



Nikhef

# Gandalf CPU-time

2019-10-31

Bouke Jung ([bjung@nikhef.nl](mailto:bjung@nikhef.nl))  
KM3NeT collaboration meeting



# The goal



# Gandalf recipe

- Essential tool for track reconstruction
  - ML-fit *around* results from prefit
  - Multi-dimensional arrival-t PDF's available in Jpp  
(for direct/scattered light from muons/EM showers/hadronic showers/...)

$$\mathcal{L} = \prod_{\text{hit PMTs}} \left[ \frac{\delta P}{\delta t} (R_i, \theta_i, \phi_i, \Delta t) \right]$$

Ó Fearraigh, B. (2019)

# Gandalf recipe

- Essential tool for track reconstruction
  - ML-fit *around* results from prefit
  - Multi-dimensional arrival-t PDF's available in Jpp  
(for direct/scattered light from muons/EM showers/hadronic showers/...)
- How does it work?

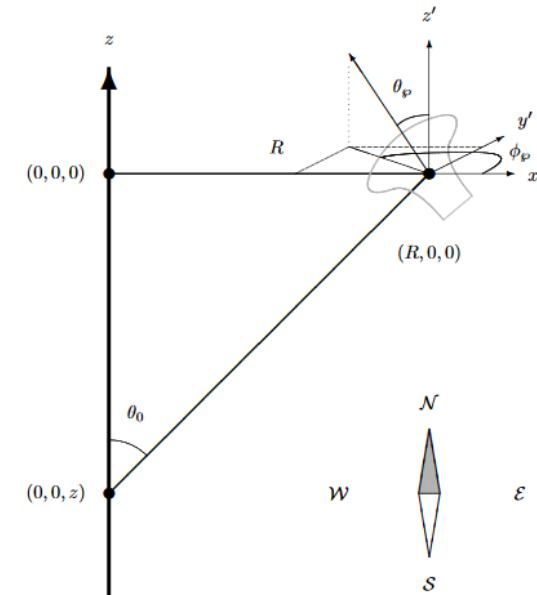
Given initial track coördinates  $\vec{\beta}_0$  from prefit; do the following:

- Evaluate  $\mathcal{L}$  at  $\vec{\beta}$
- Evaluate  $\vec{\nabla}\mathcal{L}$  at  $\vec{\beta}$
- Perform Levenberg-Marquardt update
- Check for convergence

Iterate and stop when sufficient convergence reached

$$\mathcal{L} = \prod_{\text{hit PMTs}} \left[ \frac{\delta P}{\delta t} (R_i, \theta_i, \phi_i, \Delta t) \right]$$

Ó Fearraigh, B. (2019)



de Jong, M. (2019)

# Gandalf recipe

- Essential tool for track reconstruction
  - ML-fit *around* results from prefit
  - Multi-dimensional arrival-t PDF's available in Jpp  
(for direct/scattered light from muons/EM showers/hadronic showers/...)
- How does it work?

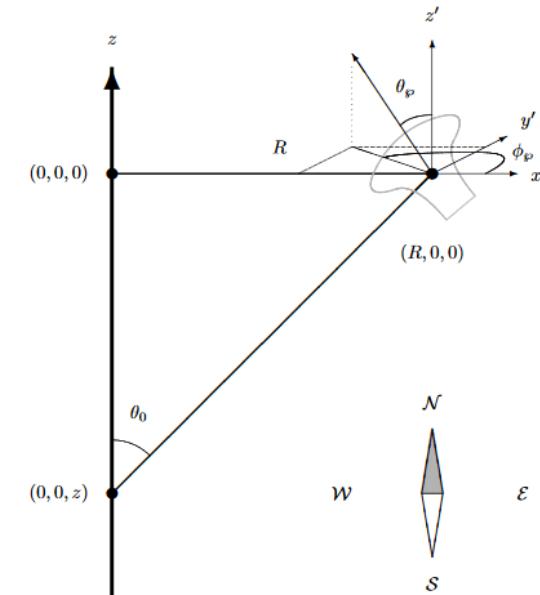
Given initial track coördinates  $\vec{\beta}_0$  from prefit; do the following:

- Evaluate  $\mathcal{L}$  at  $\vec{\beta}$
- Evaluate  $\vec{\nabla}\mathcal{L}$  at  $\vec{\beta}$
- Perform Levenberg-Marquardt update
- Check for convergence

Iterate and stop when sufficient convergence reached

$$\mathcal{L} = \prod_{\text{hit PMTs}} \left[ \frac{\delta P}{\delta t} (R_i, \theta_i, \phi_i, \Delta t) \right]$$

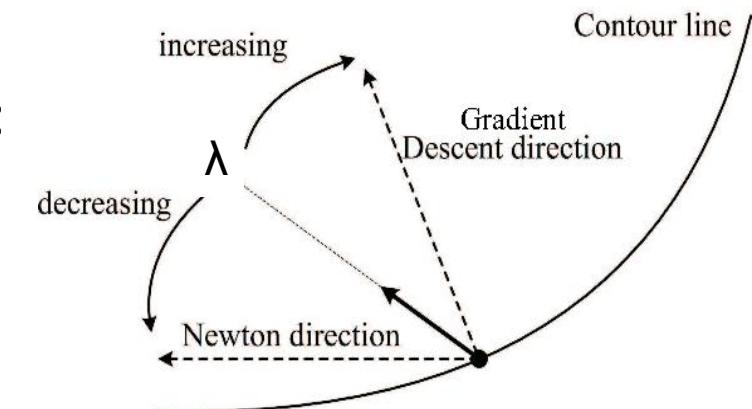
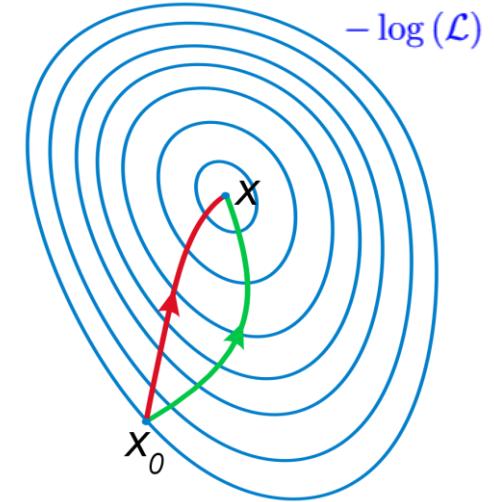
Ó Fearraigh, B. (2019)



de Jong, M. (2019)

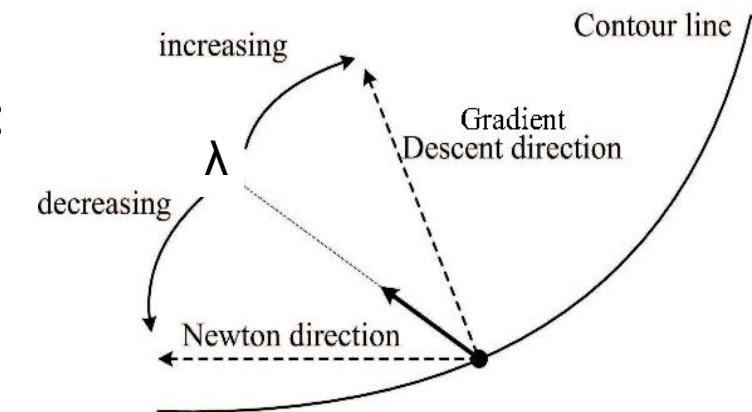
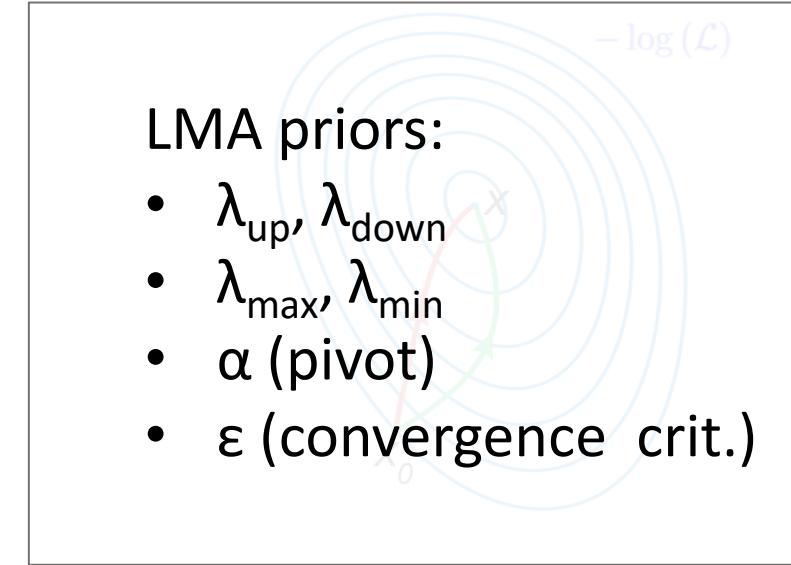
# Levenberg-Marquardt Algorithm

- Optimization algorithm based on:
  - Gradient descent method**
    - Parameter update in direction of steepest-descent
  - Newton optimization**
    - Parameter update by minimizing local 2<sup>nd</sup> order Taylor-Exp.
- LMA combines **the best of both worlds!**
  - Robustness of gradient descent
  - Quick convergence of Newton optimization
- Control parameter ( $\lambda$ ) decides in which direction to go:
  - Small  $\lambda \rightarrow$  Newton step
  - Large  $\lambda \rightarrow$  Gradient step



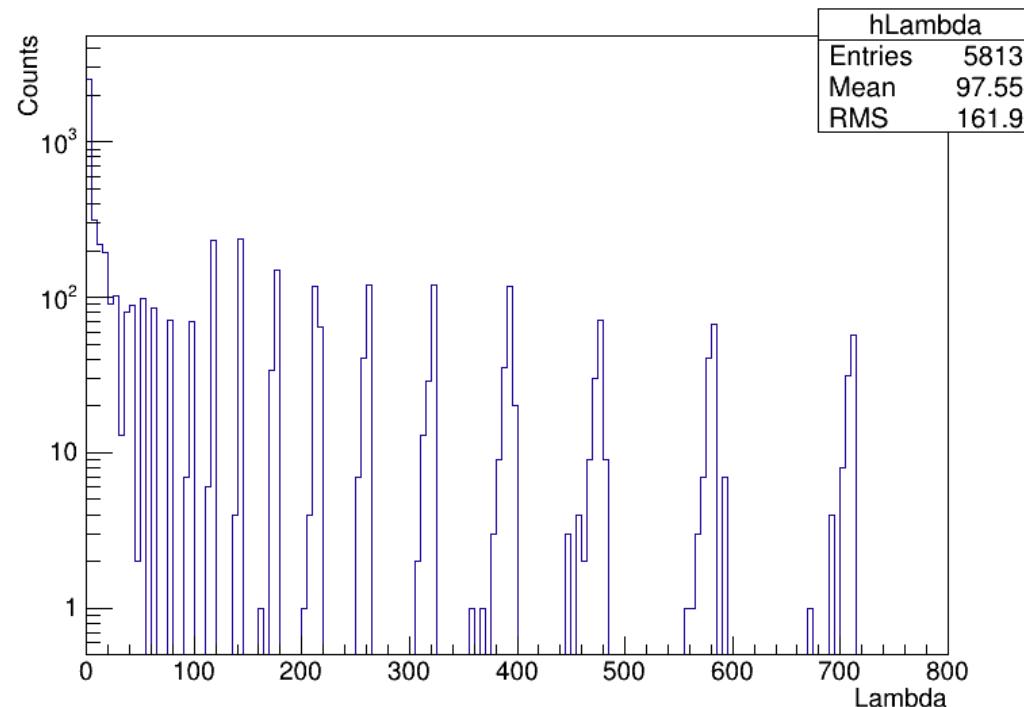
# Levenberg-Marquardt Algorithm

- Optimization algorithm based on:
  1. **Gradient descent method**
    - Parameter update in direction of steepest-descent
  2. **Newton optimization**
    - Parameter update by minimizing local 2<sup>nd</sup> order Taylor-Exp.
- LMA combines **the best of both worlds!**
  - Robustness of gradient descent
  - Quick convergence of Newton optimization
- Control parameter ( $\lambda$ ) decides in which direction to go:
  - Small  $\lambda \rightarrow$  Newton step
  - Large  $\lambda \rightarrow$  Gradient step
- If step (un)succesful, (increase) decrease  $\lambda$



# The control parameter

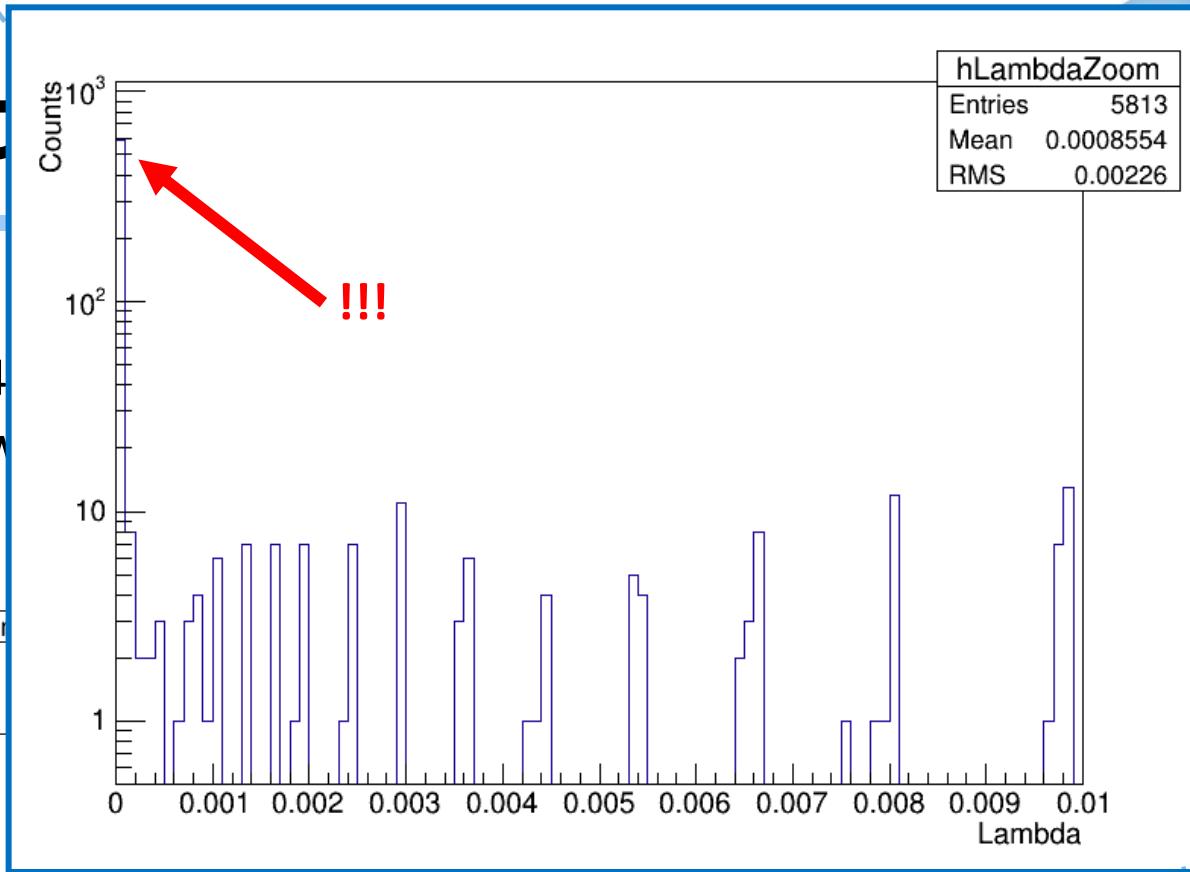
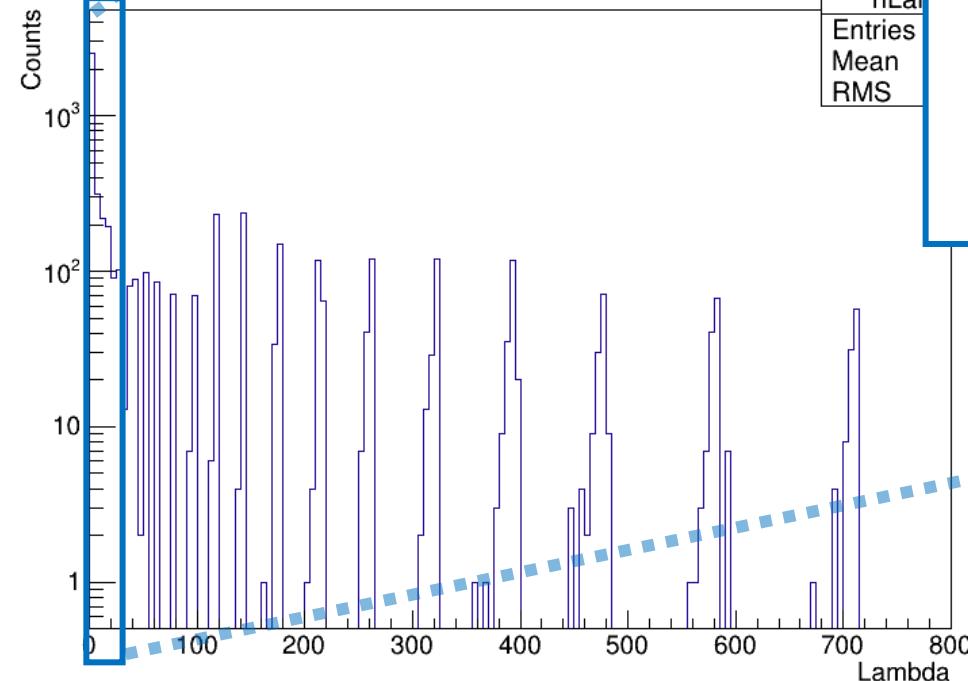
- Ran JGandalf on 4DU real data, run 6019
  - $N_{\text{prefits}} = 12$
  - $N_{\text{events}} = 500$
  - $T$  in  $[-50, +450]$  ns
  - 50 m roadwidth
- Inspected LMA control parameter



- Number of iterations quite large
  - ~25% of fits have  $\geq 50$  iterations

# The control parameter

- Ran JGandalf on 4DU real data, run
  - $N_{\text{prefits}} = 12$
  - $N_{\text{events}} = 500$
  - $T$  in  $[-50, +40]$
  - 50 m roadways
- Inspected fit control parameter



- Number of iterations quite large
  - ~25% of fits have  $\geq 50$  iterations
- LMA control parameter (lambda) **unbounded towards lower end**

# The control parameter

- Ran JGandalf on 4DU real data, run 6019
  - $N_{\text{prefits}} = 12$
  - $N_{\text{events}} = 500$
  - T in [-50,+450] ns
  - 50 m roadwidth
- Inspected fit control parameter

```

150     if (chi2 < chi2_old) {
151
152         if (numberOfIterations != 0) {
153
154             if (fabs(chi2_old - chi2) < EPSILON*fabs(chi2_old)) {
155                 return chi2;
156             }
157
158             if (lambda > LAMBDA_MIN) {
159                 lambda /= LAMBDA_DOWN;
160             }
161
162         }
163
164         chi2_old = chi2;
165         previous = value;
166

```

JGandalf.hh

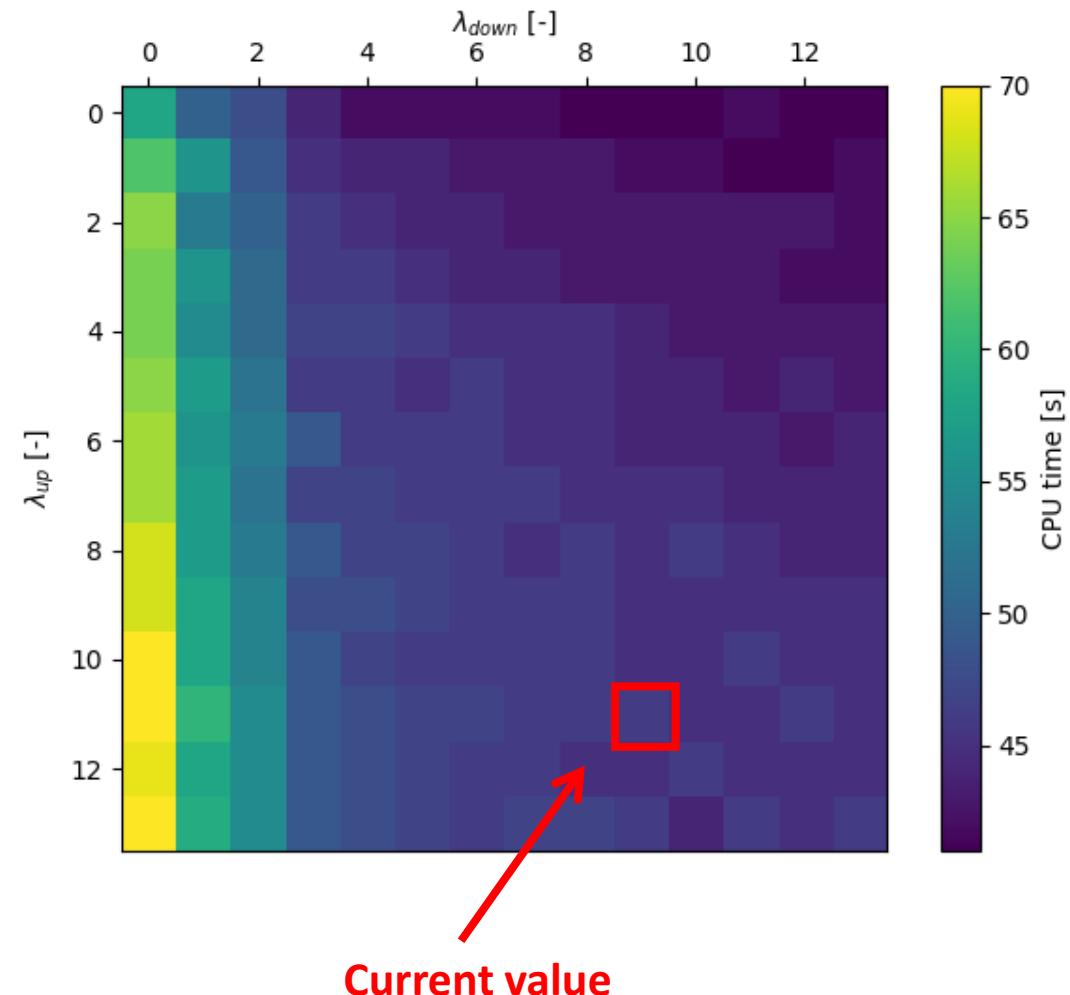
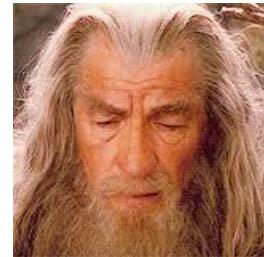
**25% speed-up**

Just by including this lower bound on  $\lambda$

- When combined with JSimplex → JGandalf, **2x speed-up**

# LMA priors

- Ran JGandalf on 4DU real data, run 601
  - $N_{\text{prefits}} = 12$
  - $N_{\text{events}} = 500$
  - T in [-50,+450]
  - 50 m roadwidth
- No strong CPU-t influence seen for  $\lambda_{\text{up}}$ ,  $\lambda_{\text{down}}$ ,  $\lambda_{\text{max}}$ ,  $\lambda_{\text{min}}$ , or  $\alpha$



# LMA priors

- Ran JGandalf on 4DU real data, run 6019
  - $N_{\text{prefits}} = 12$
  - $N_{\text{events}} = 500$
  - $T$  in [-50,+450] ns
  - 50 m roadwidth
- Some improvement possible for  $\varepsilon$ :

$\varepsilon$ [-]	$10^{-2}$	$10^{-3}$	$10^{-4}$	$10^{-5}$	$10^{-6}$
CPU time [s]	34	42	49	55	57

# LMA priors

- Ran JGandalf on 4DU real data, run 6019
  - $N_{\text{prefits}} = 12$
  - $N_{\text{events}} = 500$
  - $T$  in  $[-50, +450]$  ns
  - 50 m roadwidth
- Some improvement possible for  $\varepsilon$ :

$\varepsilon$ [-]	$10^{-2}$	$10^{-3}$	$10^{-4}$	$10^{-5}$	$10^{-6}$
CPU time [s]	34	42	49	55	57

Brían:

Setting to  $10^{-2}$  worsens median ang. res. by  $0.2^\circ$

Setting to  $10^{-3}$  worsens median ang. res. by  $0.02^\circ$

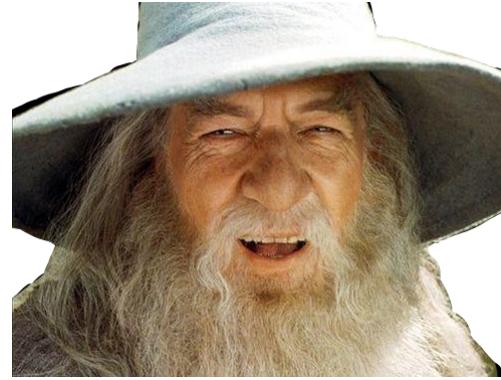
# Geodesic acceleration



Improvements to the Levenberg-Marquardt algorithm for nonlinear least-squares minimization

Mark K. Transtrum<sup>a</sup>, James P. Sethna<sup>a</sup>

<sup>a</sup>*Laboratory of Atomic and Solid State Physics, Cornell University, Ithaca, New York 14853, USA*



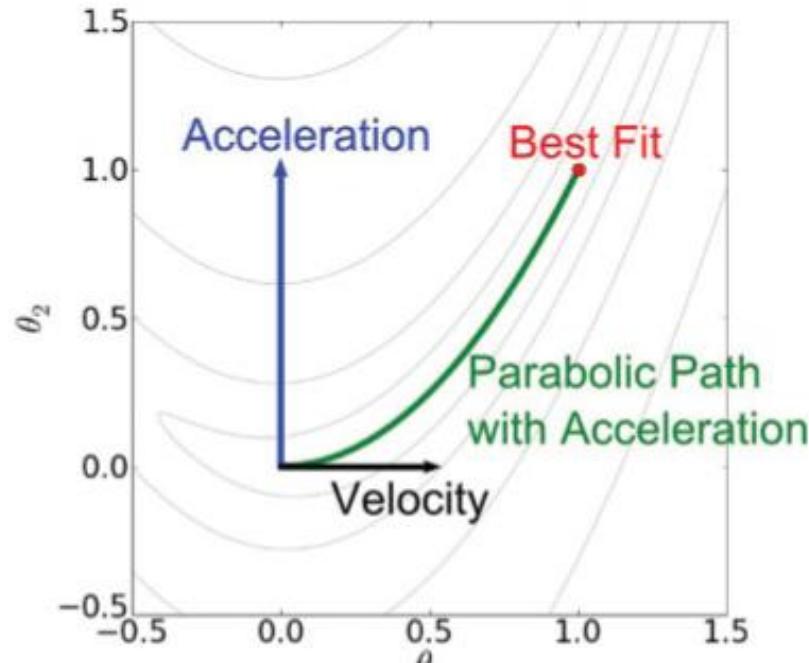
---

## Abstract

When minimizing a nonlinear least-squares function, the Levenberg-Marquardt algorithm can suffer from a slow convergence, particularly when it must navigate a narrow canyon en route to a best fit. On the other hand, when the least-squares function is very flat, the algorithm may easily become lost in parameter space. We introduce several improvements to the Levenberg-Marquardt algorithm in order to improve both its convergence speed and robustness to initial parameter guesses. We update the usual

# Geodesic acceleration

- Newton-updates (= geodesic velocity) give parameter update in optimal direction when navigating canyons
- But can lead to small **up-hill movements**
  - The direction of the optimal path may change quickly...
- Include 2<sup>nd</sup> order corrections in Taylor exp.  
→ Enhanced parameter update (Geodesic Acc.)

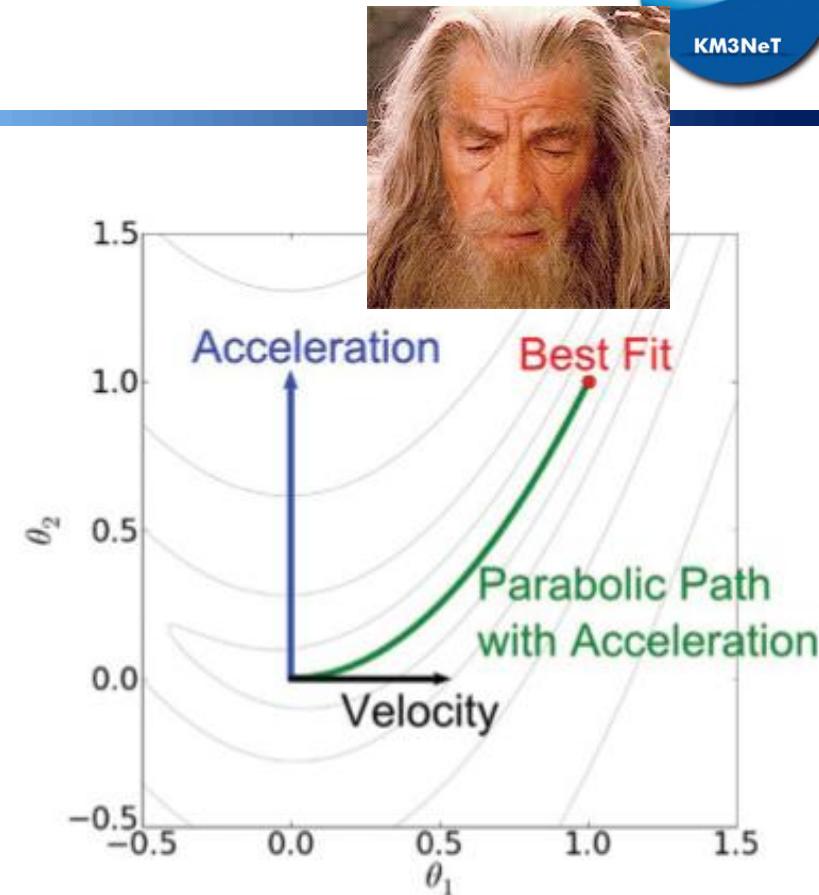


Transtrum et al. (2011) - *PhysRevE.83.036701*

# Geodesic acceleration



- Newton-updates (= geodesic velocity) give parameter update in optimal direction when navigating canyons
- But can lead to small **up-hill movements**
  - The direction of the optimal path may change quickly...
- Include 2<sup>nd</sup> order corrections in Taylor exp.
  - Enhanced parameter update (Geodesic Acc.)
- Unfortunately, no speed increase so far...



Transtrum et al. (2011) - *PhysRevE.83.036701*

# In summary

- Jsimplex helps with CPU time!
- Set a lower bound on lambda ( $\geq 10^{-4}$ )
- Combined: **x2 speed-up!**
- No CPU-t influence seen for the LMA-priors,
- Except **for  $\epsilon$** . Setting to  $10^{-3}$  (from  $10^{-4}$ ) yields:
  - **10% speed-up**
  - At the cost of ang. res. **0.02° worsening of median ang. res.**
- Implementation of Geod.Acc. attempted, but no speed-up found

# Some LMA Literature

- I. [http://www.ananth.in/Notes\\_files/lmtut.pdf](http://www.ananth.in/Notes_files/lmtut.pdf)
- II. <https://arxiv.org/pdf/1207.4999.pdf>
- III. <https://arxiv.org/pdf/1201.5885.pdf>
- IV. <https://journals.aps.org/prl/pdf/10.1103/PhysRevLett.104.060201>
- V. <https://journals-aps-org.proxy.uba.uva.nl:2443/pre/pdf/10.1103/PhysRevE.83.036701>
- VI. <https://www.gnu.org/software/gsl/doc/html/nls.html>  
(GNU Geodesic Acc. software implementation)