Levenberg-Morrison-Marquardt (LMM) based on probabilistic models

El houcine Bergou

27/11/2015

Outline

Part 1

Problem statement

Problem statement

Problem statement

X_i : the model state at time i,

Problem statement

X_i : the model state at time i,

$$X_0 = \quad x_b + V_0$$

Problem statement

X_i : the model state at time i,

$$X_0 = x_b + V_0$$

- x_b : the prior on X at initial time,

Problem statement

X_i : the model state at time i,

$$X_0 = x_b + V_0$$

- x_b : the prior on X at initial time,
- V_0 : the error on the prior $V_0 \sim N(0, \mathbf{B})$,

Problem statement

X_i : the model state at time i,

$$X_0 = x_b + V_0$$

- x_b : the prior on X at initial time,
- V_0 : the error on the prior $V_0 \sim N(0, \mathbf{B})$,

$$X_i = \mathcal{M}_i(X_{i-1}) + V_i$$

Problem statement

X_i : the model state at time i,

$$X_0 = x_b + V_0$$

- x_b : the prior on X at initial time,
- V_0 : the error on the prior $V_0 \sim N(0, \mathbf{B})$,

$$X_i = \mathcal{M}_i(X_{i-1}) + V_i$$

- $\mathcal{M}_i$: the dynamical model at time i,

Problem statement

X_i : the model state at time i,

$$X_0 = x_b + V_0$$

- x_b : the prior on X at initial time,
- V_0 : the error on the prior $V_0 \sim N(0, \mathbf{B})$,

$$X_i = \mathcal{M}_i(X_{i-1}) + V_i$$

- $\mathcal{M}_i$: the dynamical model at time i,
- V_i : the error on the model $V_i \sim N(0, \mathbf{Q}_i)$,

Problem statement

X_i : the model state at time i,

$$X_0 = x_b + V_0$$

- x_b : the prior on X at initial time,
- V_0 : the error on the prior $V_0 \sim N(0, \mathbf{B})$,

$$X_i = \mathcal{M}_i(X_{i-1}) + V_i$$

- $\mathcal{M}_i$: the dynamical model at time i,
- V_i : the error on the model $V_i \sim N(0, \mathbf{Q}_i)$,

$$d_i = \mathcal{H}_i(X_i) + W_i$$

Problem statement

X_i : the model state at time i,

$$X_0 = x_b + V_0$$

- x_b : the prior on X at initial time,
- V_0 : the error on the prior $V_0 \sim N(0, \mathbf{B})$,

$$X_i = \mathcal{M}_i(X_{i-1}) + V_i$$

- $\mathcal{M}_i$: the dynamical model at time i,
- V_i : the error on the model $V_i \sim N(0, \mathbf{Q}_i)$,

$$d_i = \mathcal{H}_i(X_i) + W_i$$

- d_i : the observation at time i,

Problem statement

X_i : the model state at time i,

$$X_0 = x_b + V_0$$

- x_b : the prior on X at initial time,
- V_0 : the error on the prior $V_0 \sim N(0, \mathbf{B})$,

$$X_i = \mathcal{M}_i(X_{i-1}) + V_i$$

- $\mathcal{M}_i$: the dynamical model at time i,
- V_i : the error on the model $V_i \sim N(0, \mathbf{Q}_i)$,

$$d_i = \mathcal{H}_i(X_i) + W_i$$

- d_i : the observation at time i,
- $\mathcal{H}_i$: the observation operator at time i,

Problem statement

X_i : the model state at time i,

$$X_0 = x_b + V_0$$

- x_b : the prior on X at initial time,
- V_0 : the error on the prior $V_0 \sim N(0, \mathbf{B})$,

$$X_i = \mathcal{M}_i(X_{i-1}) + V_i$$

- $\mathcal{M}_i$: the dynamical model at time i,
- V_i : the error on the model $V_i \sim N(0, \mathbf{Q}_i)$,

$$d_i = \mathcal{H}_i(X_i) + W_i$$

- d_i : the observation at time i,
- $\mathcal{H}_i$: the observation operator at time i,
- W_i : the error on the observation $W_i \sim N(0, \mathbf{R}_i)$,

Problem statement

X_i : the model state at time i,

$$X_0 = x_b + V_0$$

- x_b : the prior on X at initial time,
- V_0 : the error on the prior $V_0 \sim N(0, \mathbf{B})$,

$$X_i = \mathcal{M}_i(X_{i-1}) + V_i$$

- $\mathcal{M}_i$: the dynamical model at time i,
- V_i : the error on the model $V_i \sim N(0, \mathbf{Q}_i)$,

$$d_i = \mathcal{H}_i(X_i) + W_i$$

- d_i : the observation at time i,
- $\mathcal{H}_i$: the observation operator at time i,
- W_i : the error on the observation $W_i \sim N(0, \mathbf{R}_i)$,

The goal is to find the "best estimate" of the model states $X_0, X_1, \ldots$ knowing the observations $d_1, d_2, \ldots$ and the prior x_b.

Maximum a posteriori estimator

Maximum a posteriori estimator

Maximum a posteriori estimator is the minimizer of the following objective function

$$f(x_{0:k}) = \|F(x_{0:k})\|^2 = \| \underbrace{x_0 - x_b}_{\textbf{prior error}} \|^2_{\mathbf{B}^{-1}} + \sum_{i=1}^{k} \| \underbrace{x_i - \mathcal{M}_i(x_{i-1})}_{\textbf{model error}} \|^2_{\mathbf{Q}_i^{-1}}$$

$$+ \sum_{i=1}^{k} \| \underbrace{d_i - \mathcal{H}_i(x_i)}_{\textbf{observation error}} \|^2_{\mathbf{R}_i^{-1}}.$$

Gauss-Newton algorithm

Gauss-Newton algorithm

Gauss-Newton

- Solve the linearized least-squares subproblem,

$$\min_{\delta x^j} \; m_j(x^j + \delta x^j) = \|F(x^j) + J_{F_j}\delta x^j\|^2,$$

Gauss-Newton algorithm

Gauss-Newton

- Solve the linearized least-squares subproblem,

$$\min_{\delta x^j} \; m_j(x^j + \delta x^j) = \|F(x^j) + J_{F_j}\delta x^j\|^2,$$

Gauss-Newton algorithm

Gauss-Newton

- Solve the linearized least-squares subproblem,

$$\min_{\delta x^j} \; m_j(x^j + \delta x^j) = \|F(x^j) + J_{F_j}\delta x^j\|^2,$$

where J_{F_j} is the Jacobian of the function F at x^j.

Gauss-Newton algorithm

Gauss-Newton

- Solve the linearized least-squares subproblem,

$$\min_{\delta x^j} m_j(x^j + \delta x^j) = \|F(x^j) + J_{F_j}\delta x^j\|^2,$$

  where J_{F_j} is the Jacobian of the function F at x^j.

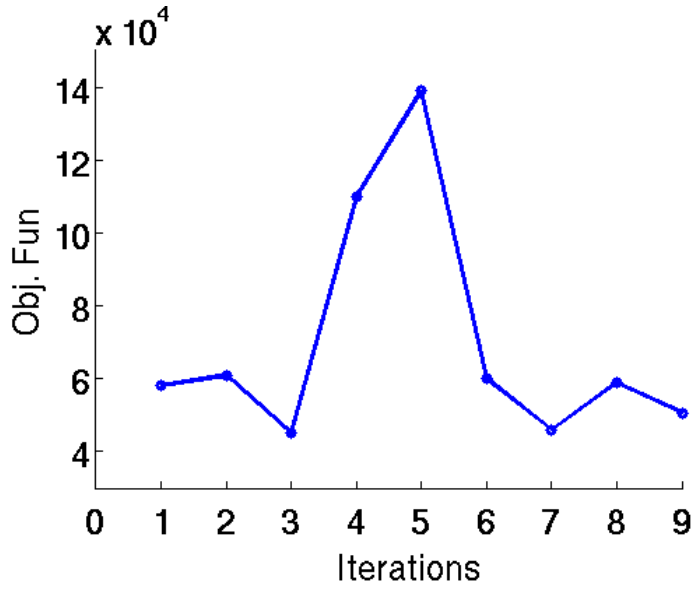
- Update $x^{j+1} = x^j + \delta x^j$.

Gauss-Newton method

- It may fail to converge to a first order stationary point.

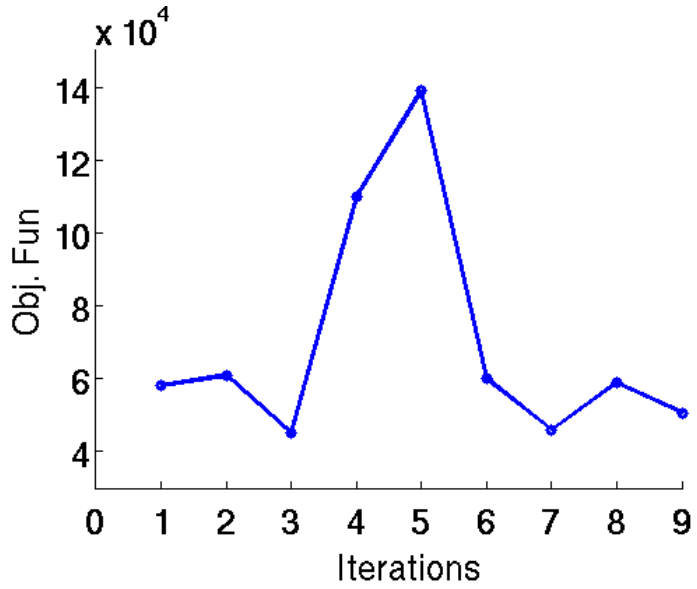
Gauss-Newton method

- It may fail to converge to a first order stationary point.



Gauss-Newton method

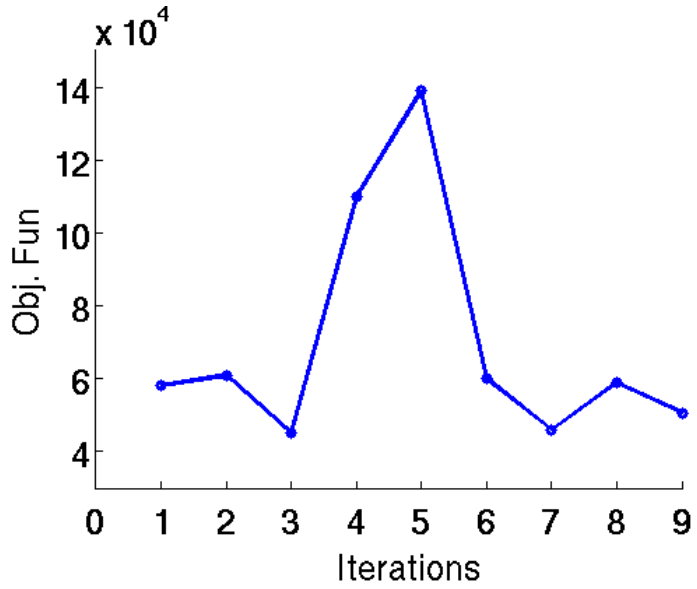
- It may fail to converge to a first order stationary point.



- Derivatives are needed.

Gauss-Newton method

- It may fail to converge to a first order stationary point.



- Derivatives are needed.
- It is difficult to parallelize (speed-up, synchronization,...).

The goal

The objective is to develop an algorithm for large-scale least-squares problems that has the following properties :

The goal

The objective is to develop an algorithm for large-scale least-squares problems that has the following properties :

1. Converges to a first order stationary point, (independently from the starting point)
   $\Rightarrow$ globally convergent algorithm.

The goal

The objective is to develop an algorithm for large-scale least-squares problems that has the following properties :

1. Converges to a first order stationary point, (independently from the starting point)
   $\Rightarrow$ globally convergent algorithm.
2. Derivative-free.

The goal

The objective is to develop an algorithm for large-scale least-squares problems that has the following properties :

1. Converges to a first order stationary point, (independently from the starting point)
   $\Rightarrow$ globally convergent algorithm.
2. Derivative-free.
   $\Rightarrow$ Uses ensemble based methods.

The goal

The objective is to develop an algorithm for large-scale least-squares problems that has the following properties :

1. Converges to a first order stationary point, (independently from the starting point)
   $\Rightarrow$ globally convergent algorithm.
2. Derivative-free.
   $\Rightarrow$ Uses ensemble based methods.
3. Handle the cases where the derivative approximations are random.

Part 2

LMM based on probabilistic models

Levenberg-Morrison-Marquardt method

Levenberg-Morrison-Marquardt method

$\Rightarrow$ The Gauss-Newton subproblem

$$\min_{\delta x^j} m_j(x^j + \delta x^j) = \underbrace{\|F(x^j) + J_{F_j}\delta x^j\|^2}_{\textbf{Gauss-Newton model}}$$

Levenberg-Morrison-Marquardt method

$\Rightarrow$ The Gauss-Newton subproblem is replaced by

$$\min_{\delta x^j} m_j(x^j + \delta x^j) = \underbrace{\|F(x^j) + J_{F_j}\delta x^j\|^2}_{\textbf{Gauss-Newton model}} + \underbrace{\gamma^2\|\delta x^j\|^2}_{\textbf{Penalty term}},$$

Levenberg-Morrison-Marquardt method

$\Rightarrow$ The Gauss-Newton subproblem is replaced by

$$\min_{\delta x^j} m_j(x^j + \delta x^j) = \underbrace{\|F(x^j) + J_{F_j}\delta x^j\|^2}_{\textbf{Gauss-Newton model}} + \underbrace{\gamma^2\|\delta x^j\|^2}_{\textbf{Penalty term}},$$

$$= m(x^j) + g_{f_j}^\top \delta x^j + \delta x^{j\top}\left(J_{F_j}^\top J_{F_j} + \gamma^2 I\right)\delta x^j,$$

Levenberg-Morrison-Marquardt method

$\Rightarrow$ The Gauss-Newton subproblem is replaced by

$$\min_{\delta x^j} m_j(x^j + \delta x^j) = \underbrace{\|F(x^j) + J_{F_j}\delta x^j\|^2}_{\textbf{Gauss-Newton model}} + \underbrace{\gamma^2\|\delta x^j\|^2}_{\textbf{Penalty term}},$$

$$= m(x^j) + g_{f_j}^\top \delta x^j + \delta x^{j\top}\left(J_{F_j}^\top J_{F_j} + \gamma^2 I\right)\delta x^j,$$

- $\gamma = 0 \Rightarrow$ Gauss-Newton step.

Levenberg-Morrison-Marquardt method

$\Rightarrow$ The Gauss-Newton subproblem is replaced by

$$\min_{\delta x^j} m_j(x^j + \delta x^j) = \underbrace{\|F(x^j) + J_{F_j}\delta x^j\|^2}_{\textbf{Gauss-Newton model}} + \underbrace{\gamma^2\|\delta x^j\|^2}_{\textbf{Penalty term}},$$

$$= m(x^j) + g_{f_j}^\top \delta x^j + \delta x^{j\top}\left(J_{F_j}^\top J_{F_j} + \gamma^2 I\right)\delta x^j,$$

- $\gamma = 0 \Rightarrow$ Gauss-Newton step.
- γ large $\Rightarrow \delta x^j \simeq -\frac{g_{f_j}}{\gamma^2}$ (small step).

LMM algorithm

LMM algorithm

Algorithm

- Initialization : $\eta_1 \in (0,1)$, $\gamma_{\min} > 0$, $\lambda > 1$, x^0 and $\gamma^0 \geq \gamma_{\min}$.

LMM algorithm

Algorithm

- Initialization : $\eta_1 \in (0,1)$, $\gamma_{\min} > 0$, $\lambda > 1$, x^0 and $\gamma^0 \geq \gamma_{\min}$.
- For $j = 0, 1, 2, \ldots$

LMM algorithm

Algorithm

- Initialization : $\eta_1 \in (0,1)$, $\gamma_{\min} > 0$, $\lambda > 1$, x^0 and $\gamma^0 \geq \gamma_{\min}$.
- For $j = 0, 1, 2, \ldots$
  1. Approximately solve

$$\min_{\delta x^j} \; m_j(x^j + \delta x^j) = m(x^j) + g_{f_j}^\top \delta x^j + \delta x^{j\top} \left(J_{F_j}^\top J_{F_j} + \gamma^2 I \right) \delta x^j .$$

LMM algorithm

Algorithm

- Initialization : $\eta_1 \in (0,1)$, $\gamma_{\min} > 0$, $\lambda > 1$, x^0 and $\gamma^0 \geq \gamma_{\min}$.
- For $j = 0, 1, 2, \ldots$
  1. Approximately solve

$$\min_{\delta x^j} \; m_j(x^j + \delta x^j) = m(x^j) + g_{f_j}^\top \delta x^j + \delta x^{j\top} \left(J_{F_j}^\top J_{F_j} + \gamma^2 I \right) \delta x^j .$$

  2. Compute $\rho = \frac{f(x^j) - f(x^j + \delta x^j)}{m(x^j) - m(x^j + \delta x^j)}$.

LMM algorithm

Algorithm

- Initialization : $\eta_1 \in (0,1)$, $\gamma_{\min} > 0$, $\lambda > 1$, x^0 and $\gamma^0 \geq \gamma_{\min}$.
- For $j = 0, 1, 2, \ldots$
  1. Approximately solve

$$\min_{\delta x^j} \; m_j(x^j + \delta x^j) = m(x^j) + g_{f_j}^\top \delta x^j + \delta x^{j\top} \left(J_{F_j}^\top J_{F_j} + \gamma^2 I \right) \delta x^j.$$

  2. Compute $\rho = \frac{f(x^j) - f(x^j + \delta x^j)}{m(x^j) - m(x^j + \delta x^j)}$.

     If $\rho \geq \eta_1$ then set $x^{j+1} = x^j + \delta x^j$ and $\gamma^{j+1} = \max\left\{\gamma_{\min}, \frac{\gamma^j}{\lambda}\right\}$.

LMM algorithm

Algorithm

- Initialization : $\eta_1 \in (0,1)$, $\gamma_{\min} > 0$, $\lambda > 1$, x^0 and $\gamma^0 \geq \gamma_{\min}$.
- For $j = 0, 1, 2, \ldots$
  1. Approximately solve

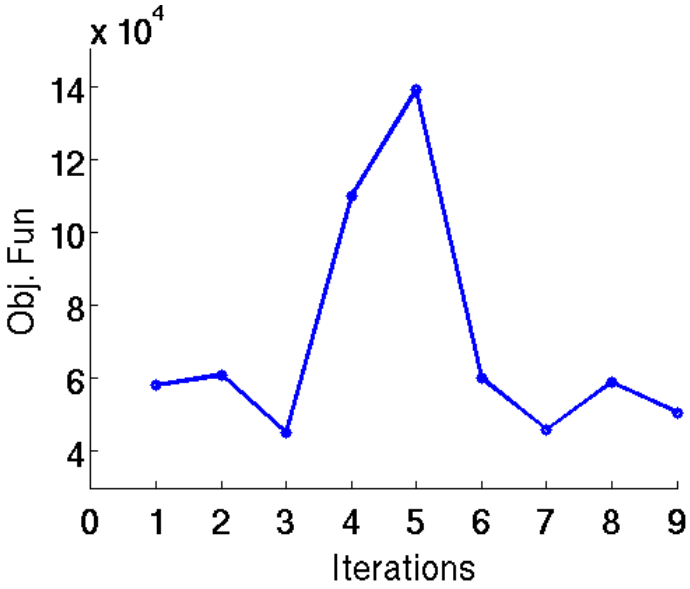
$$\min_{\delta x^j} \; m_j(x^j + \delta x^j) = m(x^j) + g_{f_j}^\top \delta x^j + \delta x^{j\top} \left(J_{F_j}^\top J_{F_j} + \gamma^2 I \right) \delta x^j.$$

  2. Compute $\rho = \frac{f(x^j) - f(x^j + \delta x^j)}{m(x^j) - m(x^j + \delta x^j)}$.

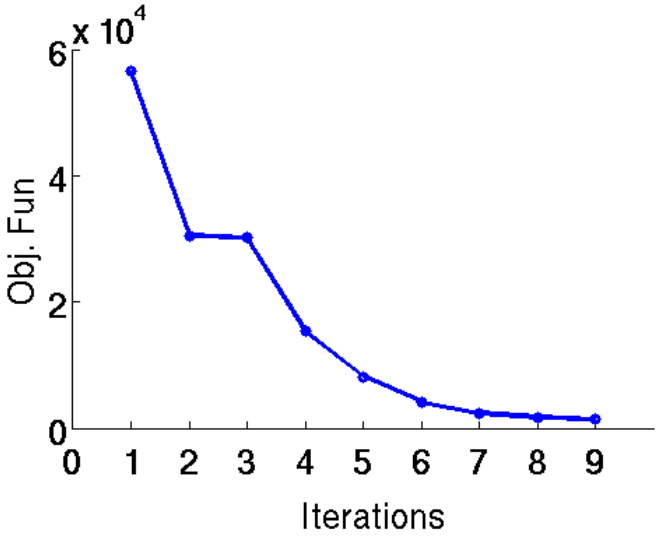
     If $\rho \geq \eta_1$ then set $x^{j+1} = x^j + \delta x^j$ and $\gamma^{j+1} = \max\left\{\gamma_{\min}, \frac{\gamma^j}{\lambda}\right\}$.

     Otherwise, the step is rejected and $\gamma^{j+1} = \lambda \gamma^j$.

Numerical experiments



Gauss-Newton iterates



LMM iterates

Our approach

We are interested with the case where the derivatives are inaccurate

Our approach

We are interested with the case where the derivatives are inaccurate

$$m(x+\delta x) = m(x) + {g_f}^\top \delta x + \delta x^\top (J_F^\top J_F + \gamma^2 I)\delta x,$$

Our approach

We are interested with the case where the derivatives are inaccurate

$$m(x+\delta x) = m(x) + \tilde{g_f}^\top \delta x + \delta x^\top (\tilde{J_F}^\top \tilde{J_F} + \gamma^2 I)\delta x,$$

Our approach

We are interested with the case where the derivatives are inaccurate

$$m(x+\delta x) = m(x) + \tilde{g_f}^\top \delta x + \delta x^\top (\tilde{J_F}^\top \tilde{J_F} + \gamma^2 I)\delta x,$$

- $\tilde{g_f}$ is a random model for the (true) gradient of the function f (g_f),
- $\tilde{J_F}$ is a random model for the (true) Jacobian of the function F (J_F).

Our approach

We are interested with the case where the derivatives are inaccurate

$$m(x+\delta x) = m(x) + \tilde{g_f}^\top \delta x + \delta x^\top (\tilde{J_F}^\top \tilde{J_F} + \gamma^2 I)\delta x,$$

- $\tilde{g_f}$ is a random model for the (true) gradient of the function f (g_f),
- $\tilde{J_F}$ is a random model for the (true) Jacobian of the function F (J_F).

Our approach

We are interested with the case where the derivatives are inaccurate

$$m(x+\delta x) = m(x) + \tilde{g_f}^\top \delta x + \delta x^\top (\tilde{J_F}^\top \tilde{J_F} + \gamma^2 I)\delta x,$$

- $\tilde{g_f}$ is a random model for the (true) gradient of the function f (g_f),
- $\tilde{J_F}$ is a random model for the (true) Jacobian of the function F (J_F).

At each iteration we minimize a realization of the previous model.

Our approach

We are interested with the case where the derivatives are inaccurate

$$m(x+\delta x) = m(x) + \tilde{g_f}^\top \delta x + \delta x^\top (\tilde{J_F}^\top \tilde{J_F} + \gamma^2 I)\delta x,$$

- $\tilde{g_f}$ is a random model for the (true) gradient of the function f (g_f),
- $\tilde{J_F}$ is a random model for the (true) Jacobian of the function F (J_F).

At each iteration we minimize a realization of the previous model.

Our approach

We are interested with the case where the derivatives are inaccurate

$$m(x+\delta x) = m(x) + \tilde{g_f}^\top \delta x + \delta x^\top (\tilde{J_F}^\top \tilde{J_F} + \gamma^2 I)\delta x,$$

- $\tilde{g_f}$ is a random model for the (true) gradient of the function f (g_f),
- $\tilde{J_F}$ is a random model for the (true) Jacobian of the function F (J_F).

At each iteration we minimize a realization of the previous model.

The choice of a realization of a random model at each iteration allows the occasional use of bad models,

Our approach

We are interested with the case where the derivatives are inaccurate

$$m(x+\delta x) = m(x) + \tilde{g}_f{}^\top \delta x + \delta x^\top (\tilde{J}_F^\top \tilde{J}_F + \gamma^2 I)\delta x,$$

- $\tilde{g}_f$ is a random model for the (true) gradient of the function f (g_f),
- $\tilde{J}_F$ is a random model for the (true) Jacobian of the function F (J_F).

At each iteration we minimize a realization of the previous model.

The choice of a realization of a random model at each iteration allows the occasional use of bad models,

$\Rightarrow$ what do we need from the model to ensure convergence ?

Assumption (probabilistic fully linearity)

Assumption (probabilistic fully linearity)

We need some control on the model error gradient : a Taylor like behavior of first-order models [Vicente et al. 2013].

Assumption

Given $\alpha \in (0, 2]$ *and* $\kappa_{eg} > 0$

Assumption (probabilistic fully linearity)

We need some control on the model error gradient : a Taylor like behavior of first-order models [Vicente et al. 2013].

Assumption

Given $\alpha \in (0, 2]$ *and* $\kappa_{eg} > 0$

$$\|\tilde{g}_{f_j} - g_{f_j}\| \leq \frac{\kappa_{eg}}{\gamma_j^{\alpha}}$$

Assumption (probabilistic fully linearity)

We need some control on the model error gradient : a Taylor like behavior of first-order models [Vicente et al. 2013].

Assumption

Given $\alpha \in (0, 2]$ *and* $\kappa_{eg} > 0$

$$\mathbb{P}\left[\|\tilde{g}_{f_j} - g_{f_j}\| \leq \frac{\kappa_{eg}}{\gamma_j^{\alpha}} \mid \textit{History} \right] \geq p_j$$

Assumption (probabilistic fully linearity)

We need some control on the model error gradient : a Taylor like behavior of first-order models [Vicente et al. 2013].

Assumption

Given $\alpha \in (0, 2]$ *and* $\kappa_{eg} > 0$, $\exists\ p_{\min} \in (0, 1]$

$$\mathbb{P}\left[\|\tilde{g}_{f_j} - g_{f_j}\| \leq \frac{\kappa_{eg}}{\gamma_j^{\alpha}} \mid \textit{History}\right] \geq p_j \quad \textit{and } p_j \geq p_{\min} > 0.$$

The proposed modifications

The proposed modifications

Algorithm [Bergou, Gratton and Vicente 2014]

The proposed modifications

Algorithm [Bergou, Gratton and Vicente 2014]

- Initialization : $\eta_1 \in (0,1)$, $\eta_2 \in (0,1)$, $\gamma_{\min} > 0$, $\lambda > 1$, x^0 and $\gamma_0 \geq \gamma_{\min}$.

The proposed modifications

Algorithm [Bergou, Gratton and Vicente 2014]

- Initialization : $\eta_1 \in (0,1)$, $\eta_2 \in (0,1)$, $\gamma_{\min} > 0$, $\lambda > 1$, x^0 and $\gamma_0 \geq \gamma_{\min}$.
- For $j = 0, 1, 2, \ldots$

The proposed modifications

Algorithm [Bergou, Gratton and Vicente 2014]

- Initialization : $\eta_1 \in (0,1)$, $\eta_2 \in (0,1)$, $\gamma_{\min} > 0$, $\lambda > 1$, x^0 and $\gamma_0 \geq \gamma_{\min}$.
- For $j = 0, 1, 2, \ldots$
  1. Approximately solve a realization of the subproblem.

The proposed modifications

Algorithm [Bergou, Gratton and Vicente 2014]

- Initialization : $\eta_1 \in (0,1)$, $\eta_2 \in (0,1)$, $\gamma_{\min} > 0$, $\lambda > 1$, x^0 and $\gamma_0 \geq \gamma_{\min}$.
- For $j = 0, 1, 2, \ldots$
  1. Approximately solve a realization of the subproblem.
  2. Compute $\rho = \frac{f(x^j) - f(x^j + \delta x^j)}{m(x^j) - m(x^j + \delta x^j)}$.

     If $\rho \geq \eta_1$ then set $x^{j+1} = x^j + \delta x^j$

The proposed modifications

Algorithm [Bergou, Gratton and Vicente 2014]

- Initialization : $\eta_1 \in (0,1)$, $\eta_2 \in (0,1)$, $\gamma_{\min} > 0$, $\lambda > 1$, x^0 and $\gamma_0 \geq \gamma_{\min}$.
- For $j = 0, 1, 2, \ldots$
  1. Approximately solve a realization of the subproblem.
  2. Compute $\rho = \frac{f(x^j) - f(x^j + \delta x^j)}{m(x^j) - m(x^j + \delta x^j)}$.

     If $\rho \geq \eta_1$ then set $x^{j+1} = x^j + \delta x^j$

The proposed modifications

Algorithm [Bergou, Gratton and Vicente 2014]

- Initialization : $\eta_1 \in (0,1)$, $\eta_2 \in (0,1)$, $\gamma_{\min} > 0$, $\lambda > 1$, x^0 and $\gamma_0 \geq \gamma_{\min}$.
- For $j = 0, 1, 2, \ldots$
  1. Approximately solve a realization of the subproblem.
  2. Compute $\rho = \frac{f(x^j) - f(x^j + \delta x^j)}{m(x^j) - m(x^j + \delta x^j)}$.

     If $\rho \geq \eta_1$ then set $x^{j+1} = x^j + \delta x^j$

$$\gamma_{j+1} = \begin{cases} \lambda \gamma_j & \text{if } \|g_{m_j}\| < \eta_2/\gamma_j, \\ \max\left\{ \frac{\gamma_j}{\lambda^{\frac{1-p_j}{p_j}}}, \gamma_{\min} \right\} & \text{if } \|g_{m_j}\| \geq \eta_2/\gamma_j. \end{cases}$$

The proposed modifications

Algorithm [Bergou, Gratton and Vicente 2014]

- Initialization : $\eta_1 \in (0,1)$, $\eta_2 \in (0,1)$, $\gamma_{\min} > 0$, $\lambda > 1$, x^0 and $\gamma_0 \geq \gamma_{\min}$.
- For $j = 0, 1, 2, \ldots$
  1. Approximately solve a realization of the subproblem.
  2. Compute $\rho = \frac{f(x^j) - f(x^j + \delta x^j)}{m(x^j) - m(x^j + \delta x^j)}$.

     If $\rho \geq \eta_1$ then set $x^{j+1} = x^j + \delta x^j$

$$\gamma_{j+1} = \begin{cases} \lambda \gamma_j & \text{if } \|g_{m_j}\| < \eta_2/\gamma_j, \\ \max\left\{ \lambda^{\frac{\gamma_j}{\frac{1-p_j}{p_j}}}, \gamma_{\min} \right\} & \text{if } \|g_{m_j}\| \geq \eta_2/\gamma_j. \end{cases}$$

Otherwise, the step is rejected and $\gamma_{j+1} = \lambda \gamma_j$.

Convergence theorem

Convergence theorem

Theorem [Bergou, Gratton and Vicente 2014]

Let $\{X^j\}$ be a sequence of random iterates generated by our Algorithm where the probability p_j satisfies the probabilistic fully linearity.

Convergence theorem

Theorem [Bergou, Gratton and Vicente 2014]

Let $\{X^j\}$ be a sequence of random iterates generated by our Algorithm where the probability p_j satisfies the probabilistic fully linearity.
Under appropriate assumptions

Convergence theorem

Theorem [Bergou, Gratton and Vicente 2014]

Let $\{X^j\}$ be a sequence of random iterates generated by our Algorithm where the probability p_j satisfies the probabilistic fully linearity. Under appropriate assumptions then almost surely,

$$\liminf_{j \to \infty} \|\nabla f(X^j)\| = 0$$

(global convergence property).

Convergence theorem

Theorem [Bergou, Gratton and Vicente 2014]

Let $\{X^j\}$ be a sequence of random iterates generated by our Algorithm where the probability p_j satisfies the probabilistic fully linearity. Under appropriate assumptions then almost surely,

$$\liminf_{j \to \infty} \|\nabla f(X^j)\| = 0$$

(global convergence property).

Numerical application

Numerical application

The model gradient is a Gaussian perturbation of the exact one.

Numerical application

The model gradient is a Gaussian perturbation of the exact one.

$$\tilde{g}_{f_j} = g_{f_j} + \varepsilon_j \text{ where } \varepsilon_j \sim N(0, \Sigma_j).$$

Numerical application

The model gradient is a Gaussian perturbation of the exact one.

$$\tilde{g}_{f_j} = g_{f_j} + \varepsilon_j \text{ where } \varepsilon_j \sim N(0, \Sigma_j).$$

Lemma

$$\mathbb{P}\left[\| \underbrace{\tilde{g}_{f_j} - g_{f_j}}_{\varepsilon_j} \| \leq \frac{\kappa_{eg}}{\gamma_j^{\alpha}} | \ \mathit{History} \right]$$

Numerical application

The model gradient is a Gaussian perturbation of the exact one.

$$\tilde{g}_{f_j} = g_{f_j} + \varepsilon_j \text{ where } \varepsilon_j \sim N(0, \Sigma_j).$$

Lemma

$$\mathbb{P}\left[\| \underbrace{\tilde{g}_{f_j} - g_{f_j}}_{\varepsilon_j} \| \leq \frac{\kappa_{eg}}{\gamma_j^{\alpha}} | \ History \right] \geq \tilde{p}_j = CDF^{-1}_{\chi 2(n)} \left(\left(\frac{\kappa_{eg}}{\sqrt{\|\Sigma_j\|} \min\{\lambda^j \gamma_0, \gamma_{\max}\}^{\alpha}} \right)^2 \right)$$

where $CDF_{\chi 2(n)}$ is the cumulative density function of a chi-squared distribution with n degrees of freedom.

Numerical application

The model gradient is a Gaussian perturbation of the exact one.

$$\tilde{g}_{f_j} = g_{f_j} + \varepsilon_j \text{ where } \varepsilon_j \sim N(0, \Sigma_j).$$

Lemma

$$\mathbb{P}\left[\| \underbrace{\tilde{g}_{f_j} - g_{f_j}}_{\varepsilon_j} \| \leq \frac{\kappa_{eg}}{\gamma_j^{\alpha}} | \ History \right] \geq \tilde{p}_j = CDF^{-1}_{\chi 2(n)} \left(\left(\frac{\kappa_{eg}}{\sqrt{\|\Sigma_j\|} \min\{\lambda^j \gamma_0, \gamma_{\max}\}^{\alpha}} \right)^2 \right)$$

$$\geq p_{\min} = CDF^{-1}_{\chi 2(n)} \left(\left(\frac{\kappa_{eg}}{\sup_j\{\sqrt{\|\Sigma_j\|}\} \gamma_{\max}^{\alpha}} \right)^2 \right),$$

where $CDF_{\chi 2(n)}$ is the cumulative density function of a chi-squared distribution with n degrees of freedom.

Numerical illustration

Numerical illustration

We tested our approach using the following toy problem :

Numerical illustration

We tested our approach using the following toy problem :

$$\min_{x,y} f(x,y) = \left(\|x-1\|^2 + 100\|y-x^2\|^2\right),$$

$(x^*, y^*) = (1,1),\ f(x^*, y^*) = 0,$

$\varepsilon_j \sim N(0, 10\mathbf{I}_2).$

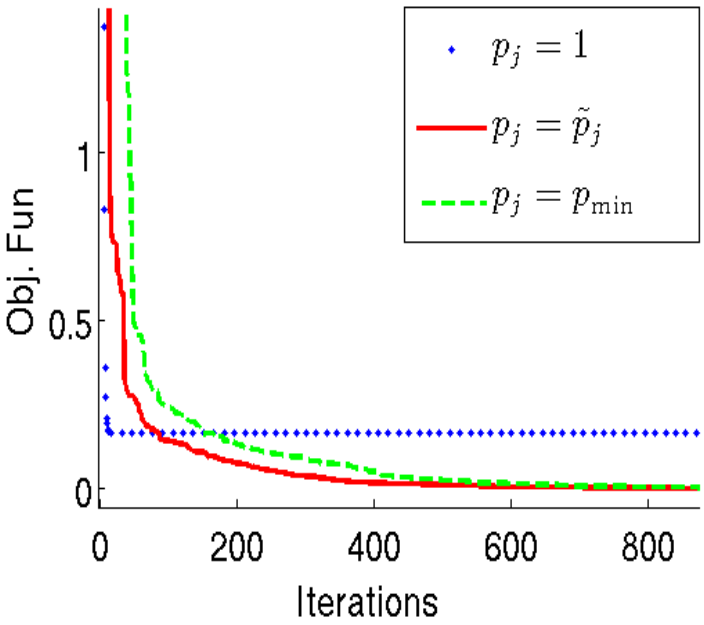
Numerical illustration

We tested our approach using the following toy problem :

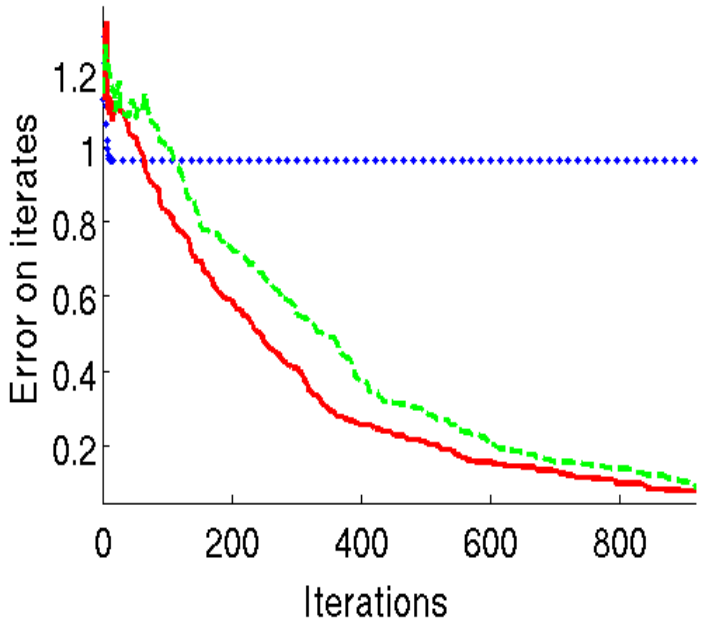
$$\min_{x,y} f(x,y) = \left(\|x-1\|^2 + 100\|y-x^2\|^2\right),$$

$(x^*, y^*) = (1,1),\ f(x^*, y^*) = 0,$

$\varepsilon_j \sim N(0, 10\mathbf{I}_2)$.



Average of function values



Average of absolute error of iterates

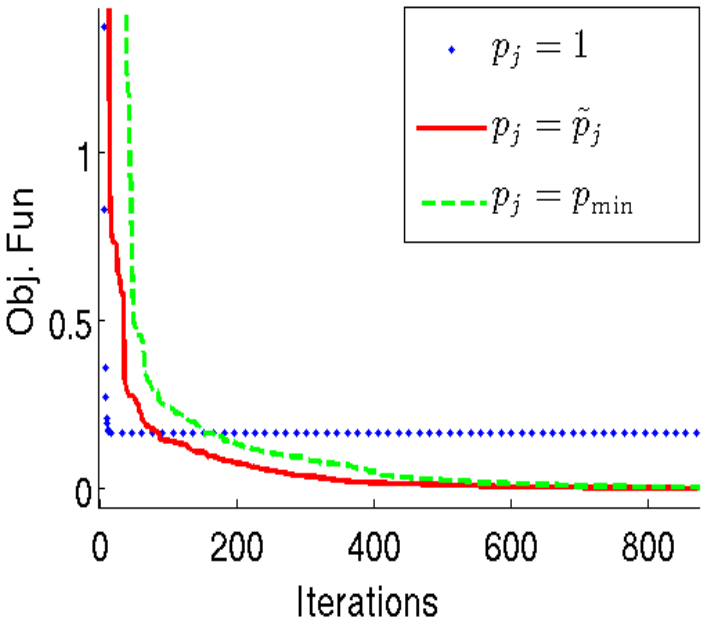
Numerical illustration

We tested our approach using the following toy problem :

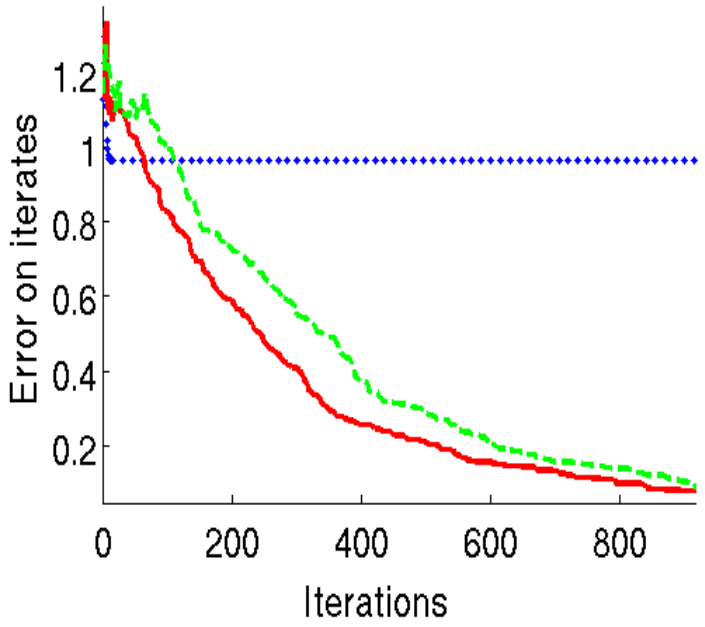
$$\min_{x,y} f(x,y) = \left(\|x-1\|^2 + 100\|y-x^2\|^2\right),$$

$(x^*, y^*) = (1,1),\ f(x^*, y^*) = 0,$

$\varepsilon_j \sim N(0, 10\mathbf{I}_2)$.



Average of function values



Average of absolute error of iterates

Part 3

Conclusions

Conclusions

Conclusions

Contributions

Conclusions

Contributions

1. We have derived a variant of the Levenberg-Morrison-Marquardt method to handle the scenarios where the linearized subproblem is solved inexactly and/or the gradient model is accurate only within a certain probability.

Conclusions

Contributions

1. We have derived a variant of the Levenberg-Morrison-Marquardt method to handle the scenarios where the linearized subproblem is solved inexactly and/or the gradient model is accurate only within a certain probability.
2. Under appropriate assumptions we have shown that our approach is globally convergent.

Conclusions

Contributions

1. We have derived a variant of the Levenberg-Morrison-Marquardt method to handle the scenarios where the linearized subproblem is solved inexactly and/or the gradient model is accurate only within a certain probability.
2. Under appropriate assumptions we have shown that our approach is globally convergent.
3. We have given some practical situations where our approach can be used.

Thank you for your attention.