Algorithms and application for special classes of nonlinear least squares problems 2023

Clément W. Royer

US-Mexico workshop on optimization and its applications

January 9, 2023
Algorithms and application for special classes of nonlinear least squares problems

Stephen J. Wright

A Nonlinear Least Squares problem is an optimization problem for which the objective function to be minimized has the form

$$F(x) = \sum_{i=1}^{m} f_i^2(x), \quad x \in \mathbb{R}^n, \quad m \geq n.$$
Rebooting the (least-squares) franchise
Today’s goals

Nonlinear least squares
- A basic problem...
- ...with modern instances.

Revisit algorithms
- Complexity bounds for Gauss-Newton methods.
- Complexity metrics.

Go a bit further
- Inexactness and stochasticity.
- Probabilistic results.
1. Problem and first results
2. More complexity results
3. Beyond the deterministic setting
4. Application: Learning dynamics
Outline

1. Problem and first results
2. More complexity results
3. Beyond the deterministic setting
4. Application: Learning dynamics
General setup

Nonlinear least-squares

\[ \min_{x \in \mathbb{R}^n} f(x) := \frac{1}{2} \| r(x) \|^2, \quad r : \mathbb{R}^n \mapsto \mathbb{R}^m, \ r \in C^2. \]

Jacobian: \[ J(x) := [\nabla r_i(x)^\top] \in \mathbb{R}^{m \times n}. \]
General setup

Nonlinear least-squares

\[ \min_{x \in \mathbb{R}^n} f(x) := \frac{1}{2} \| r(x) \|^2, \quad r : \mathbb{R}^n \mapsto \mathbb{R}^m, \ r \in C^2. \]

Jacobian: \( J(x) := [ \nabla r_i(x)^\top ] \in \mathbb{R}^{m \times n} \).

Gauss-Newton techniques

- Gauss-Newton model \( f(x + s) \approx \frac{1}{2} \| r(x) + J(x)s \|^2 \);
- Steps computed via line search/trust region;
- Hessian approximated by \( J(x)^\top J(x) \).
General setup

Nonlinear least-squares

\[
\min_{x \in \mathbb{R}^n} f(x) := \frac{1}{2} \| r(x) \|^2, \quad r : \mathbb{R}^n \mapsto \mathbb{R}^m, \ r \in C^2.
\]

**Jacobian:** \( J(x) := \left[ \nabla r_i(x)^\top \right] \in \mathbb{R}^{m \times n}. \)

Gauss-Newton techniques

- Gauss-Newton model \( f(x + s) \approx \frac{1}{2} \| r(x) + J(x)s \|^2; \)
- Steps computed via line search/trust region;
- Hessian approximated by \( J(x)^\top J(x). \)

Levenberg(-Morrison)-Marquardt

- Regularized Gauss-Newton model:
  \[
  f(x + s) \approx \frac{1}{2} \| r(x) + J(x)s \|^2 + \frac{\gamma}{2} \| s \|^2;
  \]
- **Regularization parameter** \( \gamma \) set adaptively.
Inputs: $x_0 \in \mathbb{R}^n$, $\gamma_0 \geq \gamma_{\text{min}} > 0$, $\eta > 0$.

Iteration $k$: Given $(x_k, \gamma_k)$,

- Compute

$$s_k \approx \arg\min_s m_k(s) := \frac{1}{2} \| r(x_k) + J(x_k)s \|^2 + \frac{\gamma_k}{2} \| s \|^2.$$

- If

$$\frac{1}{2} \| r(x_k) \|^2 - \frac{1}{2} \| r(x_k + s_k) \|^2 \geq \eta$$

$$m_k(0) - m_k(s_k)$$

set $x_{k+1} = x_k + s_k$ and $\gamma_{k+1} = \max\{0.5\gamma_k, \gamma_{\text{min}}\}$;

- Otherwise, set $x_{k+1} = x_k$ and $\gamma_{k+1} = 2\gamma_k$. 

Goal: Prove a complexity result for the method.
Levenberg-Marquardt for $\min_{x \in \mathbb{R}^n} \frac{1}{2} \| r(x) \|^2$

**Inputs:** $x_0 \in \mathbb{R}^n$, $\gamma_0 \geq \gamma_{\text{min}} > 0$, $\eta > 0$.

**Iteration $k$:** Given $(x_k, \gamma_k)$,

- Compute
  
  $$s_k \approx \arg\min_s m_k(s) := \frac{1}{2} \| r(x_k) + J(x_k)s \|^2 + \frac{\gamma_k}{2} \| s \|^2.$$

- If
  
  $$\frac{\frac{1}{2} \| r(x_k) \|^2 - \frac{1}{2} \| r(x_k + s_k) \|^2}{m_k(0) - m_k(s_k)} \geq \eta,$$

  set $x_{k+1} = x_k + s_k$ and $\gamma_{k+1} = \max\{0.5\gamma_k, \gamma_{\text{min}}\}$;

- Otherwise, set $x_{k+1} = x_k$ and $\gamma_{k+1} = 2\gamma_k$.

**Goal:** Prove a complexity result for the method.
Setup: Sequence of points $\{x_k\}$ generated by an algorithm applied to $
abla f(x)$. 

**Standard complexity result** 

Given $\epsilon \in (0, 1)$: 

Worst-case number of iterations to obtain $x_k$ such that $\|\nabla f(x_k)\| \leq \epsilon$.

**Focus:** Dependency on $\epsilon$.

Some examples:

Gradient descent: $O(\epsilon^{-2/3})$ iterations.

Newton: $O(\epsilon^{-2/3})$ iterations.

Cubic regularization/Modified Newton: $O(\epsilon^{-3/2})$ iterations.
**Setup:** Sequence of points \( \{x_k\} \) generated by an algorithm applied to \( \min_{x \in \mathbb{R}^n} f(x) \).

**Standard complexity result**

Given \( \epsilon_g \in (0, 1) \):

- **Worst-case** number of iterations to obtain \( x_k \) such that 
  \[ \|\nabla f(x_k)\| \leq \epsilon_g. \]
- **Focus:** Dependency on \( \epsilon_g \).
**Setup:** Sequence of points \( \{x_k\} \) generated by an algorithm applied to \( \min_{x \in \mathbb{R}^n} f(x) \).

**Standard complexity result**

Given \( \epsilon_g \in (0, 1) \):

- **Worst-case** number of iterations to obtain \( x_k \) such that \( \|\nabla f(x_k)\| \leq \epsilon_g \).
- **Focus:** Dependency on \( \epsilon_g \).

**Some examples**

- Gradient descent: \( \mathcal{O}(\epsilon_g^{-2}) \) iterations.
- Newton: \( \mathcal{O}(\epsilon_g^{-2}) \) iterations.
- Cubic regularization/Modified Newton: \( \mathcal{O}(\epsilon_g^{-3/2}) \) iterations.
The least-squares setting

Problem: \( \min_{x \in \mathbb{R}^n} \frac{1}{2} \| r(x) \|^2 \):

Goal: Find \( x_k \) such that \( \| J(x_k)^T r(x_k) \| \leq \epsilon_g \).
The least-squares setting

Problem: \( \min_{x \in \mathbb{R}^n} \frac{1}{2} \| r(x) \|^2 \):

Goal: Find \( x_k \) such that \( \| J(x_k)^T r(x_k) \| \leq \varepsilon_g \).

Some results

- Gradient descent: \( O(\varepsilon_g^{-2}) \) iterations.
- Gauss-Newton + line search/trust region: \( O(\varepsilon_g^{-2}) \) iterations.
- Levenberg-Marquardt: \( O(\varepsilon_g^{-2}) \) iterations.
Levenberg-Marquardt for $\min_{x \in \mathbb{R}^n} \frac{1}{2} \| r(x) \|^2$

**Inputs:** $x_0 \in \mathbb{R}^n$, $\gamma_0 \geq \gamma_{\text{min}} > 0$, $\eta > 0$.

**Iteration $k$:** Given $(x_k, \gamma_k)$,

- Compute

  $$s_k \approx \arg\min_s m_k(s) := \frac{1}{2} \| r(x_k) + J(x_k)s \|^2 + \frac{\gamma_k}{2} \| s \|^2.$$

- If

  $$\frac{\frac{1}{2} \| r(x_k) \|^2 - \frac{1}{2} \| r(x_k + s_k) \|^2}{m_k(0) - m_k(s_k)} \geq \eta,$$

  set $x_{k+1} = x_k + s_k$ and $\gamma_{k+1} = \max\{0.5\gamma_k, \gamma_{\text{min}}\}$;

- Otherwise, set $x_{k+1} = x_k$ and $\gamma_{k+1} = 2\gamma_k$.

Goal: Prove a complexity result for the method.
Levenberg-Marquardt for \( \min_{x \in \mathbb{R}^n} \frac{1}{2} \| r(x) \|^2 \)

**Inputs:** \( x_0 \in \mathbb{R}^n, \gamma_0 \geq \gamma_{\min} > 0, \eta > 0. \)

**Iteration \( k \):** Given \((x_k, \gamma_k)\),

- Compute

\[
s_k \approx \arg\min_s m_k(s) := \frac{1}{2} \| r(x_k) + J(x_k)s \|^2 + \frac{\gamma_k}{2} \| s \|^2.
\]

- If

\[
\frac{\frac{1}{2} \| r(x_k) \|^2 - \frac{1}{2} \| r(x_k + s_k) \|^2}{m_k(0) - m_k(s_k)} \geq \eta,
\]

set \( x_{k+1} = x_k + s_k \) and \( \gamma_{k+1} = \max\{0.5\gamma_k, \gamma_{\min}\} \);

- Otherwise, set \( x_{k+1} = x_k \) and \( \gamma_{k+1} = 2\gamma_k \).

**Goal:** Prove a complexity result for the method.
Main arguments

Decrease guarantee

For any successful iteration \((x_{k+1} \neq x_k)\),

\[ ||r(x_k)||^2 - ||r(x_{k+1})||^2 \geq O \left( \frac{||J(x_k)^T r(x_k)||^2}{\gamma_k} \right). \]
Main arguments

Decrease guarantee

For any successful iteration \((x_{k+1} \neq x_k)\),

\[
\| r(x_k) \|^2 - \| r(x_{k+1}) \|^2 \geq O \left( \frac{\| J(x_k) \,^T \, r(x_k) \|^2}{\gamma_k} \right).
\]

Regularization parameter

- If \(\gamma_k\) large enough, the iteration is successful.
- \(\gamma_k \leq \gamma_{\text{max}}\) for all \(k\).
Main arguments

Decrease guarantee

For any successful iteration \((x_{k+1} \neq x_k)\),

\[
\|r(x_k)\|^2 - \|r(x_{k+1})\|^2 \geq O\left(\frac{\|J(x_k)^T r(x_k)\|^2}{\gamma_k}\right).
\]

Regularization parameter

- If \(\gamma_k\) large enough, the iteration is successful.
- \(\gamma_k \leq \gamma_{\text{max}}\) for all \(k\).

Complexity of standard Levenberg-Marquardt

The method reaches \(x_k\) such that \(\|J(x_k)^T r(x_k)\| \leq \epsilon_g\) in at most \(O(\epsilon_g^{-2})\) iterations.
1 Problem and first results

2 More complexity results

3 Beyond the deterministic setting

4 Application: Learning dynamics
An alternate complexity measure

Our problem: $\min_{x \in \mathbb{R}^n} \frac{1}{2} \| r(x) \|^2$

- Used $\| J(x)^T r(x) \|$ as a complexity metric;
- Oblivious to the least-square structure;
- May want to stop when residuals are small.

Scaled gradient (Cartis, Gould, Toint '13; Gould, Rees, Scott '19)

$g(x) := \begin{cases} 
J(x)^T r(x) & \text{if } \| r(x) \| > 0 \\
0 & \text{otherwise.}
\end{cases}$

Stopping criterion for complexity:

$\| r(x) \| \leq \epsilon_r$ or $\| g(x) \| \leq \epsilon_g$. 

C. W. Royer

Nonlinear Least Squares Problems 2023

Huatulco 2023
An alternate complexity measure

Our problem: \( \min_{x \in \mathbb{R}^n} \frac{1}{2} \| \mathbf{r}(x) \|^2 \)

- Used \( \| \mathbf{J}(x)^T \mathbf{r}(x) \| \) as a complexity metric;
- Oblivious to the least-square structure;
- May want to stop when residuals are small.

Scaled gradient (Cartis, Gould, Toint '13; Gould, Rees, Scott '19)

\[
\mathbf{g}(x) := \begin{cases} 
\frac{\mathbf{J}(x)^T \mathbf{r}(x)}{\| \mathbf{r}(x) \|} & \text{if } \| \mathbf{r}(x) \| > 0 \\
0 & \text{otherwise.}
\end{cases}
\]

- Stopping criterion for complexity:

\[
\| \mathbf{r}(x) \| \leq \epsilon_r \quad \text{or} \quad \| \mathbf{g}(x) \| \leq \epsilon_g.
\]
An alternate complexity measure ('ed)

**Goal:** Find $x_k$ such that

$$\|r(x_k)\| \leq \epsilon_r \quad \text{or} \quad \|g(x_k)\| \leq \epsilon_g, \quad g(x_k) := \begin{cases} \frac{J(x_k)^T r(x_k)}{\|r(x_k)\|} & \text{if } \|r(x_k)\| > 0 \\ 0 & \text{otherwise.} \end{cases}$$
**Goal:** Find $x_k$ such that

$$\|r(x_k)\| \leq \epsilon_r \quad \text{or} \quad \|g(x_k)\| \leq \epsilon_g,$$

$$g(x_k) := \begin{cases} \frac{J(x_k)^T r(x_k)}{\|r(x_k)\|} & \text{if } \|r(x_k)\| > 0 \\ 0 & \text{otherwise.} \end{cases}$$

**Complexity of Levenberg-Marquardt (Gould, Rees, Scott '19)**

For any $i \in \mathbb{N} \cup \{-1\}$, the method finds a suitable $x_k$ in at most

$$O(2^i \epsilon_g^{-2} \epsilon_r^{-1/2} i)$$

iterations.
An alternate complexity measure ('ed)

Goal: Find $x_k$ such that

$$\|r(x_k)\| \leq \epsilon_r \quad \text{or} \quad \|g(x_k)\| \leq \epsilon_g,$$

$$g(x_k) := \begin{cases} 
\frac{J(x_k)^T r(x_k)}{\|r(x_k)\|} & \text{if } \|r(x_k)\| > 0 \\
0 & \text{otherwise.}
\end{cases}$$

Complexity of Levenberg-Marquardt (Gould, Rees, Scott '19)

For any $i \in \mathbb{N} \cup \{-1\}$, the method finds a suitable $x_k$ in at most

$$O(2^i \epsilon^{-2} \epsilon^{-1/2^i}_r) \quad \text{iterations.}$$

- Part of more results on high-order regularization methods.
- Asymptotically: $\epsilon^{-1/2^i}_r \to 1$ but $2^i \to \infty$. 
New scaled gradient

Given $i \in \mathbb{N} \cup \{-1\}$,

$$g^i(x) := \begin{cases} \frac{\|J(x)^T r(x)\|}{\|r(x)\|^{2-2^{-i}}} & \text{if } \|r(x)\| \neq 0, \\ 0 & \text{otherwise.} \end{cases}$$

- Stopping criterion for complexity:

$$\|r(x)\| \leq \epsilon_r \quad \text{or} \quad \|g^i(x)\| \leq \epsilon_g.$$
New scaled gradient

Given $i \in \mathbb{N} \cup \{-1\}$,

$$g^i(x) := \begin{cases} 
\frac{\|J(x)^T r(x)\|}{\|r(x)\|^{2^{-2^{-i}}}} & \text{if } \|r(x)\| \neq 0, \\
0 & \text{otherwise.}
\end{cases}$$

- Stopping criterion for complexity:

$$\|r(x)\| \leq \epsilon_r \quad \text{or} \quad \|g^i(x)\| \leq \epsilon_g.$$

- $i = -1$: Classical gradient;
- $i = 0$: CGT scaled gradient;
- $i \to \infty$: $\|g^i(x)\| > \epsilon_g$ akin to gradient dominance.
Decrease guarantees

For any successful iteration \((x_{k+1} \neq x_k)\),

\[
\| r(x_k) \|^2 - \| r(x_{k+1}) \|^2 \geq O \left( \frac{\| J(x_k)^T r(x_k) \|^2}{\gamma_k} \right)
\]

and (if \( \| r(x_k) \| \neq 0 \))

\[
\| r(x_k) \|^{1/2^i} - \| r(x_{k+1}) \|^{1/2^i} \geq O \left( \frac{\| g^i(x_k) \|^2 \| r(x_k) \|^{(4-2^{1-i})}}{\gamma_k} \right).
\]
Main arguments

Decrease guarantees

For any successful iteration \((x_{k+1} \neq x_k)\),

\[
\|r(x_k)\|^2 - \|r(x_{k+1})\|^2 \geq O\left(\frac{\|J(x_k)^T r(x_k)\|^2}{\gamma_k}\right)
\]

and (if \(\|r(x_k)\| \neq 0\))

\[
\|r(x_k)\|^\frac{1}{2^i} - \|r(x_{k+1})\|^\frac{1}{2^i} \geq O\left(\frac{\|g^i(x_k)\|^2 \|r(x_k)\|^{(4-2^{1-i})}}{\gamma_k}\right).
\]

Regularization parameter

- If \(\gamma_k\) large enough, the iteration is successful.
- \(\gamma_k \leq \gamma_{\text{max}}\) for all \(k\).
**Goal:** Find $x_k$ such that

$$\| r(x_k) \| \leq \epsilon_r \quad \text{or} \quad \| g^i(x_k) \| \leq \epsilon_g.$$ 

### Complexity results (BDKR ’22)

<table>
<thead>
<tr>
<th>$i$</th>
<th>Arbitrary</th>
<th>$i = -1$</th>
<th>$i = 0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$g^i(x)$</td>
<td>$\frac{|J(x)^T r(x)|}{|r(x)|^2 - 2^{-i}}$</td>
<td>$|J(x)^T r(x)|$</td>
<td>$\frac{|J(x)^T r(x)|}{|r(x)|}$</td>
</tr>
<tr>
<td>Order</td>
<td>$\epsilon_g^{-2} \epsilon_r^{-(4 - 2^{1-i})}$</td>
<td>$\epsilon_g^{-2}$</td>
<td>$\epsilon_g^{-2} \epsilon_r^{-2}$</td>
</tr>
</tbody>
</table>

- Matches existing results for $i = -1$.
- For $i = 0$: previous results get better bounds in terms of $\epsilon_r$ but with very large constants ($2^i$).
Problem and first results

More complexity results

Beyond the deterministic setting

Application: Learning dynamics
A stochastic problem

**Stochastic nonlinear least-squares**

\[
\min_{x \in \mathbb{R}^n} f(x) = \frac{1}{2} \| r(x) \|^2
\]

- Values of \( r \) and Jacobian \( J \) only accessed through stochastic estimates.

**Challenges**

- Every evaluation is replaced by a random estimate;
- Decrease no longer guaranteed;
- Accuracy of evaluation matters.
Inputs: $x_0 \in \mathbb{R}^n$, $\gamma_0 \geq \gamma_{\text{min}} > 0$.

Iteration $k$: Given $(x_k, \gamma_k)$,

- Compute $r_{mk} \approx r(x_k)$, $J_{mk} \approx J(x_k)$ and $s_k \approx \arg\min_s m_k(s) = \frac{1}{2} \|r_{mk} + J_{mk}s\|^2 + \frac{\gamma_k}{2} \|s\|^2$.
- Compute $r^0_k \approx r(x_k)$ and $r^s_k \approx r(x_k + s_k)$.
- If $\frac{1}{2} \|r^0_k\|^2 - \frac{1}{2} \|r^s_k\|^2 \geq \eta$, set $x_{k+1} = x_k + s_k$ and $\gamma_{k+1} = \max\{0.5\gamma_k, \gamma_{\text{min}}\}$;
- Otherwise, set $x_{k+1} = x_k$ and $\gamma_{k+1} = 2\gamma_k$. 
Using inexact values

**Inputs:** $x_0 \in \mathbb{R}^n$, $\gamma_0 \geq \gamma_{\text{min}} > 0$.

**Iteration $k$:** Given $(x_k, \gamma_k)$,

- Compute $r_{m_k} \approx r(x_k)$, $J_{m_k} \approx J(x_k)$ and $s_k \approx \arg\min_s m_k(s) = \frac{1}{2} \| r_{m_k} + J_{m_k} s \|^2 + \frac{\gamma_k}{2} \| s \|^2$.
- Compute $r^0_k \approx r(x_k)$ and $r^s_k \approx r(x_k + s_k)$.
- If $\frac{1}{2} \| r^0_k \|^2 - \frac{1}{2} \| r^s_k \|^2 \geq \eta$, set $x_{k+1} = x_k + s_k$ and $\gamma_{k+1} = \max\{0.5 \gamma_k, \gamma_{\text{min}}\}$;
- Otherwise, set $x_{k+1} = x_k$ and $\gamma_{k+1} = 2 \gamma_k$.

**Goal:** Prove a complexity result for this inexact method.

**Key:** Use $\gamma_k$ to monitor the inexactness and the convergence.
Complexity analysis in an inexact setting

Accuracy requirements (model)

For every $k$,

$$\left\| J(x_k)^\top r(x_k) - \left( J^\top_{m_k} r_{m_k} \right) \right\| \leq O \left( \frac{1}{\gamma_k} \right)$$

and

$$\left| \frac{1}{2} \left\| r(x_k) \right\|^2 - \frac{1}{2} \left\| r_{m_k} \right\|^2 \right| \leq O \left( \frac{1}{\gamma_k} \right).$$

Accuracy requirements (evaluations)

For every $k$,

$$\left| \frac{1}{2} \left\| r_0^k \right\|^2 - \frac{1}{2} \left\| r(x_k) \right\|^2 \right| \leq O \left( \frac{1}{\gamma_k^2} \right)$$

and

$$\left| \frac{1}{2} \left\| r_s^k \right\|^2 - \frac{1}{2} \left\| r(x_k + s_k) \right\|^2 \right| \leq O \left( \frac{1}{\gamma_k^2} \right).$$
Using inexactness

- With the same theory as in the exact case, get $O(\epsilon^{-3})$ instead of $O(\epsilon^{-2})$ to obtain $\|J(x_k)^T r(x_k)\| \leq \epsilon$.
Using inexactness

- With the same theory as in the exact case, get $O(\epsilon^{-3})$ instead of $O(\epsilon^{-2})$ to obtain $\|J(x_k)^T r(x_k)\| \leq \epsilon!$
- Arguments:
  - Still decrease in $O\left(\frac{\|J(x_k)^T r(x_k)\|^2}{\gamma_k}\right)$;
  - But now $\gamma_k$ grows as $O(\epsilon^{-1})$!
Complexity analysis

Using inexactness

- With the same theory as in the exact case, get $O(\epsilon^{-3})$ instead of $O(\epsilon^{-2})$ to obtain $\|J(x_k)^T r(x_k)\| \leq \epsilon$!
- Arguments:
  - Still decrease in $O\left(\frac{\|J(x_k)^T r(x_k)\|^2}{\gamma_k}\right)$;
  - But now $\gamma_k$ grows as $O(\epsilon^{-1})$!

A fix (BDKR '22)

- The analysis reveals $\gamma = O(\|J(x)^T r(x)\|/\|s\|)$;
- By analogy with trust-region, we want $\gamma = 1/\|s\|$;
- A scaling will help us achieve that.
**A corrected Levenberg-Marquardt method**

**Inputs:** \( x_0 \in \mathbb{R}^n, \gamma_0 \geq \gamma_{\text{min}} > 0. \)

**Iteration \( k \):** Given \((x_k, \gamma_k)\),
- Compute \( r_{mk} \approx r(x_k), \ J_{mk} \approx J(x_k) \) and \( s_k \approx \text{argmin}_s \ m_k(s) = \frac{1}{2} \| r_{mk} + J_{mk} s \|^2 + \frac{\gamma_k \| J_{mk}^T r_{mk} \|}{2} \| s \|^2 \).
- Compute \( r_k^0 \approx r(x_k) \) and \( r_k^s \approx r(x_k + s_k) \).
- If \( \frac{1}{2} \| r_k^0 \|^2 - \frac{1}{2} \| r_k^s \|^2 \frac{m_k(0) - m_k(s)}{m_k(0) - m_k(s)} \geq \eta \), set \( x_{k+1} = x_k + s_k \) and \( \gamma_{k+1} = \max\{0.5 \gamma_k, \gamma_{\text{min}}\}; \)
- Otherwise, set \( x_{k+1} = x_k \) and \( \gamma_{k+1} = 2 \gamma_k \).
A corrected Levenberg-Marquardt method

**Inputs:** \( \mathbf{x}_0 \in \mathbb{R}^n, \gamma_0 \geq \gamma_{\text{min}} > 0. \)

**Iteration \( k \):** Given \((\mathbf{x}_k, \gamma_k)\),

- Compute \( \mathbf{r}_{m_k} \approx \mathbf{r}(\mathbf{x}_k), \mathbf{J}_{m_k} \approx \mathbf{J}(\mathbf{x}_k) \) and
  \[
  \mathbf{s}_k \approx \text{argmin}_s \ m_k(s) = \frac{1}{2}\|\mathbf{r}_{m_k} + \mathbf{J}_{m_k}\mathbf{s}\|^2 + \frac{\gamma_k\|\mathbf{J}_{m_k}^\top\mathbf{r}_{m_k}\|}{2}\|\mathbf{s}\|^2.
  \]
- Compute \( \mathbf{r}_k^0 \approx \mathbf{r}(\mathbf{x}_k) \) and \( \mathbf{r}_k^s \approx \mathbf{r}(\mathbf{x}_k + \mathbf{s}_k) \).
- If \( \frac{1}{2}\|\mathbf{r}_k^0\|^2 - \frac{1}{2}\|\mathbf{r}_k^s\|^2 \geq \eta \), set \( \mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{s}_k \) and \( \gamma_{k+1} = \max\{0.5\gamma_k, \gamma_{\text{min}}\} \);
- Otherwise, set \( \mathbf{x}_{k+1} = \mathbf{x}_k \) and \( \gamma_{k+1} = 2\gamma_k \).

- **Two sources of inexactness** (models/estimates);
- Analysis can be deterministic or probabilistic.
Probabilistic models

Accuracy property

For any realization, \((J_{m_k}, r_{m_k})\) is called accurate if

\[
\|J(x_k)^T r(x_k) - J_{m_k}^T r_{m_k}\| \leq O\left(\frac{1}{\gamma_k}\right) \quad \left|\frac{1}{2}\|r(x_k)\|^2 - \frac{1}{2}\|r_{m_k}\|^2\right| \leq O\left(\frac{1}{\gamma_k^2}\right).
\]

Probabilistic accuracy property

The random model sequence \(\{(J_{m_k}, r_{m_k})\}\) is called \(p\)-accurate if

\[
\forall k, \quad P\left((J_{m_k}, r_{m_k}) \text{ accurate} \mid F_{k-1}\right) \geq p.
\]

\(F_{k-1} = \sigma(m_0, \ldots, m_{k-1}, r_0^0, r_0^s, \ldots, r_{k-1}^0, r_{k-1}^s)\) represents the history of the algorithm up to iteration \(k\).
Probabilistic function estimates

Accurate function estimates

\[
\left| \frac{1}{2} \| r_k^0 \|^2 - \frac{1}{2} \| r(x_k) \|^2 \right| \leq O \left( \frac{1}{\gamma_k^2} \right)
\]

\[
\left| \frac{1}{2} \| r_k^s \|^2 - \frac{1}{2} \| r(x_k + s_k) \|^2 \right| \leq O \left( \frac{1}{\gamma_k^2} \right).
\]

Probabilistically accurate estimates

The random estimate sequence \{\( (r_k^0, r_k^s) \)\} is \( q \)-accurate if

\[
\forall k, \quad \mathbb{P} \left( (r_k^0, r_k^1) \text{ accurate} \mid \mathcal{F}_{k-1/2} \right) \geq q.
\]

\[
\mathcal{F}_{k-1/2} = \sigma(m_0, \ldots, m_{k-1}, m_k, r_0^0, r_0^s, \ldots, r_{k-1}^0, r_{k-1}^s)
\]
represents the iteration of the algorithm up to the computation of \( r_k^0 \) and \( r_k^s \).
Goal: Bound the \textit{stopping time}

\[ K_\epsilon = \min \{ k \mid \| r(x) \| \leq \epsilon_r \text{ or } \| g^i(x) \| \leq \epsilon_g \}. \]
Probabilistic complexity results

Goal: Bound the stopping time

\[ K_\epsilon = \min\{ k \mid \| r(x) \| \leq \epsilon_r \text{ or } \| g^i(x) \| \leq \epsilon_g \}. \]

Theorem (BDKR '22)

If \( \{ (J_{m_k}, r_{m_k}) \} \) are \( p \)-accurate and \( \{ (r^0_k, r^s_k) \} \) are \( q \)-accurate, then

\[ \mathbb{E}[K_\epsilon] \leq O \left( \frac{pq}{pq - 1/2} \epsilon_g^{2} \epsilon_r^{-(4-2^{1-i})} \right). \]
Outline

1. Problem and first results
2. More complexity results
3. Beyond the deterministic setting
4. Application: Learning dynamics
Motivation: Learning ODE parameters

Problem

- **Data:** \( \{z(t_i)\}_{i=0}^{m} \) obtained from the solution \( z(t) \) of an ODE

\[
\frac{dz}{dt}(t) = \phi_A(z(t)).
\]

with \( z(0) = z_0 \).

- **Goal:** Learn the parameters \( A \) of the dynamics \( \phi \).
Motivation: Learning ODE parameters

**Problem**

- **Data:** \( \{z(t_i)\}_{i=0}^{m} \) obtained from the solution \( z(t) \) of an ODE

\[
\frac{dz}{dt}(t) = \phi_A(z(t)).
\]

with \( z(0) = z_0 \).

- **Goal:** Learn the parameters \( A \) of the dynamics \( \phi \).

**Model: NeuralODE**

- A neural network defined as the solution of an ODE: \( z \mapsto y(1) \), where \( y \) solution of

\[
\frac{dy}{dt}(t) = \phi_X(y(t))
\]

with \( y(0) = y_0 \).

- **Goal:** Learn \( X \) close to \( A \).
Illustration: Linear ODE

**Problem**
- Noisy data \( \{z_i\}_{i=0}^m \) generated by a linear ODE \( \frac{dz}{dt}(t) = Az(t) \);
- Closed-form expression: \( z(t) = e^{At}z(0) \).

**Training problem**

\[
\text{minimize} \quad \frac{1}{m} \sum_{i=1}^m \left\| \left( I + \frac{X}{\ell} \right)^\ell z_i - z_{i+1} \right\|^2
\]

- Euler’s formula:
  \[
e^X \approx \left( I + \frac{X}{\ell} \right)^\ell, \ell \geq 1.
\]
- Nonconvex nonlinear least squares for \( \ell \geq 2 \) (strict, even high-order saddle points).
Setup

- 100 trajectories on a spiral (2-dimensional linear ODE)
- Comparison: Levenberg-Marquardt with two complexity metrics as stopping criteria.
Applying Levenberg-Marquardt

Setup

- 100 trajectories on a spiral (2-dimensional linear ODE)
- Comparison: Levenberg-Marquardt with two complexity metrics as stopping criteria.

<table>
<thead>
<tr>
<th>Criterion</th>
<th>Best error in $X_*$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$|J(x_k)^T r(x_k)| \leq 10^{-3}$</td>
<td>49</td>
</tr>
<tr>
<td>$\frac{|J(x_k)^T r(x_k)|}{|r(x_k)|} \leq 10^{-3}$ or $|r(x_k)| \leq 10^{-6}$</td>
<td>56.</td>
</tr>
</tbody>
</table>
Complexity and nonlinear least squares

- A family of complexity metrics and results.
- Derived for Gauss-Newton methods (Levenberg-Marquardt type).
- Works with inexact/stochastic values and derivatives.

Complexity and nonlinear least squares

- A family of complexity metrics and results.
- Derived for Gauss-Newton methods (Levenberg-Marquardt type).
- Works with inexact/stochastic values and derivatives.


Next

- Gauss-Newton vs Newton steps?
- Application to NeuralODE training.
- Go beyond least squares (cross-entropy loss).

Thank you, and happy birthday Steve!

<table>
<thead>
<tr>
<th>Harrison Ford</th>
<th>Steve Wright</th>
</tr>
</thead>
<tbody>
<tr>
<td>Part of the Star Wars saga</td>
<td>Part of the IPM saga</td>
</tr>
<tr>
<td>Plays a professor/adventurer with a hat and a whip</td>
<td>Is a professor and from Australia</td>
</tr>
</tbody>
</table>