

Multilevel optimization methods

Elisa Riccietti

LIP-ENS Lyon

Joint work with:

- ▶ H. Calandra (TOTAL), S. Gratton - V. Mercier (IRIT), P. Toint (Univ. Namur), X. Vasseur (ISAE-SUPAERO)

Context: large scale optimization problems

$$\min_x f(x) \quad \rightarrow \quad \min_{x \in \mathbb{R}^n} f(x) = \sum_{i=1}^m f_i(x)$$

Large scale problems

- ▶ f has a large number of unknowns: **large n** (ex: deep learning)
- ▶ f is the sum of a large number of terms: **large m** (ex: classification of large datasets)

Context: large scale optimization problems

$$\min_x f(x) \quad \rightarrow \quad \min_{x \in \mathbb{R}^n} f(x) = \sum_{i=1}^m f_i(x)$$

Large scale problems

- ▶ f has a large number of unknowns: **large n** (ex: deep learning)
→ **Multilevel methods**
- ▶ f is the sum of a large number of terms: **large m** (ex: classification of large datasets)

Outline

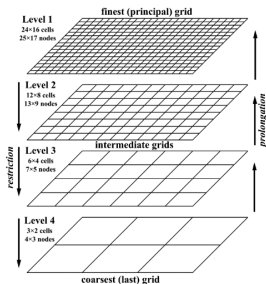
- ▶ **Part I:** Brief recap on **multigrid methods** for the solution of PDEs
- ▶ **Part II:** Their transposition to a nonlinear context: **high-order multilevel optimization** methods
- ▶ **Part III:** **Artificial neural networks** for the solution of PDEs
- ▶ **Part IV:** Multilevel methods for their **training**

Part I

Multigrid methods

Multigrid methods for PDEs

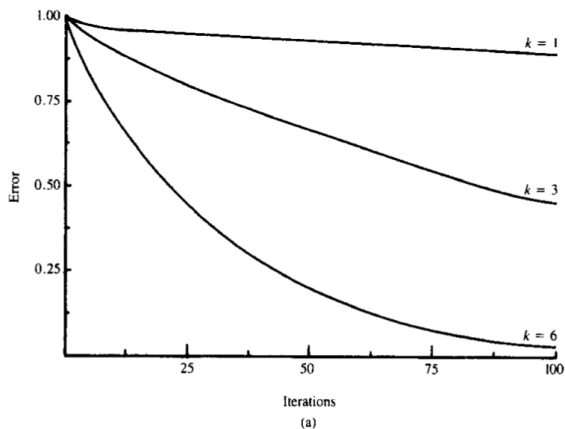
State-of-the-art methods for PDEs: exploit representation of the problem at different scales



- ▶ Fine scales: eliminate **high frequency** components of the error
- ▶ Coarse scales: eliminate **low frequency** components of the error

The intuition behind multigrid methods

- ▶ The **smoothing property**: hard for fixed point iterative methods to reduce the low frequency components of the error



Two-level multigrid methods

Consider a linear PDE:

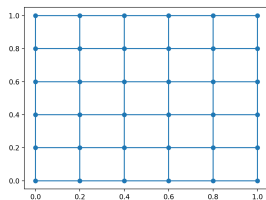
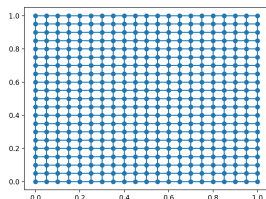
$$Au = f.$$

Consider two discretizations of the same system:

- ▶ Fine grid: $A_h u_h = f_h$
- ▶ Coarse grid: $A_H u_H = f_H$

Idea: write the solution u as the sum of a fine and a coarse term:

$$u \sim \underbrace{v_h}_{\in \mathbb{R}^h} + P(\underbrace{e_H}_{\in \mathbb{R}^H}), \quad H < h.$$



Two-level multigrid methods

Build operators to transfer the information between the two levels

R :

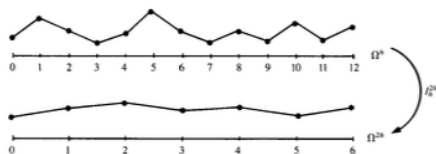


Figure 3.4: Restriction by full weighting of a fine-grid vector to the coarse grid.

P :

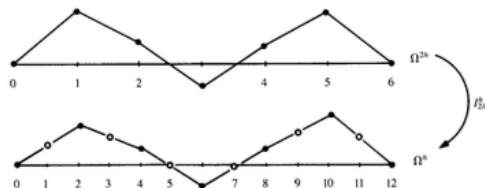


Figure 3.2: Interpolation of a vector on coarse grid Ω^{2h} to fine grid Ω^h .

Two-level multigrid methods

Update the two components in an alternate fashion:

$$u \sim v + e$$

$$r = f - Av$$

$$Ae = r \text{ residual equation}$$

- ▶ *Fine level*: get v_h by iterating on

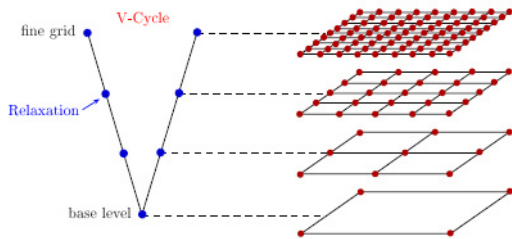
$$A_h u = f_h$$

- ▶ Compute $r_h = f - Av_h$ and project $r_H = Rr_h$
- ▶ *Coarse level*: compute correction by the residual equation:

$$A_H e_H = r_H$$

- ▶ Correct: $v_h \leftarrow v_h + P(e_H)$

General multigrid methods



Part II

High-order multilevel optimization methods

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:

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

$r(\lambda_k)$ regularization term. Set $x_{k+1} = x_k + s_k$

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):

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



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): 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 $\|\nabla f(x_k)\| \leq \epsilon$: $k = O(\epsilon^?)$

Family of higher-order methods generalizing ARC

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

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

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

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

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.



S. Gratton, A. Sartenaer and Ph. L. Toint, 2008: second order multilevel trust-region methods

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



On high-order multilevel optimization strategies, H. Calandra, S. Gratton, E. Riccietti, X. Vasseur, 2020

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 $f^{\ell-1}$

$$\begin{array}{ccc} x_k^\ell & & x_{k+1}^\ell = x_k^\ell + s_k^\ell \\ \downarrow R^\ell & & \uparrow s_k^\ell = P^\ell(x_{*,k}^{\ell-1} - x_{0,k}^{\ell-1}) \\ x_{0,k}^{\ell-1} := R^\ell x_k^\ell & \xrightarrow{\min_x \mu^{\ell-1}(x)} & x_{*,k}^{\ell-1} \end{array}$$

The procedure is recursive: more levels can be used

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:

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

This ensures that

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

→ **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

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

→ 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

$$\mu_{q,k}^{\ell-1}(x_{0,k}^{\ell-1}, s^{\ell-1}) = f^{\ell-1}(x_{0,k}^{\ell-1} + s^{\ell-1}) + \sum_{i=1}^q \frac{1}{i!} [\mathcal{R}(\nabla^i f^\ell(x_k)) - \nabla^i f^{\ell-1}(x_{0,k}^{\ell-1})] \underbrace{(s_1^{\ell-1}, \dots, s_i^{\ell-1})}_{i \text{ times}},$$

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

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



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

Theoretical results: global convergence

Theorem

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

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

-  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:

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

-  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 $q = 2, 3$







		$n = 1024$		$n = 4096$	
$d = 2, q = 2$		AR2	MAR2	AR2	MAR2
\bar{u}_1	it_T/it_f save	11/11	7/2 2.2	23/23	15/4 4.1
\bar{u}_2	it_T/it_f save	27/27	13/4 3.9	56/56	22/6 6.1

		$n = 256$		$n = 512$	
$d = 1, q = 3$		AR3	MAR3	AR3	MAR3
\bar{u}_1	it_T/it_f save	7/7	9/2 2.5	18/18	15/2 4.3
\bar{u}_2	it_T/it_f save	23/23	14/1 4.1	34/34	20/5 4.4

Part III

Artificial neural networks for PDEs

Solution of PDEs with ANNs, an active field of research

-  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).

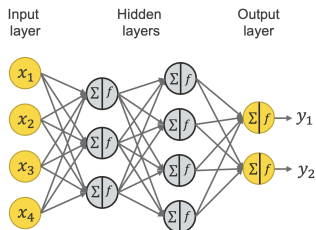
Why this approach ?

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)

General NN strategy for learning problems

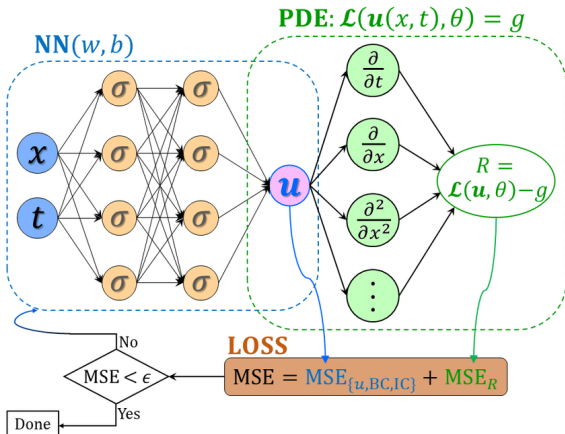


- ▶ Database composed of inputs $x = (x_1, \dots, x_n)$ and outputs $y = (y_1, \dots, y_m)$.
- ▶ Loss function noted $L(y, x, \theta) = \sum_{i=1}^m (NN(x_i) - y_i)^2$
- ▶ The associated minimization problem : $\min_{\theta \in \Theta} L(y, x, \theta)$
- ▶ Optimize by SGD

How to integrate further physical knowledge in the model?

Physics Informed Neural Networks (PINNs)

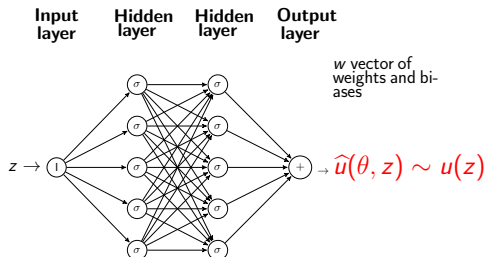
General principle



How does a PINN work?

Physics informed neural networks

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



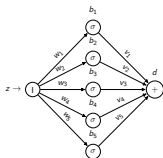
Training problem: find the network weights θ by minimizing

$$\min_{\theta} \underbrace{\| D(z, \hat{u}(\theta, z_t)) - g(z_t) \|^2}_{L_R: \text{Equation residual}} + \lambda_p \underbrace{((\hat{u}(\theta, a) - A)^2 + (\hat{u}(\theta, b) - B)^2)}_{L_B: \text{Boundary conditions}}$$

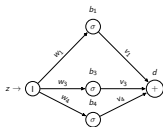
Part IV

Multilevel training methods

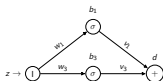
How to exploit multilevel method for training of ANNs?



$R_1 \Downarrow P_1 \Uparrow$



$R_2 \Downarrow P_2 \Uparrow$



Large-scale problem

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

Two-level multigrid methods

Update the two components in an alternate fashion:

$$u \sim \underbrace{v_h}_{\in \mathbb{R}^h} + P(\underbrace{e_H}_{\in \mathbb{R}^H}), \quad H < h.$$

- ▶ *Fine level*: get v_h by iterating on

$$A_h(u) = f_h$$

- ▶ *Coarse level*: compute correction by the residual equation:

$$A_H(Rv_h + e_H) = A_H(Rv_h) + R(f_h - A_h(v_h))$$

- ▶ Correct: $v_h \leftarrow v_h + P(e_H)$

Problem definition

$$\begin{aligned} D(z, u(z)) &= g(z), \quad z \in \Omega, \\ u_{sol}(z) &\sim u_h(\theta_h, z) + u_H(\theta_H, z) \end{aligned}$$

$$L_h(\theta_h) = L_{R,h}(\theta_h) + L_{B,h}(\theta_h)$$

$$L_{R,h}(\theta_h) = \|D(\hat{u}_h(\theta_h) + u_H) - g\|^2$$

$$L_{B,h}(\theta_h) = \|\hat{u}_h(\theta_h) + u_H - u\|^2$$

Computed on z_h the fine sampling

$$L_H(\theta_H) = L_{R,H}(\theta_H) + L_{B,H}(\theta_H)$$

$$L_{R,H}(\theta_H) = \|D(\hat{u}_H(\theta_H) + u_h) - g\|^2$$

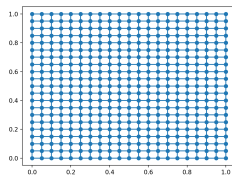
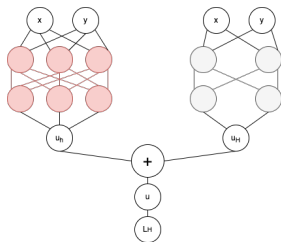
$$L_{B,H}(\theta_H) = \|\hat{u}_H(\theta_H) + u_h - u\|^2$$

Computed on z_H the coarse sampling

Multilevel PINNs

Algorithm 1 2-levels training of PINNs

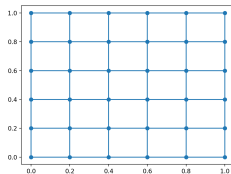
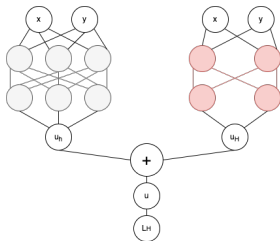
- 1: Freeze coarse-network parameters, unfreeze fine-network parameters
- 2: **for** $i=1,2,\dots$ **do**
- 3: Perform ν_1 epochs for the minimization of the fine problem
- 4: Freeze fine-network parameters, unfreeze coarse-network parameters
- 5: Perform ν_2 epochs for the minimization of the coarse problem
- 6: **end for**
- 7: Return : $u_H + u_h$



Multilevel PINNs

Algorithm 2 2-levels training of PINNs

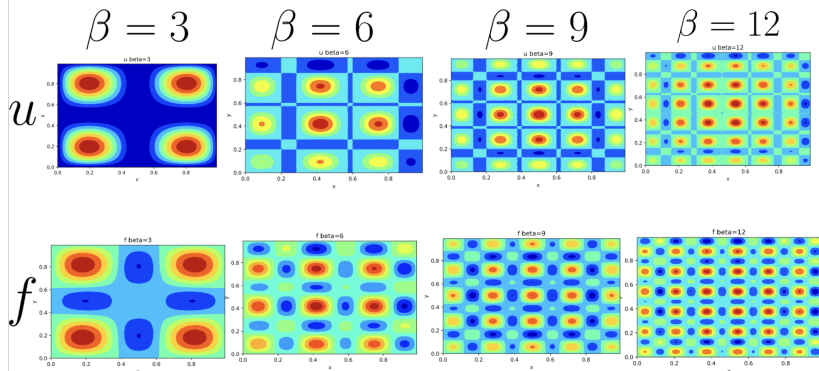
- 1: Freeze coarse-network parameters, unfreeze fine-network parameters
- 2: **for** $i=1,2,\dots$ **do**
- 3: Perform ν_1 epochs for the minimization of the fine problem
- 4: Freeze fine-network parameters, unfreeze coarse-network parameters
- 5: Perform ν_2 epochs for the minimization of the coarse problem
- 6: **end for**
- 7: Return : $u_H + u_h$



Experimental results

A simple Poisson problem :

- ▶ $\Omega = [0, 1] \times [0, 1]$
- ▶ $\Delta u = f \quad \forall x \in \Omega$
- ▶ $u = 0 \quad \forall x \in \partial\Omega$
- ▶ $u(x, y) = (\sin(\pi x) + \sin(\beta\pi x)) * (\sin(\pi y) + \sin(\beta\pi y))$



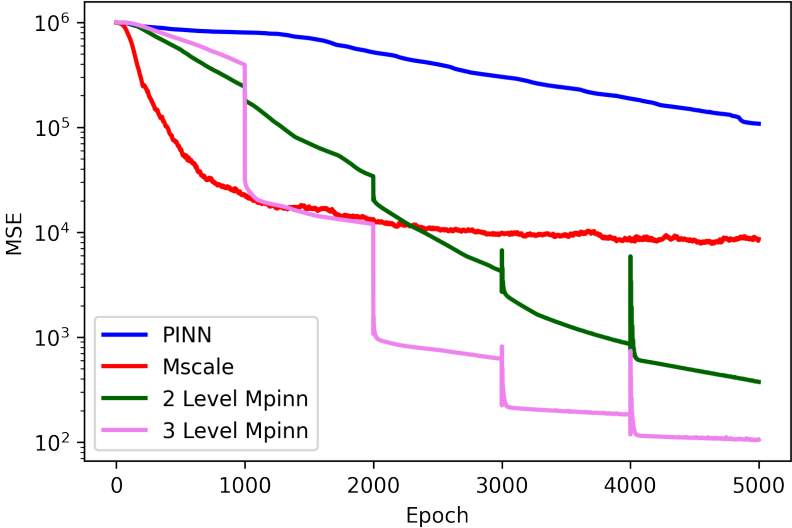
Experimental results

Experimental settings :

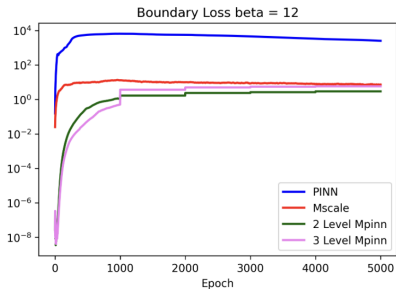
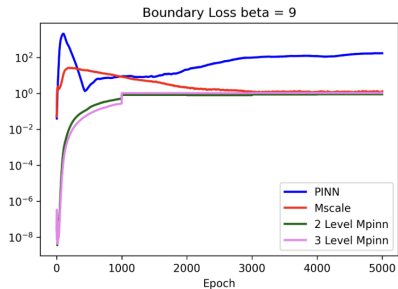
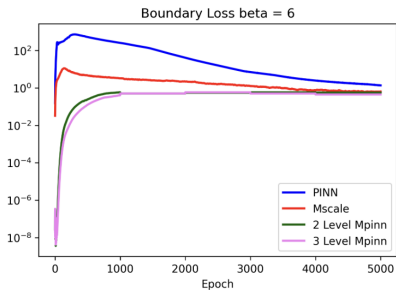
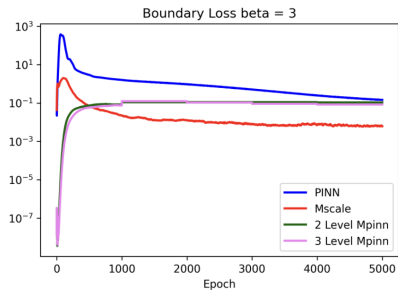
- ▶ The PINNs have two hidden layers of 300 neurons each.
- ▶ The Mscale have four subnetworks of two hidden layers of 150 neurons each
- ▶ The two-levels MPINN is composed of two networks of two hidden layers of 210 neurons each and trained in a **V-cycle** ($\nu_1 = \nu_2 = 1000$)
- ▶ The three level MPINN is composed of three networks of two hidden layers of 150 neurons each and trained in a **V-cycle** ($\nu_1 = \nu_2 = 1000$)
- ▶ The input of all network is a regular grid sample of 80×80 points
- ▶ In all cases, we plot the median of ten random runs.

Experimental results

Loss beta = 9

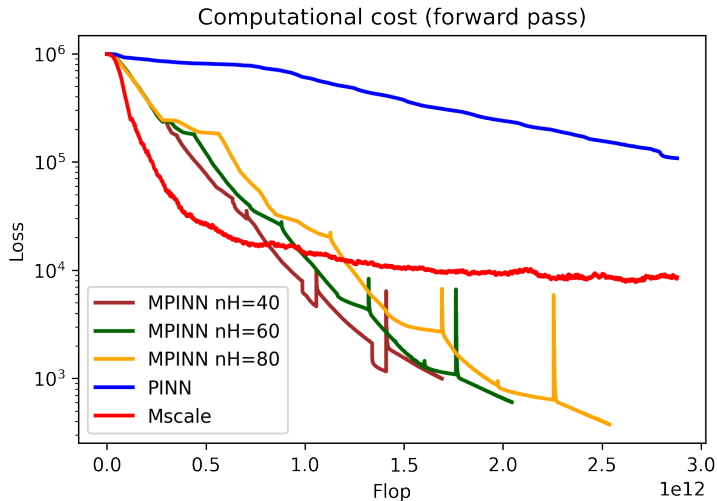


Convergence of boundary conditions



Computational cost for two levels...

... as a function of coarse grid size (nH)



Conclusions

- ▶ We have presented a general family of **high-order multilevel optimization methods**.
- ▶ We have presented a new **multigrid-inspired training** framework using recent advances in NN to efficiently solve PINN-type problems.

Perspectives

For the multigrid training method:




- ▶ Perform further **extensive testing**, including more complex problems
- ▶ Pursue the **sensitivity analysis** for the relative sizes of the grids
- ▶ Investigate **theoretical aspects**:
 - ▶ convergence of the iterates from an optimization point of view
 - ▶ convergence to the solution in functional space

A new perspective: **proximal multilevel methods** for image denoising (ongoing work)

Thank you for your attention!

Slides and papers available here

bit.ly/elisaIRIT

-  *On high-order multilevel optimization strategies*, H. Calandra, S. Gratton, E. Riccietti, X. Vasseur, 2020
-  *On a multilevel Levenberg-Marquardt method for the training of artificial neural networks and its application to the solution of partial differential equations*, H. Calandra, S. Gratton, E. Riccietti, X. Vasseur, 2020
-  *Multilevel physics informed neural networks (MPINNs)*, E. Riccietti, V. Mercier, S. Gratton, P. Boudier, 2022