

# Second order methods for the solution of large-scale nonlinear noisy problems

Elisa Riccietti

## Short bio

- **PhD thesis (Oct. 2014 – Oct. 2017):**
  - at Università degli Studi di Firenze (**Florence**, Italy)
  - supervised by **Stefania Bellavia**
- **Postdoctorate (Nov. 2017 – ongoing)**
  - at Institut de Recherche en Informatique de Toulouse (**IRIT**)
  - supervised by **Serge Gratton**

## Continuous optimization problems

$$\min_x f(x)$$

### Large-scale problems

$$f : \mathbb{R}^n \rightarrow \mathbb{R}$$

- has **many variables** (large  $n$ , ex: deep learning)
- is the result of **many computations**,  $f(x) = \sum_{i=1}^m f_i(x)$  (large  $m$ , ex: classification of large datasets)

### Difficult problems

- **Nonconvex** and highly **nonlinear**
- **Ill-posed** and **ill-conditioned**
- Several local stationary points (local minima and **saddle points**)

The solution is approximated by a sequence  $x_k$  converging to a stationary point  $x^*$  such that  $\nabla f(x^*) = 0$ .

## First order

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k),$$

where  $\alpha_k$  is the step length (*learning rate*).

- 😊 Low computational cost and memory consumption
- ☹ Better suited for convex problems, dependent on the choice of  $\alpha_k$ , slow convergence

## Second order

$$x_{k+1} = x_k - H(x_k)^{-1} \nabla f(x_k)$$

where  $H$  is the Hessian matrix.

- ☹ Need for linear systems solution, high computational cost and memory consumption
- 😊 Efficient on nonconvex problems, robust, fast convergence

The solution is approximated by a sequence  $x_k$  converging to a stationary point  $x^*$  such that  $\nabla f(x^*) = 0$ .

## First order

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k),$$

where  $\alpha_k$  is the step length  
(*learning rate*).

- 😊 Low computational cost and memory consumption
- ☹ Better suited for convex problems, dependent on the choice of  $\alpha_k$ , slow convergence

## Second order

$$x_{k+1} = x_k - H(x_k)^{-1} \nabla f(x_k)$$

where  $H$  is the Hessian matrix.

- ☹ Need for linear systems solution, high computational cost and memory consumption
- 😊 Efficient on nonconvex problems, robust, fast convergence

## Nonlinear least-squares problems

Given  $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$ , nonlinear, continuously differentiable solve

$$\min_{x \in \mathbb{R}^n} f(x) = \frac{1}{2} \|F(x)\|^2$$

Let  $x^*$  be a solution of the problem.

## Nonlinear least-squares problems

Given  $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$ , nonlinear, continuously differentiable solve

$$\min_{x \in \mathbb{R}^n} f(x) = \frac{1}{2} \|F(x)\|^2$$

Let  $x^*$  be a solution of the problem.

## Noisy least-squares problems

In many cases  $f$  and its derivatives are not available exactly. We seek an approximation to  $x^*$  considering a sequence of **noisy functions**:

$$f_{\delta_k} \sim f$$

# Outline: two sources of noise

- **Part I: Ill-posed problems**

**Aim:** stable methods for problems with noisy data



Bellavia, S. and Morini, B. and Riccietti, E.. *On an adaptive regularization for ill-posed nonlinear systems and its trust-region implementation*. Comput. Optim. Appl. (2016).



Bellavia, S. and Riccietti, E.. *On an elliptical trust-region procedure for ill-posed nonlinear least squares problems*. J. Optim. Theory Appl. (2018).



Bellavia, S. and Donatelli, M. and Riccietti, E.. *An inexact non stationary Tikhonov procedure for large-scale nonlinear ill-posed problems*. Submitted to: Inverse Probl. (2020).

- **Part II: Large-scale problems**

**Aim:** fast methods exploiting cheaper approximations



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).



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*. Submitted to Optim. Methods Softw. (2019).



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).

### Levenberg–Marquardt and Trust-region methods

- LM:  $\min_p \frac{1}{2} \|F(x_k) + J(x_k)p\|^2 + \frac{\lambda_k}{2} \|p\|^2$
- TR:  $\min_p \frac{1}{2} \|F(x_k) + J(x_k)p\|^2$ , s.t.  $\|p\| \leq \Delta_k$

Both methods need the solution of a **linear system**:

$$(B_k + \lambda_k I)p_k = -g_k, \quad B_k = J(x_k)^T J(x_k), \quad g_k = J(x_k)^T F(x_k)$$

(For TR  $\lambda_k$  is such that  $\lambda_k(\|p_k\| - \Delta_k) = 0$ )



## Global convergence

For any initial guess, the sequence of iterates **converges to a first-order stationary point**:

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

## Worst case complexity

Given  $\epsilon > 0$ , compute the number of iterations required to achieve an iterate  $x_k$  such that

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

## Local convergence and rate of convergence

The sequence  $\{x_k\}$  converges to  $x^*$  if the initial approximation is close enough to the solution and it exist  $c > 0$ ,  $\beta \geq 1$  such that:

$$\lim_{k \rightarrow \infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|^\beta} = c$$

Rates of convergence:

	sublinear	linear	superlinear	quadratic
$\beta$	1	1	1	2
$c$	1	$]0, 1[$	0	$> 0$

Ill-posed least squares problems

# Ill-posed problems with noisy data

- Original problem:

$$\min_{x \in \mathbb{R}^n} f(x) = \|F(x) - y\|^2, \quad F : \mathbb{R}^n \rightarrow \mathbb{R}^m, \quad m \geq n \quad (1)$$

## Ill-posed

- solution is **not unique**,
- solution does **not depend continuously** on the data

- Noise on the data:

$$\min_{x \in \mathbb{R}^n} f^\delta(x) = \|F(x) - y^\delta\|^2, \quad \|y - y^\delta\| \leq \delta. \quad (2)$$

The solutions of (2) may not be meaningful approximations of the solutions of (1): need for **regularization methods**

# Drawbacks of state-of-the-art regularization methods

- **Tikhonov method:**  $\min_p \frac{1}{2} \|F(x_k) - y^\delta + J(x_k)p\|^2 + \frac{\lambda}{2} \|p\|^2$   
Choice of  $\lambda$  is often based on **a-priori information** on the solutions (such as an estimate of the error)
- **Levenberg–Marquardt method** [Hanke 1997,2010]:  
 $\min_p \frac{1}{2} \|F(x_k) - y^\delta + J(x_k)p\|^2 + \frac{\lambda_k}{2} \|p\|^2$  Automatic  $\lambda_k$  but convergence guaranteed for a **starting guess close to a solution** is provided  
 $\Rightarrow$  both methods need a-priori information on the solution
- **All methods** but [Binder et al. 1994 (Tikhonov method)]: need hypothesis of **zero residual**: it exists  $x^*$  such that  $r = F(x^*) - y = 0$ . **This is not the case in many applications**  
The problem can be reduced to zero residual:  $y^\delta \leftarrow y^\delta + r$ , but only if  $\|r\|$  can be estimated: this is difficult to do

Ill-posed least squares problems

Zero residual problems

Nonzero residual problems

Large-scale nonzero residual problems

Large-scale problems

Subsampled methods for large  $m$

Multilevel methods for large  $n$

Conclusion & research project

# A globally convergent, regularizing trust-region method

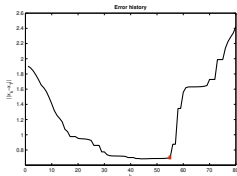
**Our idea:** combine trust-region scheme with regularization

Two **key ingredients**:

**1) New trust-region radius update:**  $\Delta_k \leq \frac{1-q}{\|B_k\|} \|\nabla f^\delta(x_k)\|$

- Exploit properties of trust-region schemes to enforce global convergence
- Exploit adaptive choice of  $\lambda_k$  from regularizing Levenberg–Marquardt to enforce regularizing properties

**2) Suitable stopping criterion. Discrepancy principle:** stop at first  $k^*(\delta)$  such that:  $\|F(x_{k^*(\delta)}) - y^\delta\| \leq \tau\delta$ ,  $\tau > 1$



**Semi convergence**

Plot of the error  $\|x_k - x^*\|$   
versus iteration number.

# Theoretical results

Under suitable assumptions on the nonlinearity of the function

$$\delta = 0$$

- Global convergence
- Complexity  $O(\epsilon^{-2})$
- Local convergence to  $x^*$  such that  $F(x^*) = y$  at linear rate

$$\delta > 0$$

- Finite termination  $k^*(\delta) = O(\delta^{-2})$
- Convergence to  $x^*$  of  $\{x_{k^*(\delta)}\}$  for  $\delta \rightarrow 0$



Bellavia, S. and Morini, B. and Riccietti, E.. *On an adaptive regularization for ill-posed nonlinear systems and its trust-region implementation*. Comput. Optim. Appl. (2016).

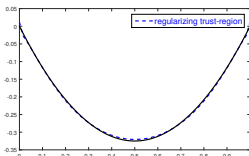
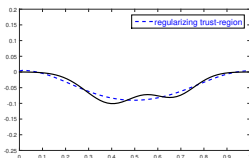


# Comparison between regularizing TR e LM [Hanke]

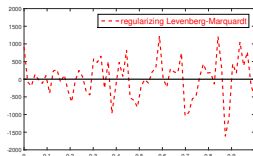
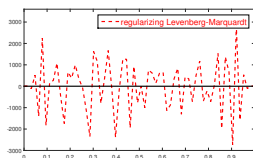
Test problems: first-kind Fredholm integral equation

$$\int_0^1 k(t, s, x(s)) ds = y(t), \quad t \in [0, 1],$$

**Regularizing TR**  
(our method)



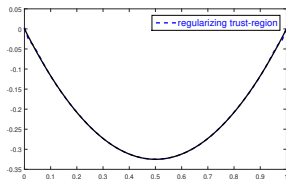
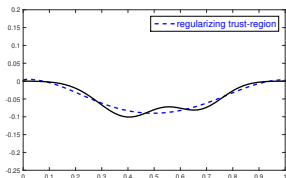
**Regularizing LM**  
(state-of-the-art)



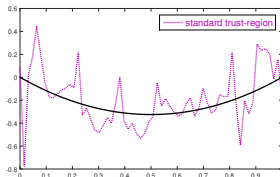
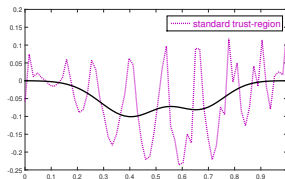
Results for an initial guess  $x_0$  not close from  $x^* \Rightarrow$  global convergence of TR allows for stable solution even in this case

# Comparison between regularizing and standard TR

## Regularizing TR



## Standard TR



⇒ Improved robustness comes from **combination of TR scheme and regularization**, not just TR

## Ill-posed least squares problems

Zero residual problems

**Nonzero residual problems**

Large-scale nonzero residual problems

## Large-scale problems

Subsampled methods for large  $m$

Multilevel methods for large  $n$

Conclusion & research project

The proposed method cannot handle problems with **nonzero residual**

- Tikhonov regularization with a general penalty term ( $M$  spd)

$$\min_p \frac{1}{2} \|F(x_k) - y^\delta + J(x_k)p\|^2 + \frac{\lambda}{2} \|Mp\|^2$$

- **Our proposal: elliptical trust-region approach**

$$\min_p \frac{1}{2} \|F(x_k) - y^\delta + J(x_k)p\|^2, \quad \text{s.t. } \|M_k p\| \leq \Delta_k.$$

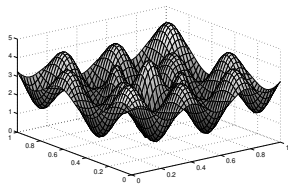
- To ensure decrease of the error when the residual is nonzero we choose:  $M_k = B_k^{-1/2}$ ,  $B_k = J(x_k)^T J(x_k)$ .

⇒ We have generalized trust-region update, stopping criterion and theoretical results to nonzero residual problems

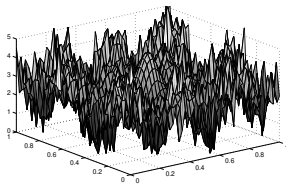


Bellavia, S. and Riccietti, E.. *On an elliptical trust-region procedure for ill-posed nonlinear least squares problems*. J. Optim. Theory Appl. (2018).

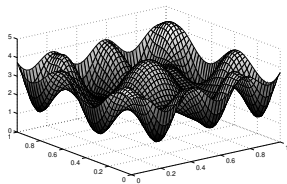
# Geophysics and Biomedical problems



**True solution**



**Standard TR**



**Our method**

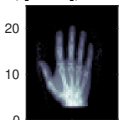
**Geo:**  $\delta = 10^{-2}$ ,

$\|r\| = 10^{-2}$

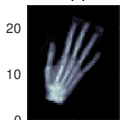
**Biom:**  $\delta = 10^{-1}$ ,

$\|r\| = 10^2$

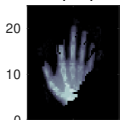
**R, [ 64 64],  $\alpha=2000$**



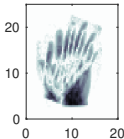
**T(0)**



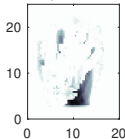
**T(230)**



**|T(xc)-R(xc)|**



**|T(yc)-R(xc)|**



## Ill-posed least squares problems

- Zero residual problems

- Nonzero residual problems

- Large-scale nonzero residual problems

## Large-scale problems

- Subsampled methods for large  $m$

- Multilevel methods for large  $n$

Conclusion & research project

- The proposed methods require several times the solution of

$$(B_k + \lambda I)p(\lambda) = -g_k, \quad B_k \in \mathbb{R}^{n \times n} \quad \text{Large } n \Rightarrow \text{Expensive!}$$

$$(B_k^2 + \lambda I)z(\lambda) = -B_k^{1/2} g_k, \quad p(\lambda) = B_k^{1/2} z(\lambda) \Rightarrow \text{Even more expensive!}$$

Our solution: **Lanczos bidiagonalization**

$$J(x_k) = P_\ell T_\ell Q_\ell^T, \quad B_k = Q_\ell T_\ell^T T_\ell Q_\ell^T \quad \text{with } T_\ell \text{ bidiagonal matrix}$$

- Exact solution of the system is not affordable  $\rightarrow$  solution is sought in  $\mathcal{K}_\ell(B_k, g_k)$  generated Krylov space
- Exact computation of the RHS is not affordable  $\rightarrow$

$$B_k^{1/2} g_k \sim Q_\ell (T_\ell^T T_\ell)^{1/2} Q_\ell^T$$



Bellavia, S. and Donatelli, M. and Riccietti, E.. *An inexact non stationary Tikhonov procedure for large-scale nonlinear ill-posed problems*. Submitted to: Inverse Probl. (2020).

# Theoretical and numerical results

- Have to take into account **two sources of inexactness** in the analysis
- **Structured perturbations**: structure induced by Lanczos process allows us to prove **decrease of the error**
- The inexact strategy provides considerable **computational savings** without affecting the solution's quality

Comparison of exact and inexact methods ( $\ell = 10$ )

	Geophysics ( $n = 4096$ )		Image registration ( $n = 8320$ )	
	exact	inexact	exact	inexact
iterations	67	67	36	137
time (s)	9519	2612	10730	55

$\div 3.6$        $\div 195$



Large-scale problems

## Part II: Large-scale problems

We consider large-scale problems for which the objective function is expensive to evaluate:

$$\min_x f(x) = \frac{1}{2} \|F(x)\|^2 \quad F : \mathbb{R}^n \rightarrow \mathbb{R}^m$$

We distinguish two cases:

- $F$  has many components: **large  $m$**
- $F$  depends on a large number of variables: **large  $n$**

In many applications  $f$  can be approximated by **cheaper** approximations  $\Rightarrow$  we want to exploit them to **reduce the computational cost** of the solution

We consider two classes of methods:

- Large  $m \Rightarrow$  **subsampling methods**
- Large  $n \Rightarrow$  **multilevel methods**

Ill-posed least squares problems

Zero residual problems

Nonzero residual problems

Large-scale nonzero residual problems

Large-scale problems

Subsampled methods for large  $m$

Multilevel methods for large  $n$

Conclusion & research project

## Subsampling

Large set of data at disposal:  $\{1, \dots, m\}$ .

**Subsampling:**  $X_k \subseteq \{1, \dots, m\}$  such that  $|X_k| \leq m$  is selected.

## Typical strategies

- **Stochastic gradient:**  $|X_k| = 1$ , choice of learning rate difficult, needs  $\alpha_k \searrow 0$ , sublinear convergence
  - **Mini-batch methods:**  $|X_k| = \gamma \ll m$ , less noise in the gradient, but choice of  $\gamma$  still difficult
  - **Gradient with dynamic accuracy:**  $|X_k| = \gamma_k \nearrow m$ , allows for convergence with constant  $\alpha$
- ⇒ But **no second order methods** based on subsampling with dynamic accuracy!

**Our proposal: a subsampled Levenberg–Marquardt method** with dynamic control of the accuracy

When to increase  $\gamma_k$ ?

- $f_{\delta_k}$  corresponding to  $X_k$
- If  $|f_{\delta_k}(x_k) - f(x_k)| \leq \frac{1}{2}\lambda_k \|p_k\|^2$ , then a reduction of  $f_{\delta_k} \Rightarrow$  a reduction of  $f$

$\Rightarrow$  Gives a criterion to dynamically increase  $\gamma_k$

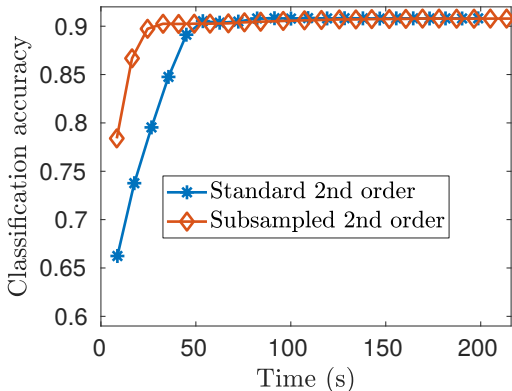
Theoretical properties

- Global convergence
- Worst case complexity:  $O(\epsilon^{-2})$  iterations to get  $\|\nabla f(x_k)\| \leq \epsilon$
- Local convergence at a linear rate

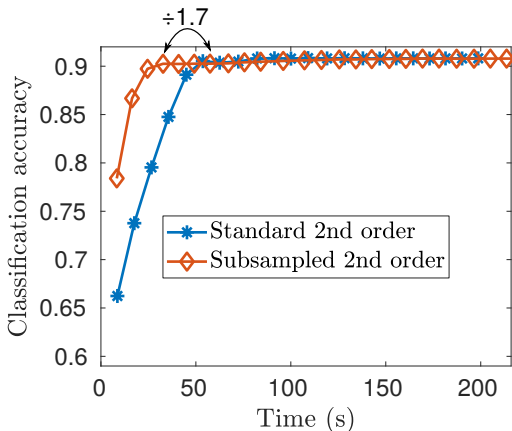


Bellavia, S. and Gratton, S. and Riccietti, E.. *A Levenberg-Marquardt method for large nonlinear least-squares problems with noisy functions and gradients.* Numer. Math. (2018).

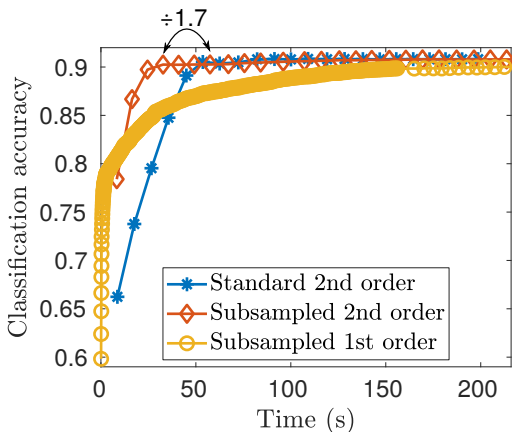
**Machine learning problem:** binary classification problem (CINA from the UCI machine-learning repository. Predict whether income exceeds a given amount based on census datas,  $m \sim 16000$ ).



**Machine learning problem:** binary classification problem (CINA from the UCI machine-learning repository. Predict whether income exceeds a given amount based on census datas,  $m \sim 16000$ ).

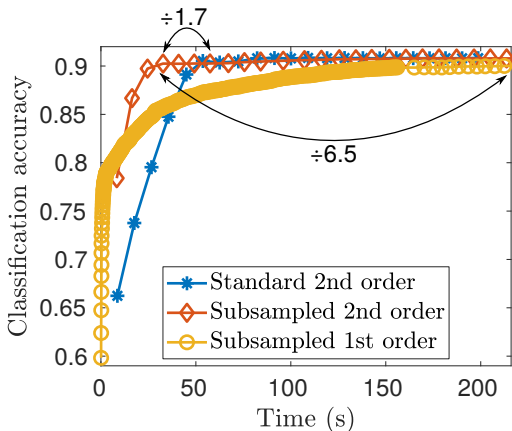


**Machine learning problem:** binary classification problem (CINA from the UCI machine-learning repository. Predict whether income exceeds a given amount based on census datas,  $m \sim 16000$ ).





**Machine learning problem:** binary classification problem (CINA from the UCI machine-learning repository. Predict whether income exceeds a given amount based on census datas,  $m \sim 16000$ ).



Ill-posed least squares problems

Zero residual problems

Nonzero residual problems

Large-scale nonzero residual problems

Large-scale problems

Subsampled methods for large  $m$

Multilevel methods for large  $n$

Conclusion & research project

We consider large-scale **nonlinear least-squares problems**:

$$\min_x f(x) = \frac{1}{2} \|F(x)\|^2, \quad F : \mathbb{R}^n \rightarrow \mathbb{R}^m.$$

## Typical application: deep learning

- $n = \#edges + \#nodes \Rightarrow$  very large for large & deep networks
- How to efficiently train the network?
- Common approach: stochastic gradient methods
  - They depend on **algorithmic parameters**, their choice may be difficult and it affects the convergence (a bad choice may prevent convergence)
  - They may be **slow** and better suited and studied for the **convex** case
  - They may be **inefficient** for complex networks architectures

## Hierarchy of problems

- $\{f_\ell(x_\ell)\}$ ,  $x_\ell \in \mathbb{R}^{n_\ell}$ ,  $n_{\ell-1} < n_\ell$
- $f_{\ell-1}$  is cheaper to optimize compared with  $f_\ell$

$$\begin{array}{ccc} x_k^\ell & & x_{k+1}^\ell = x_k^\ell + p_k^\ell \\ \downarrow R_\ell & & \uparrow p_k^\ell = P_\ell(x_{*,k}^{\ell-1} - x_{0,k}^{\ell-1}) \\ x_{0,k}^{\ell-1} := R_\ell x_k^\ell & \xrightarrow{\mu_{\ell-1}} & x_{*,k}^{\ell-1} \end{array}$$

- To compute the step  $p_k^\ell$  at level  $\ell$ , we minimize the function at level  $\ell-1$  using a model  $\mu_{\ell-1}$  (described later)
- The procedure is recursive: more levels can be used

# Theoretical results in a general framework

We consider the general framework of high-order methods in [Birgin et al, 2017] minimizing

$$T_q(x_k, p) + \frac{\lambda_k}{q+1} \|p\|^{q+1}, \quad (\text{order } q)$$

where  $T_q$  is the  $q$ th order Taylor series of  $f$

## A family of high-order multilevel methods

For a multilevel method of order  $q$ , we have proved its:

- **Global convergence**
- **Complexity**:  $\|\nabla f(x_k)\| \leq \epsilon$  in at most  $O(\epsilon^{-(q+1)/q})$  iterations
- **Local convergence** at a rate of order  $q$ , i.e.,  $\exists c > 0$

$$\lim_{k \rightarrow \infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|^q} \leq c$$



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

# Theoretical results in a general framework

We consider the general framework of high-order methods in [Birgin et al, 2017] minimizing

$$T_q(x_k, p) + \frac{\lambda_k}{q+1} \|p\|^{q+1}, \quad (\text{order } q)$$

where  $T_q$  is the  $q$ th order Taylor series of  $f$

## A family of high-order multilevel methods

For a multilevel method of order  $q$ , we have proved its:

- **Global convergence**
- **Complexity**:  $\|\nabla f(x_k)\| \leq \epsilon$  in at most  $O(\epsilon^{-(q+1)/q})$  iterations
- **Local convergence** at a rate of order  $q$ , i.e.,  $\exists c > 0$

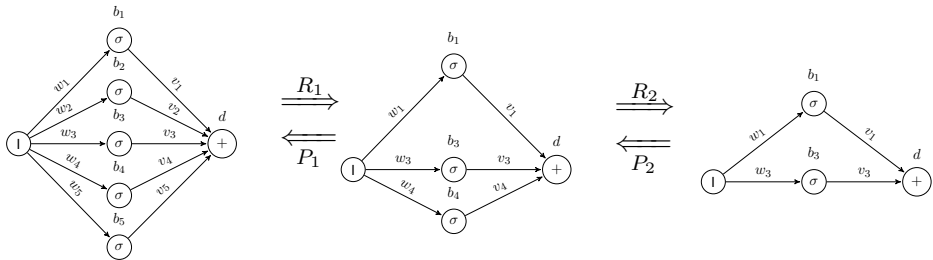
$$\lim_{k \rightarrow \infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|^q} \leq c$$



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

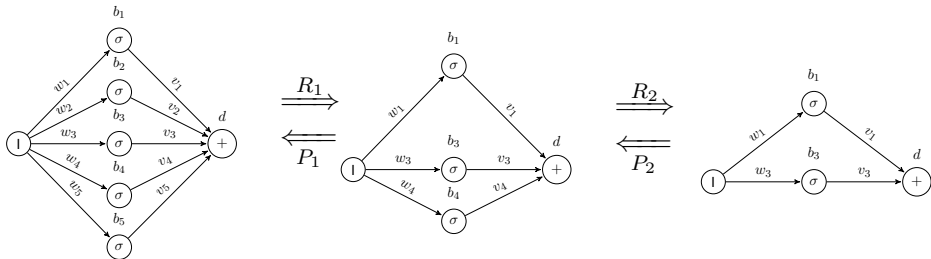
⇒ We now focus on the **multilevel Levenberg–Marquardt method**

# Multilevel training methods for ANNs



- Networks are algebraic objects, no geometry  $\Rightarrow$  how to build  $R$ ,  $P$ ?

# Multilevel training methods for ANNs



- Networks are **algebraic objects, no geometry**  $\Rightarrow$  how to build  $R$ ,  $P$ ?
- We propose the use of an **algebraic multigrid** approach [Ruge and Stueben] for  $Ax = b$  which only uses the matrix  $A$
- Which matrix should we use? We propose to use  $B_k \simeq \nabla^2 f(x_k)$ , which contains second order information



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*. Submitted to Optim. Methods Softw. (2019).



# Minimization problem at level $\ell$

Classical model to compute  $p_k$ :

$$\min_p f(x_k) + \nabla f(x_k)^T p + \frac{1}{2} p^T J(x_k)^T J(x_k) p + \frac{\lambda_k}{2} \|p\|^2$$

which leads to the linear system

$$(J(x_k)^T J(x_k) + \lambda_k I) p = -J(x_k)^T F(x_k)$$

known as **normal equations**.

# Minimization problem at level $\ell$

Classical model to compute  $p_k$ :

$$\min_p f(x_k) + \nabla f(x_k)^T p + \frac{1}{2} p^T J(x_k)^T J(x_k) p + \frac{\lambda_k}{2} \|p\|^2$$

which leads to the linear system

$$(J(x_k)^T J(x_k) + \lambda_k I) p = -J(x_k)^T F(x_k)$$

known as **normal equations**. However, with a multilevel method, the model to compute  $p_k$  at level  $\ell$  is instead:

$$\begin{aligned} \min_p f^\ell(x_k^\ell) + \nabla f^\ell(x_k^\ell)^T p + \frac{1}{2} p^T J^\ell(x_k^\ell)^T J^\ell(x_k^\ell) p + \frac{\lambda_k}{2} \|p\|^2 \\ + \underbrace{\left( R^{\ell+1} \nabla f^{\ell+1}(x_k^{\ell+1}) - \nabla f^\ell(x_{0,k}^\ell) \right)^T p}_{:=c} \end{aligned}$$

which leads to the linear system

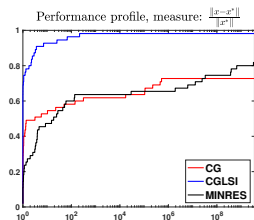
$$(J^\ell(x_k^\ell)^T J^\ell(x_k^\ell) + \lambda_k I) p = -J^\ell(x_k^\ell)^T F^\ell(x_k^\ell) + c$$

known as **extended normal equations**







# Numerical stability of solution of extended normal equations

- Extended normal equations  $A^T Ax = A^T b + c$  arise in several applications but their numerical solution is challenging
  - Specialized methods like CGLS cannot be used because of extra  $+c$  term, and general methods like CG are not numerically stable
- ⇒ We propose **CGLSI**, a new efficient and stable method outperforming classical iterative methods:

$$\widehat{A}^T \widehat{I} \widehat{A} x = \widehat{A}^T \widehat{b}, \quad \widehat{A} = \begin{bmatrix} A \\ c^T \end{bmatrix},$$
$$\widehat{I} = \begin{bmatrix} I_m & 0 \\ 0 & 0 \end{bmatrix}, \quad \widehat{b} = \begin{bmatrix} b \\ 1 \end{bmatrix}.$$



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).

-  Overcoming the curse of dimensionality in the numerical approximation of semilinear parabolic partial differential equations (2018).
-  The Deep Ritz method: A deep learning-based numerical algorithm for solving variational problems (2017)
-  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 (2018).
-  Solving stochastic differential equations and Kolmogorov equations by means of deep learning (2018).
-  Deep Neural Networks motivated by Partial Differential Equations (2018).

# Why try to solve PDEs with ANNs?

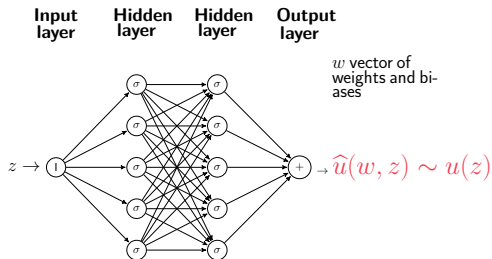
Compared with classical approaches (FDM, FEM), approaches using ANNs present the following advantages.

## 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**
- Allows to alleviate the effect of the **curse of dimensionality** (highly effective for more than 4,5 dimensions)
- The training is highly **parallelizable** on GPU

# Our approach: express the solution as a neural network

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



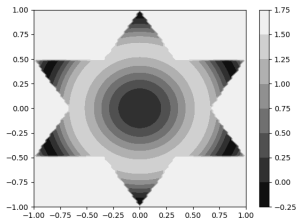
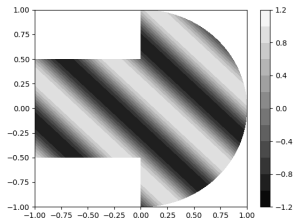
Optimization problem: find the network weights  $w$  by minimizing

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

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

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

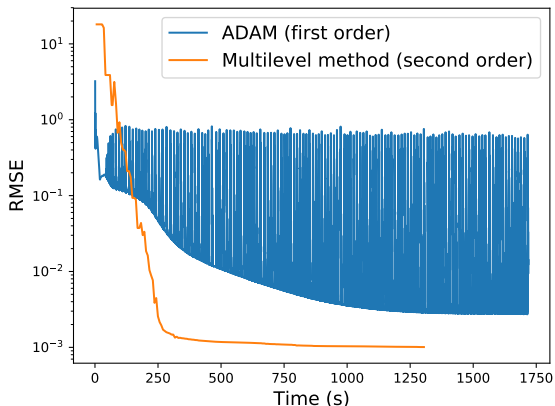
Right:  $-\Delta u + \nu u^2 = g_1$ ,  $u(x, y) = (x^2 + y^2) + \sin(\nu(x^2 + y^2))$ ,  $\nu = \frac{1}{2}$



	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

# Comparison with 1st order method ADAM (Tensorflow)

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



⇒ Solution to PDEs constitutes a challenging objective,  
first order methods struggle to achieve good training



Conclusion & research project

## A wide spectrum of novel second order methods ...

- **Regularizing, globally convergent trust-region method**
  - and its elliptical extension to handle nonzero residuals
  - and its Lanczos-based inexact extension to handle large problems
- **Subsampled Levenberg–Marquardt method**
  - with a dynamic control of the accuracy
- **Multilevel Levenberg–Marquardt method**
  - and its specialization to the training of neural networks
  - using a numerically stable solution to the extended normal equations

## A wide spectrum of novel second order methods ...

- **Regularizing, globally convergent trust-region method**
  - and its elliptical extension to handle nonzero residuals
  - and its Lanczos-based inexact extension to handle large problems
- **Subsampled Levenberg–Marquardt method**
  - with a dynamic control of the accuracy
- **Multilevel Levenberg–Marquardt method**
  - and its specialization to the training of neural networks
  - using a numerically stable solution to the extended normal equations

## ... for the solution of challenging problems

- Stable and fast solution of **large ill-posed problems** in **geophysics** and **biomedecine** (image registration)
  - Without any need of prior information
  - Possibly with nonzero residual
- Fast, high-quality **classification of large datasets**
- Fast, high-quality **training of deep neural networks**
- Promising preliminary results for the **solution of PDEs** with ANNs

Until now first order methods have been preferred to second order ones in the machine learning community: many variants of **gradient method** (stochastic, minibatch, accelerated, ...)

**BUT**

## New challenges in optimization for machine learning

- Increasingly **difficult problems** (highly nonlinear, nonconvex, ill-conditioned, many saddle points)
- Opportunities for **parallelism** and **mixed precision** with new hardware (GPUs with tensor cores, ...)

⇒ **Second order methods emerge as natural candidates to meet these challenges... but need to work on their cost to tackle increasingly large problems** ( $10^5$ – $10^6$  variables)

## Hessian free methods

(L-)BFGS methods approximate the inverse of the Hessian matrix using 1st order information to reduce the cost and memory. Open problems:

- Deal with **inexact gradients** (ex: BFGS + subsampling)
- **Exploit Hessian's structure** to include 2nd order information

## Hessian free methods

(L-)BFGS methods approximate the inverse of the Hessian matrix using 1st order information to reduce the cost and memory. Open problems:

- Deal with **inexact gradients** (ex: BFGS + subsampling)
- **Exploit Hessian's structure** to include 2nd order information

## Reduced/mixed precision methods

Many machine learning applications need limited accuracy and can thus leverage new hardware with reduced precisions. Open problems:

- Deal with **inexact function and gradients**, with inexactness coming from reduced precision (discrete set of precisions available)
- Linear systems in second order methods: from a drawback to an opportunity thanks to **mixed precision** and **GPU computing**

## Hessian free methods

(L-)BFGS methods approximate the inverse of the Hessian matrix using 1st order information to reduce the cost and memory. Open problems:

- Deal with **inexact gradients** (ex: BFGS + subsampling)
- **Exploit Hessian's structure** to include 2nd order information

## Reduced/mixed precision methods

Many machine learning applications need limited accuracy and can thus leverage new hardware with reduced precisions. Open problems:

- Deal with **inexact function and gradients**, with inexactness coming from reduced precision (discrete set of precisions available)
- Linear systems in second order methods: from a drawback to an opportunity thanks to **mixed precision** and **GPU computing**

## New applications in deep learning

Many applications possess an **underlying physics** and require **higher accuracy** than that provided by current deep learning techniques. Open problems:

- Inject classical numerical analysis techniques in networks (ex: **domain decomposition** to train **coupled GANs in parallel**)
- **High-order methods** for highly nonlinear and nonconvex problems

**Thank you for your attention!**

Slides and papers available here

[bit.ly/elisaIRIT](https://bit.ly/elisaIRIT)



# Backup slides

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

$$m_k^{LM}(p) = \frac{1}{2} \|R(x_k) + J(x_k)p\|^2 + \frac{1}{2} \lambda_k \|p\|^2.$$

- Set  $\Phi(x) = \frac{1}{2} \|R(x)\|^2$ , and compute

$$\rho_k(p_k) = \frac{\Phi(x_k) - \Phi(x_k + p_k)}{m_k^{LM}(0) - m_k^{LM}(p_k)}.$$

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

- Given  $x_k$  and the trust-region radius  $\Delta_k > 0$  find the step  $p_k$  solving

$$\begin{aligned} \min_p m_k^{TR}(p) &= \frac{1}{2} \|R(x_k) + J(x_k)p\|^2, \\ \text{s.t. } \|p\| &\leq \Delta_k \end{aligned}$$

- Set  $\Phi(x) = \frac{1}{2} \|R(x)\|^2$ . Compute

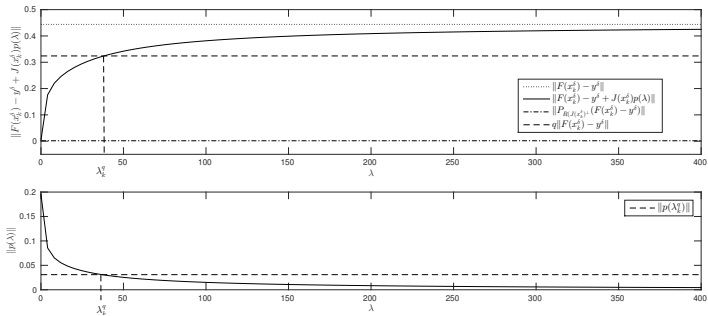
$$\rho_k(p_k) = \frac{\Phi(x_k) - \Phi(x_k + p_k)}{m_k^{TR}(0) - m_k^{TR}(p_k)}.$$

- Step acceptance and trust-region radius update. Given  $\eta \in (0, 1)$ :
  - If  $\rho_k < \eta$  then set  $\Delta_{k+1} < \Delta_k$  and  $x_{k+1} = x_k$ .
  - If  $\rho_k \geq \eta$  then set  $\Delta_{k+1} \geq \Delta_k$  and  $x_{k+1} = x_k + p_k$ .

**Iterative regularization methods** generate a sequence  $\{x_k\}$ . If the process is stopped at iteration  $k^*(\delta)$  the method is supposed to guarantee the following properties, given  $x^\dagger$  a solution of the unperturbed problem:

- $x_{k^*(\delta)}^\delta$  is an approximation of  $x^\dagger$ ;
- $\{x_{k^*(\delta)}^\delta\}$  tends to  $x^\dagger$  if  $\delta$  tends to zero;
- local convergence to  $x^\dagger$  in the noise-free case.

**2) q-condition:**  $\|F(x_k) - y^\delta + J(x_k)p\| \geq q\|F(x_k) - y^\delta\|$ ,  $q \in (0, 1)$



→ If  $\Delta_k \leq \frac{1-q}{\|B_k\|} \|g_k^\delta\|$  then  $p_k$  satisfies the q-condition and the trust region is active.

- **Assumption 1:** For index  $\bar{k}$  it exist positive  $\rho$  and  $c$  such that
  - 1 the system  $F(x) = y$  is solvable in  $B_\rho(x_{\bar{k}}^\delta)$ ;
  - 2 for  $x, \tilde{x} \in B_{2\rho}(x_{\bar{k}}^\delta)$  the following **tangential cone condition** holds

$$\|F(x) - F(\tilde{x}) - J(x)(x - \tilde{x})\| \leq c\|x - \tilde{x}\|\|F(x) - F(\tilde{x})\|.$$

For well-posed systems:  $\|F(x) - F(\tilde{x}) - J(x)(x - \tilde{x})\| \leq c\|x - \tilde{x}\|^2$ .

- **Assumption 2:** It exists positive  $K_J$  such that

$$\|J(x)\| \leq K_J$$

for all  $x \in \mathcal{L} = \{x \in \mathbb{R}^n \text{ s.t. } \Phi(x) \leq \Phi(x_0)\}$ .

- Four nonlinear ill-posed systems arising from the discretization of nonlinear first-kind Fredholm integral equation are considered, they model gravimetric and geophysics problems:

$$\int_0^1 k(t, s, x(s)) ds = y(t), \quad t \in [0, 1],$$

**P1,P2**, [Vogel, 1990], **P3,P4** [Kaltenbacher,2007];

- Their kernel is of the form

$$k(t, s, x(s)) = \log \left( \frac{(t-s)^2 + H^2}{(t-s)^2 + (H-x(s))^2} \right);$$
$$k(t, s, x(s)) = \frac{1}{\sqrt{1 + (t-s)^2 + x(s)^2}};$$

To maintain the regularizing properties of the trust-region approach we assume equivalent conditions on the gradient instead on the function.

1. **discrepancy principle :**

$$\|J(x_{k^*(\delta)})^T(F(x_{k^*(\delta)}) - y^\delta)\| \leq \tau\delta < \|J(x_k)^T(F(x_k) - y^\delta)\|$$

2. **q-condition:**

$$\|J(x_k)^T(F(x_k) - y^\delta + J(x_k)p_k)\| \geq q\|J(x_k)^T(F(x_k) - y^\delta)\|$$

If  $\Delta_k \leq \frac{1-q}{\|B_k\|^2} \|(B_k)^{1/2} g_k^\delta\|$  then  $p_k$  satisfies the q-condition and the trust-region is active.



- **Assumption1:** there exists  $\bar{k}$  s.t. a solution exists in  $B_\rho(x_{\bar{k}})$  and for  $x, \tilde{x} \in B_{2\rho}(x_{\bar{k}})$

$$\|\nabla f(\tilde{x}) - \nabla f(x) - J(x)^T J(x)(\tilde{x} - x)\| \leq (c\|\tilde{x} - x\| + \sigma)\|\nabla f(x) - \nabla f(\tilde{x})\|.$$

$$\begin{aligned}\nabla^2 f(x) &= J(x)^T J(x) + S(x) = \\ &J(x)^T J(x) + \sum_{j=1}^m (F_j(x) - y_j) \nabla^2 F_j(x).\end{aligned}$$

- **Assumption2:**  $\|S(x^\dagger)\| \leq \sigma < q < 1$  (small residual problems)

1. **P1:** We want to reconstruct  $c$  in the 2D-elliptic problem

$$\begin{aligned} -\Delta u + cu &= \hat{f} \text{ in } \Omega = (0, 1) \times (0, 1) \\ u &= \hat{g} \text{ on } \partial\Omega \end{aligned}$$

from the knowledge of  $u$  in  $\Omega$ ,  $\hat{f} \in L^2(\Omega)$ ,  $\hat{g}$  the trace of a function in  $H^2(\Omega)$ . If  $F : D(F) \rightarrow L^2(\Omega)$  is the operator mapping parameter  $c$  to the solution  $u$  we solve

$$\min_c \frac{1}{2} \|F(c) - \tilde{u}\|^2$$

$\tilde{u}$  measured values of  $u$ .

2. In case of noisy problems, given the error level  $\delta$ , the exact data  $y$  was perturbed by normally distributed values using the Matlab function `randn`, in a way that  $\|y - y^\delta\| = \delta$ .

## Large-scale problems: approximate solution of LM subproblem

- $p$  provides the **sufficient Cauchy decrease**:

$$m_k(0) - m_k(p_k) \geq \frac{\theta}{2} \frac{\|g_{\delta_k}(x_k)\|^2}{\|J_{\delta_k}(x_k)\|^2 + \lambda_k}, \quad \theta > 0.$$

- The Levenberg-Marquardt step computed as

$$(J_{\delta_k}(x_k)^T J_{\delta_k}(x_k) + \lambda_k I)p_k = -g_{\delta_k}(x_k) + r_k$$

for a residual  $r_k$  satisfying  $\|r_k\| \leq \epsilon_k \|g_{\delta_k}(x_k)\|$ , with  $\epsilon_k$  such that

$$0 \leq \epsilon_k \leq \min \left\{ \frac{\theta_1}{\lambda_k^\alpha}, \sqrt{\theta_2 \frac{\lambda_k}{\|J_{\delta_k}(x_k)\|^2 + \lambda_k}} \right\},$$

where  $\theta_1 > 0$ ,  $\theta_2 \in (0, \frac{1}{2}]$  and  $\alpha \in [\frac{1}{2}, 1)$  achieves the Cauchy decrease.

## Asymptotic step behaviour

The LM step asymptotically tends to the direction of the **negative perturbed gradient**:

$$\lim_{k \rightarrow \infty} (p_k^{LM})_i + \frac{\theta}{\kappa_J^2 + \lambda_k} (g_{\delta_k}(x_k))_i = 0 \quad \text{for } i = 1, \dots, n,$$

where  $(\cdot)_i$  denotes the  $i$ -th vector component.

## Lemma

Let  $p_k^{SD} = -\frac{\theta}{\kappa_J^2 + \lambda_k} g_{\delta_k}(x_k)$  and  $x_{k+1} = x_k + p_k^{SD}$ . If  $x_{\bar{k}} \in B_r(x^*)$  and  $\lambda_{\bar{k}}$  big enough,

- $\|x_{k+1} - x^*\| < \|x_k - x^*\|$ , for all  $k \geq \bar{k}$ .
- $\|x_k - x^*\|$  tends to zero.

- **Machine learning problem.** Binary classification problem:  $\{(z^i, y^i)\}$  with  $z^i \in \mathbb{R}^n$ ,  $y^i \in \{-1, +1\}$  and  $i = 1, \dots, N$ .  
Training objective function: logistic loss with  $l_2$  regularization

$$f(x) = \frac{1}{2N} \sum_{i=1}^N \log(1 + \exp(-y^i x^T z^i)) + \frac{1}{2N} \|x\|^2.$$

$$\|e_{k+1}\| < \|e_k\| + \rho(\ell), \quad \lim_{\ell \rightarrow n} \rho(\ell) = 0$$

