex geometries
- The training is highly parallelizable on GPU
- Allows to alleviate the effect of the curse of dimensionality (highly effective for more than 4 dimensions)

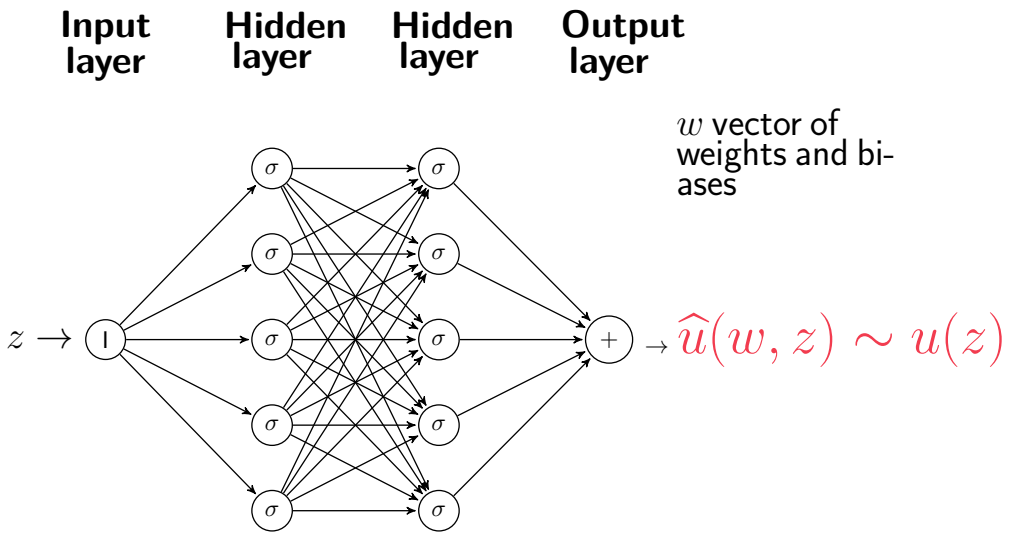
Our approach: express the solution as a neural network

1D case: $D(z, u(z)) = g(z),\ z \in (a, b) \quad u(a) = A,\ u(b) = B$



Our approach: express the solution as a neural network

1D case: $D(z, u(z)) = g(z),\ z \in (a, b) \quad u(a) = A,\ u(b) = B$



Training problem: find the network weights w by minimizing

$$\min_w \frac{1}{2T} \sum_{t=1}^{T} \Big(\underbrace{D(z, \widehat{u}(w, z_t)) - g(z_t)}_{\text{Equation residual}} \Big)^2 + \lambda_p \Big(\underbrace{(\widehat{u}(w, a) - A)^2 + (\widehat{u}(w, b) - B)^2}_{\text{Boundary conditions}} \Big)$$

Least-squares problem $\rightarrow$ multilevel Levenberg-Marquardt method

Classical Levenberg-Marquardt method

$$\min_x f(x) = \|F(x)\|^2$$

- Given $x_k \in \mathbb{R}^n$ and $\lambda_k \geq 0$, find the step $s_k \in \mathbb{R}^n$ minimizing

$$\begin{aligned} m_k^{LM}(x_k, s) &= \frac{1}{2}\|F(x_k) + J(x_k)s\|^2 + \frac{1}{2}\lambda_k\|s\|^2 \\ &= f(x_k) + \nabla f(x_k)^T s + \frac{1}{2}s^T B_k s + \frac{1}{2}\lambda_k\|s\|^2 \\ B_k &= J(x_k)^T J(x_k) \sim \nabla^2 f(x_k) \end{aligned}$$

- Compute

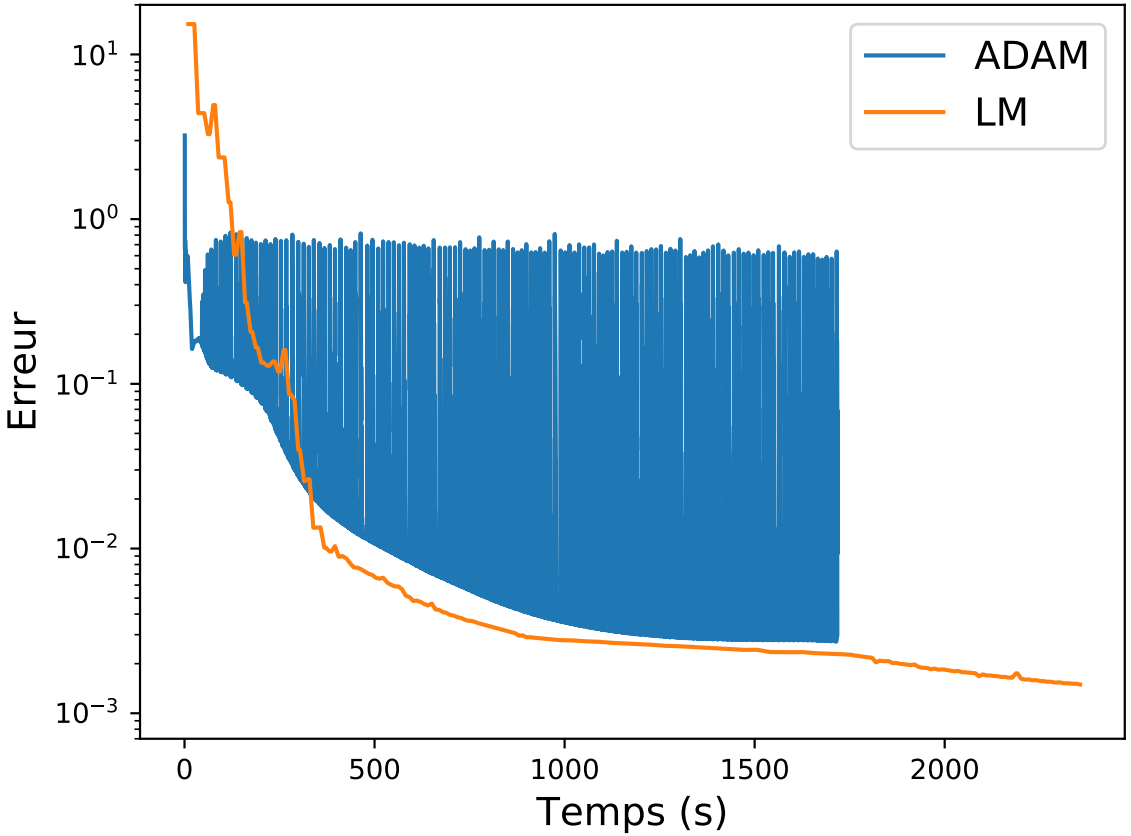
$$\rho_k(s_k) = \frac{f(x_k) - f(x_k + s_k)}{m_k^{LM}(x_k, 0) - m_k^{LM}(x_k, s_k)}.$$

- Step acceptance. Given $\eta \in (0, 1)$:
  - If $\rho_k < \eta$ reject the step: $x_{k+1} = x_k$ and increase λ_k.
  - If $\rho_k \geq \eta$ accept the step: $x_{k+1} = x_k + s_k$.

Solution of PDEs: Numerical example

Poisson's equation
(2D, $n = 4096$)

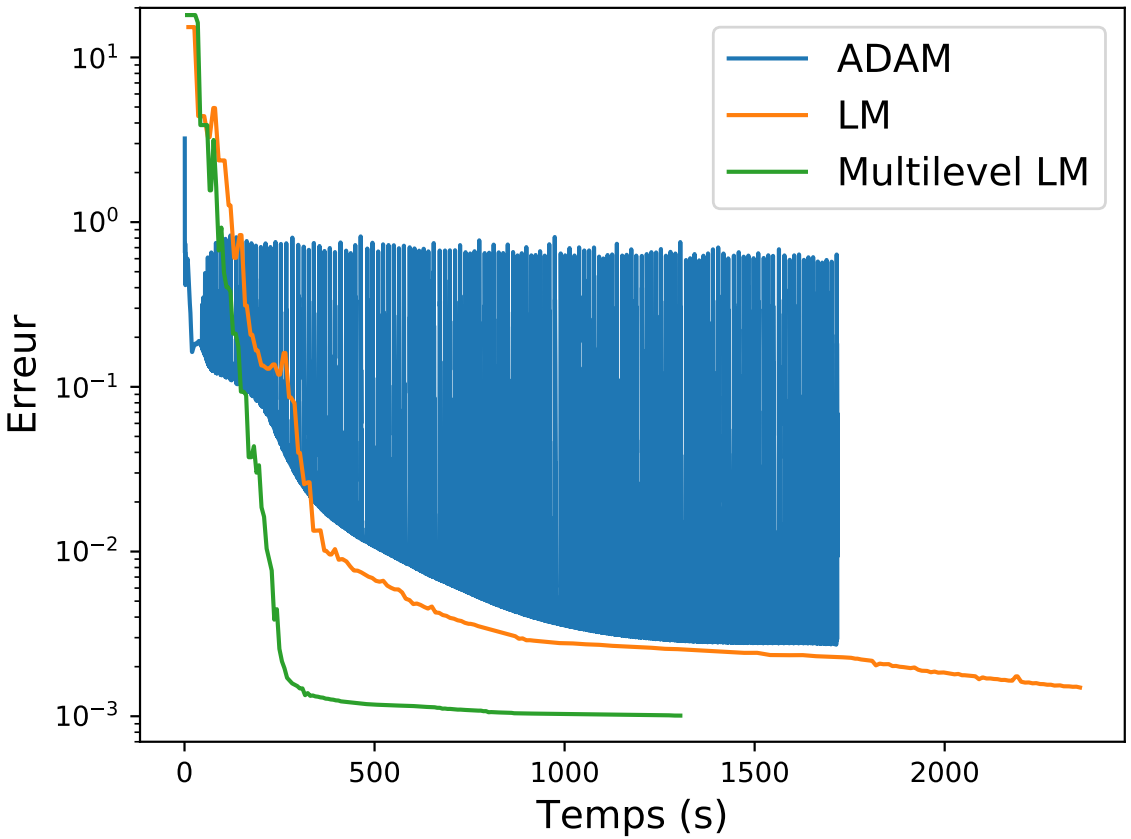
Method	ADAM	1 level
Iterations	10000	200



Solution of PDEs: Numerical example

Poisson's equation
(2D, $n = 4096$)

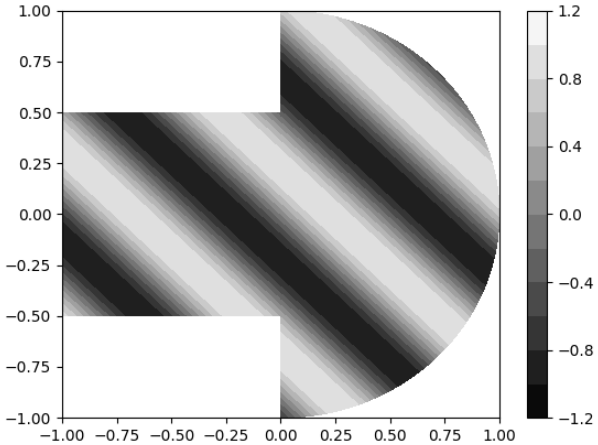
Method	ADAM	1 level	2 levels
Iterations	10000	200	200

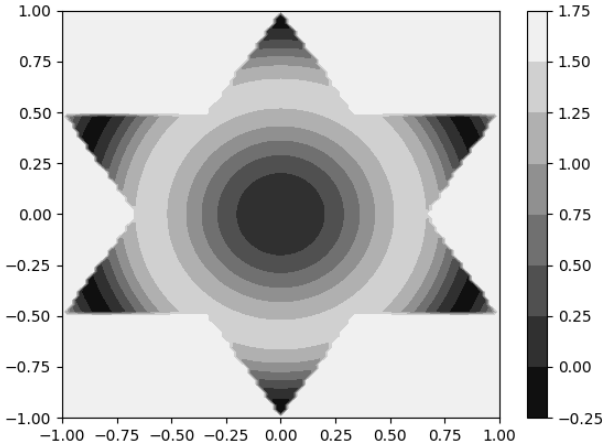


Calandra, H. and Gratton, S. and Riccietti, E. and Vasseur, X.. *On a multilevel Levenberg-Marquardt method for the training of artificial neural networks and its application to the solution of partial differential equations*. **Optim. Methods Softw. (2020)**.

Numerical results on difficult domains ($n = 4096$)

Left: $-\Delta u + \nu^2 u = g_1$, $u(x,y) = \sin(\nu(x+y))$ $\nu = 3$
Right: $-\Delta u + \nu u^2 = g_1$, $u(x,y) = (x^2+y^2) + \sin(\nu(x^2+y^2))$, $\nu = \frac{1}{2}$





	iter	RMSE	savings			iter	RMSE	savings		
			min	avg	max			min	avg	max
1 level	395	10^{-4}				1408	10^{-3}			
2 levels	110	10^{-4}	1.3	5.6	10.0	1301	10^{-3}	1.2	1.9	2.4

Conclusion

Conclusion

- Theoretical contribution: We have presented a class of multilevel high-order methods for optimization and proved their global and local convergence and complexity.
- Practical contribution: We have got further insight on the methods proposing a AMG strategy to build coarse representations of the problem to use some methods in the family for the training of artificial neural networks.

Perspectives

- Hessian-free variant. Make it a competitive training method: the method needs to compute and store the Hessian matrix (for step computation and to build transfer operators): too expensive for large-scale problems.

Thank you for your attention!

Slides and papers available here

bit.ly/elisaIRIT

Calandra, H. and Gratton, S. and Riccietti, E. and Vasseur, X.. *On high-order multilevel optimization strategies*. **Submitted to SIAM J. Optim. (2019)**.

Calandra, H. and Gratton, S. and Riccietti, E. and Vasseur, X.. *On a multilevel Levenberg-Marquardt method for the training of artificial neural networks and its application to the solution of partial differential equations*. **Optim. Methods Softw. (2020)**.

Calandra, H. and Gratton, S. and Riccietti, E. and Vasseur, X.. *On the iterative solution of the extended normal equations*. **Submitted to SIAM J. Matrix Anal. Appl. (2019)**.

Backup slides

Tensor of order 3

Definition

Let $T \in \mathbb{R}^{n^3}$, and $u, v, w \in \mathbb{R}^n$. Then $T(u, v, w) \in \mathbb{R}$, $T(v, w) \in \mathbb{R}^n$

$$T(u, v, w) = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} T(i, j, k) u(i) v(j) w(k),$$

$$T(v, w)(i) = \sum_{j=1}^{n} \sum_{k=1}^{n} T(i, j, k) v(j) w(k), \quad i = 1, \ldots, n.$$

Tensor of order i

Definition

Let $i \in \mathbb{N}$ and $T \in \mathbb{R}^{n^i}$, and $u \in \mathbb{R}^n$. Then $T(\underbrace{u, \dots, u}_{i\,times}) \in \mathbb{R}$,

$T(\underbrace{u, \dots, u}_{i-1\,\text{times}}) \in \mathbb{R}^n$ and

$$T(\underbrace{u, \dots, u}_{i\,\text{times}}) = \sum_{j_1=1}^{n} \cdots \sum_{j_i=1}^{n} T(j_1, \dots, j_i) u(j_1) \dots u(j_i),$$

$$T(\underbrace{u, \dots, u}_{i-1\,\text{times}})(j_1) = \sum_{j_2=1}^{n} \cdots \sum_{j_i=1}^{n} T(j_1, \dots, j_i) u(j_2), \dots u(j_i), \quad j_1 = 1, \dots, n.$$

Stopping criterion for inner iterations

$$m_{q,k}(x_k, s_k; \lambda_k) < m_{q,k}(x_k, 0; \lambda_k), \quad \|\nabla_s m_{q,k}(x_k, s_k; \lambda_k)\| \leq \theta \|s_k\|^q,$$

Hessian approximation in least squares

The Hessian of f is given by

$$\nabla^2 f(x) = J(x)^T J(x) + S(x) = J(x)^T J(x) + \sum_{i=1}^{m} F_i(x) \nabla^2 F_i(x).$$

Notice that term $S(x)$ contains the second derivatives $\nabla^2 F_i$ of R. Its norm depends both on the nonlinear residual $F(x)$ and on the magnitude of such derivatives.

Prolongation operator for PDE problem without network

P_l: standard interpolation operator for $d = 1$ and on the nine-point interpolation scheme defined by the stencil $\begin{pmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{pmatrix}$ for $d = 2$,

$R_l = \frac{1}{2^d} P_l^T$ full weighting operators

Prolongation operator for PDE problem with network

$$x_i^h = (Px^H)_i = \begin{cases} x_i^H & \text{if } \ i \in C, \\ \sum_{k \in P_i} \delta_{i,k} x_k^H & \text{if } \ i \in F, \end{cases}$$

with

$$\delta_{i,k} = \begin{cases} -\alpha_i a_{i,k}/a_{i,i} & \text{if } \ k \in P_i^-, \\ -\beta_i a_{i,k}/a_{i,i} & \text{if } \ k \in P_i^+, \end{cases} \qquad \alpha_i = \frac{\sum_{j \in N_i} a_{i,j}^-}{\sum_{k \in P_i} a_{i,k}^-}, \beta_i = \frac{\sum_{j \in N_i} a_{i,j}^+}{\sum_{k \in P_i} a_{i,k}^+},$$

where $a_{i,j}^+ = \max\{a_{i,j}, 0\}$, $a_{i,j}^- = \min\{a_{i,j}, 0\}$, N_i is the set of variables connected to i (i.e. all j such that $a_{i,j} \neq 0$), P_i the set of coarse variables strongly connected to i, which is partitioned in P_i^- (negative couplings) and P_i^+ (positive couplings).

The multilevel LM method fits in the family

- If $q = 1$, the regularized model is defined as

$$m_k(x_k, s) = f(x_k) + \nabla f(x_k)^T s + \frac{\lambda_k}{2}\|s\|^2,$$

- Least-squares problems:

$$\begin{aligned} m_k^{LM}(x_k, s) &= \frac{1}{2}\|F(x_k) + J(x_k)s\|^2 + \frac{1}{2}\lambda_k\|s\|^2 \\ &= \frac{1}{2}\|F(x_k)\|^2 + s^T J(x_k)^T F(x_k) + \frac{1}{2}s^T B_k s + \frac{1}{2}\lambda_k\|s\|^2 \\ &= f(x_k) + \nabla f(x_k)^T s + \frac{1}{2}s^T B_k s + \frac{1}{2}\lambda_k\|s\|^2 \end{aligned}$$

- Let $M = \frac{B_k}{\lambda_k} + I \Rightarrow \frac{\lambda_k}{2}\|s\|_M^2 = \frac{1}{2}s^T B_k s + \frac{\lambda_k}{2}\|s\|^2$*
- $m_k^{LM}(x_k, s) = f(x_k) + \nabla f(x_k)^T s + \frac{\lambda_k}{2}\|s\|_M^2$.

*For a positive definite matrix $M \in \mathbb{R}^{n\times n}$ and $x \in \mathbb{R}^n$, $\|x\|_M = x^T M x$.

Model minimization at level ℓ

- Step computation in classical methods:

$$\min_s m_k(x_k, s)$$

which is equivalent to the solution of the normal equations:

$$(J(x_k)^T J(x_k) + \lambda_k I)s = -J(x_k)^T F(x_k) \quad \Rightarrow \; CGLS$$

Model minimization at level ℓ

- Step computation in classical methods:

$$\min_s m_k(x_k, s)$$

which is equivalent to the solution of the normal equations:

$$(J(x_k)^T J(x_k) + \lambda_k I)s = -J(x_k)^T F(x_k) \quad \Rightarrow \; CGLS$$

- Multilevel method:

$$\min_s m_k^{\ell-1}(x_k, s) + c^T s,$$

$c^T s$ ensures coherence among levels.
This model leads to extended normal equations:

$$(J(x_k)^T J(x_k) + \lambda_k I)s = -J(x_k)^T F(x_k) + c \quad \Rightarrow \; ?$$

Stable solution of extended normal equations

- CGLS cannot be used, due to the presence of term $+c$
- CG is not stable on normal equations $A^T Ax = A^T b$

CG

$$r_k = A^T(b - Ax_k).$$

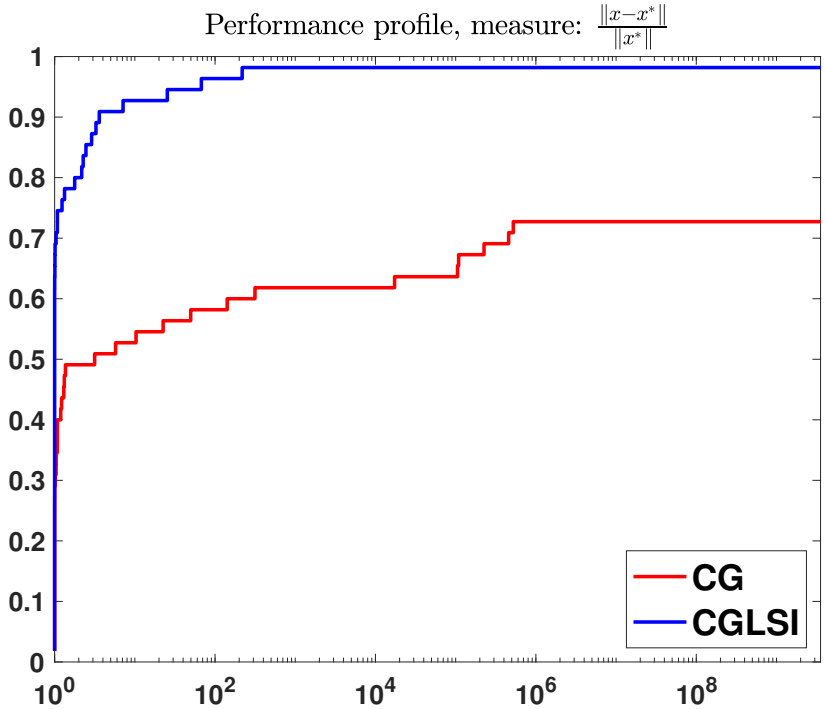
CGLS

$$\begin{aligned} r_k =& b - Ax_k, \\ s_k =& A^T r_k. \end{aligned}$$

Stable solution of extended normal equations

$\Rightarrow$ We propose (CGLSI), a modification of CGLS, suitable for our problem and more stable than CG

CGLS	CGLSI
$r_k = b - Ax_k,$	$r_k = b - Ax_k,$
$s_k = A^T r_k.$	$s_k = A^T r_k + c.$



Calandra, H. and Gratton, S. and Riccietti, E. and Vasseur, X.. *On the iterative solution of the extended normal equations*. **Submitted to SIAM J. Matrix Anal. Appl. (2019)**.