Linking models with data - The Nelder-Mead method

Patrick Ludl

Faculty of Physics, University of Vienna

March 29, 2012

FWF Project P 24161-N16

Outline of the talk

- Linking models with data — A figure of merit function
- Linking models with data — The Nelder-Mead method
- Towards application — The pinning term method
- Towards application — An illustrative example

Linking models with data - A figure of merit function

A figure of merit function

Common situation in model building:

- On the theoretical side:
  Model with n (real) free parameters $x_\alpha \quad (\alpha = 1, ..., n)$.
- On the experimental side:
  Experimental results for q observables $\mathcal{O}_i$

$$\mathcal{O}_i = \bar{\mathcal{O}}_i \pm \sigma_i \quad (i = 1, ..., q).$$

$\longrightarrow$ Most important question:

How well can the exp. results $\bar{\mathcal{O}}_i$ be accommodated within the model?

$\longrightarrow$ We need some measure how well the predictions of the model agree with the experiment.

$\Rightarrow$ figure of merit function χ^2.

A figure of merit function

Experiment: q observables $\mathcal{O}_i = \bar{\mathcal{O}}_i \pm \sigma_i \quad (i = 1, ..., q)$.

Model: n free parameters $x_\alpha \Rightarrow$ Predictions $P_i(\vec{x})$ for the observables $\mathcal{O}_i$.

$$\chi^2(\vec{x}) := \sum_{i=1}^{q} \left(\frac{P_i(\vec{x}) - \bar{\mathcal{O}}_i}{\sigma_i} \right)^2 .$$

Properties of χ^2:

- $\chi^2(\vec{x}) \geq 0$,
- **Global minimum**: $\chi^2(\vec{x}) = 0$ if $P_i(\vec{x}) = \bar{\mathcal{O}}_i$.

The smaller the **global minimum of** χ^2, the better the agreement between model predictions and observations.

A figure of merit function

Problem: In many cases the local minima of χ^2 are not known analytically.

$\Rightarrow$ We need **numerical methods.**

Using numerical methods we need to take into account several issues.

- Probably very large number of local minima.
  $\Rightarrow$ Algorithm must avoid to get stuck in a local minimum.
- "Landscape" of the χ^2-function may show a complicated topology.
  $\Rightarrow$ Algorithm should adapt to this topology.
- The functions $P_i(\vec{x})$ may be complicated.
  $\Rightarrow$ Large computational effort needed.
- Finite accuracy of numerical methods.
- Convergence properties of the algorithm.

Linking models with data - The Nelder-Mead method

The Nelder-Mead method (NMM)

First described by[1] J.A. Nelder and R. Mead (1965) (>**10000 citations**).

The NMM is an algorithm to minimize scalar functions

$$f : \mathbb{R}^n \to \mathbb{R}.$$

It is a so-called direct search method:

Direct search methods

A direct search method is an algorithm which is based on comparison of function values only.

$$\text{E.g.} \quad f_1 < f_2, \ldots$$

It does not need any information on derivatives (neither analytical, nor numerical).

$\Rightarrow$ Advantage: **The function does not need to be differentiable or continuous.**

[1] Computer Journal 7 (1965) 308-313;

The simplex

The basic element of the NMM is the so-called **simplex**.

The simplex

Consider $n+1$ points in $\mathbb{R}^n$. These points describe the vertices of a **simplex**.
The simplex is the *convex hull* of its vertices.

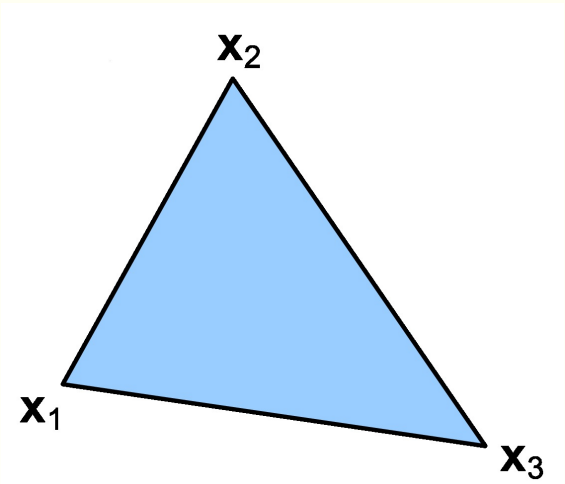
2 dimensions: triangle, 3 dimensions: tetrahedron,...

In the course of the algorithm the simplex can **change its form, orientation and position.**

The Nelder-Mead method: Initial simplex

Creation of the initial simplex:

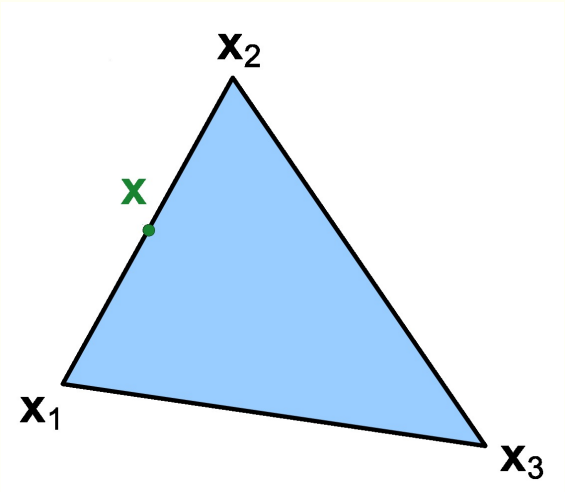
- Create a random simplex (in the domain of choice).
- Calculate function values $f_i = f(x_i)$.
- Order vertices such that $f_1 \leq f_2 \leq \ldots \leq f_{n+1}$.



The Nelder-Mead method: Centroid

Calculation of the centroid (barycenter of the n best points):

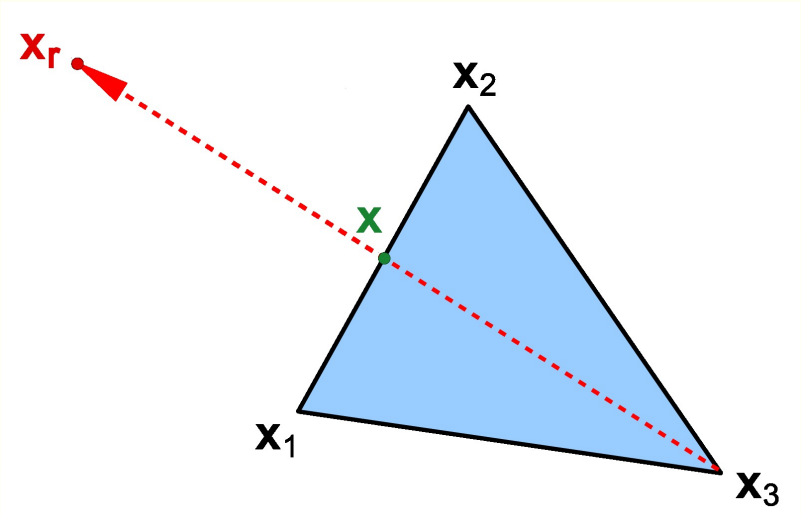
$$x = \frac{1}{n} \sum_{i=1}^{n} x_i.$$



The Nelder-Mead method: Reflection point

Calculation of the reflection point:

$$x_r = x + \rho(x - x_{n+1}).$$



$\rho > 0$ is the reflection parameter (standard choice: $\rho = 1$).

The Nelder-Mead method: Reflection point

We calculate the value f_r of f at the reflection point x_r.

$\Rightarrow$ 4 possibilities:

(1) x_r is better than x_n, but worse than x_1.
(2) x_r is better than all other points.
(3) x_r is better than x_{n+1}, but worse than all other points.
(4) x_r is worse than all other points.

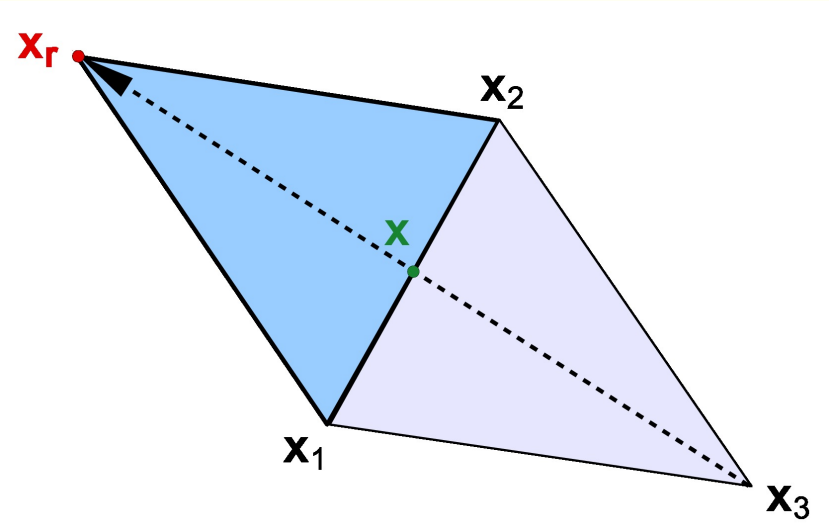
For each of this possibilities the NMM algorithm proceeds differently.

The Nelder-Mead method: Reflection

(1) x_r is better than x_n, but worse than x_1.

$$f_1 \leq f_r \leq f_n.$$

$\Rightarrow$ New simplex: $(x_1, \ldots, x_n, x_r)$ (*reflection*).



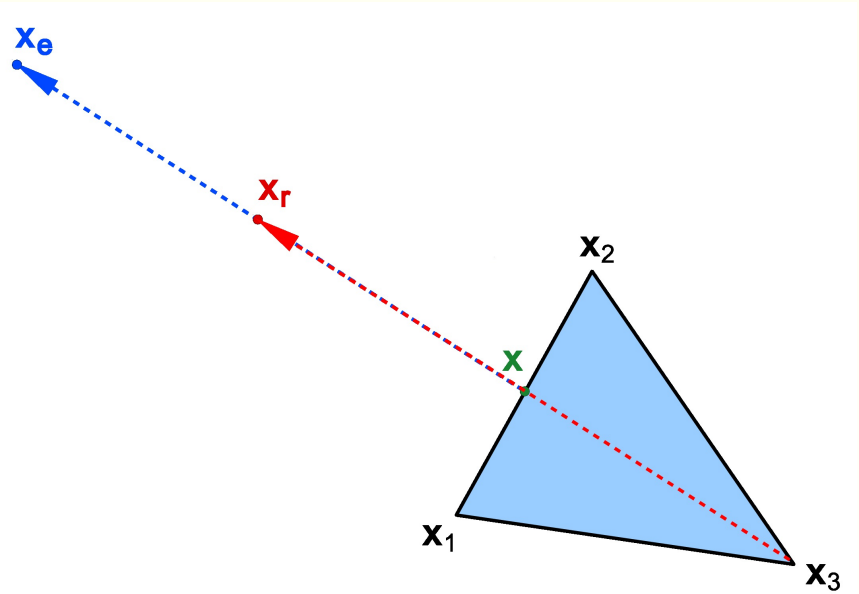
The Nelder-Mead method: Expansion point

(2) x_r is better than all other points.

$$f_r \leq f_1 \leq f_2 \leq \ldots \leq f_{n+1}.$$

$\Rightarrow$ Calculate expansion point x_e.

$$x_e = x + \chi(x_r - x).$$



$\chi > 1$ is the expansion parameter (standard choice: $\chi = 2$).

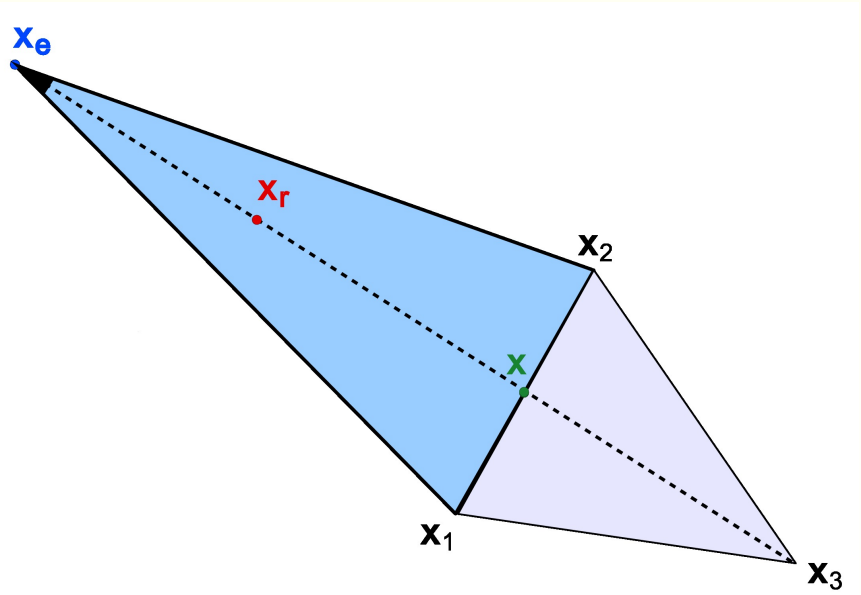
The Nelder-Mead method: Expansion

(2) x_r is better than all other points. $\Rightarrow$ Expansion point x_e.

$\rightarrow$ 2 possibilities

(2a) x_e is worse than x_r $\Rightarrow$ Accept x_r (reflection).

(2b) x_e is better than x_r $\Rightarrow$ Accept x_e (*expansion*).
$\Rightarrow$ New simplex: $(x_1, \ldots, x_n, x_e)$

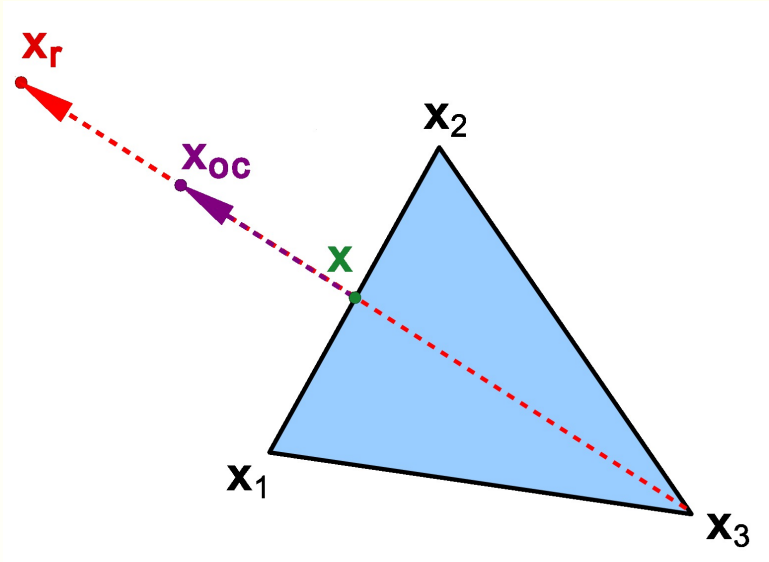


The Nelder-Mead method: Outside contraction point

(3) x_r is better than x_{n+1}, but worse than x_n.

$$f_n \leq f_r \leq f_{n+1}.$$

$\Rightarrow$ Try outside contraction



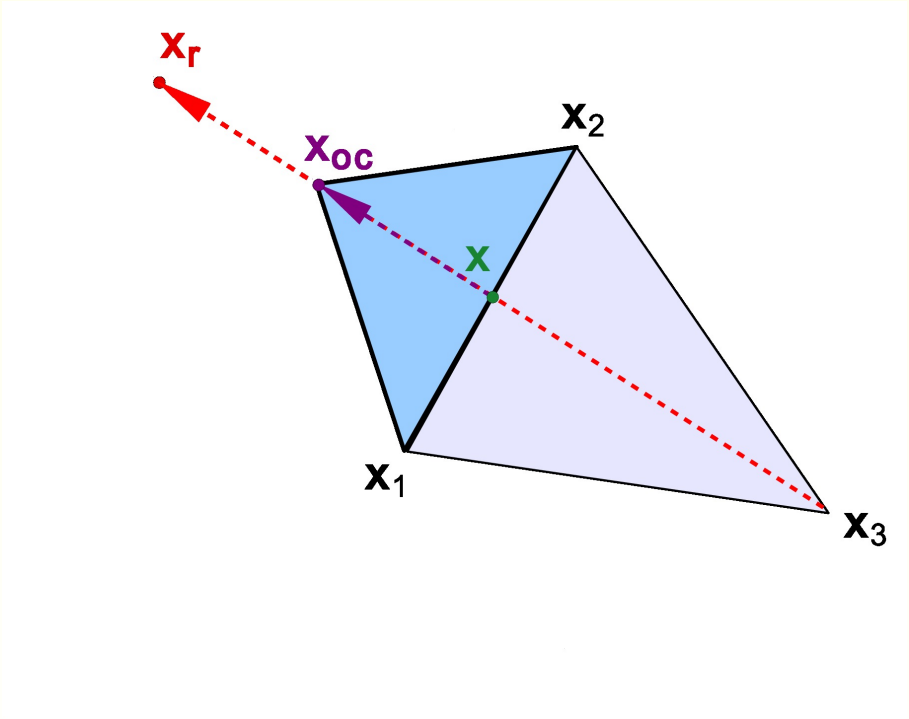
$$x_{oc} = x + \gamma(x_r - x).$$

$0 < \gamma < 1$ is the contraction parameter (standard choice: $\gamma = \frac{1}{2}$).

The Nelder-Mead method: Outside contraction

If x_{oc} is better than x_r, accept x_{oc} (*outside contraction*).

$\Rightarrow$ New simplex: $(x_1, \ldots, x_n, x_{oc})$



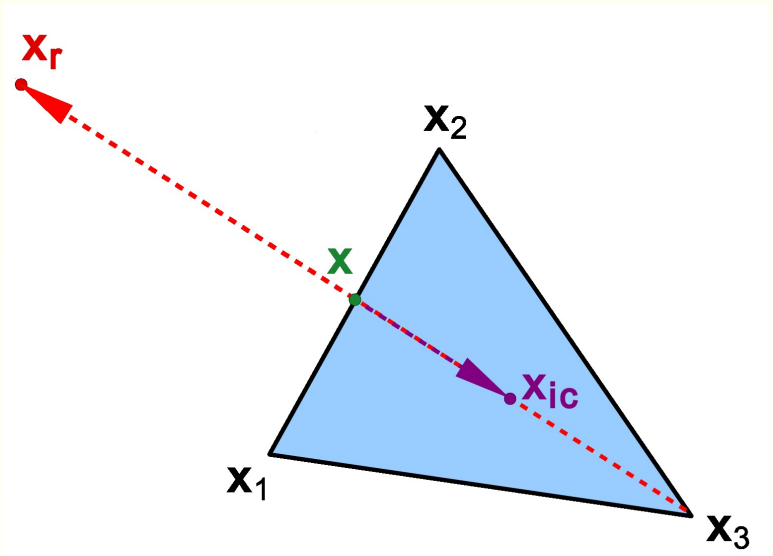
Else perform a shrinkage of the simplex.

The Nelder-Mead method: Inside contraction point

(4) x_r is worse than all other points.

$$f_r \geq f_{n+1}.$$

$\Rightarrow$ Try inside contraction

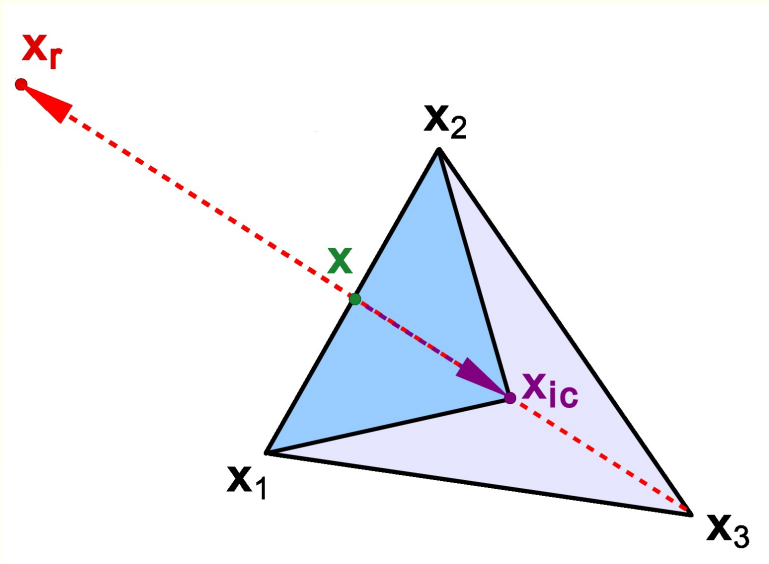


$$x_{ic} = x - \gamma(x - x_{n+1}).$$

The Nelder-Mead method: Inside contraction

If x_{ic} is better than x_{n+1}, accept x_{ic} (*inside contraction*).

$\Rightarrow$ New simplex: $(x_1, \ldots, x_n, x_{ic})$



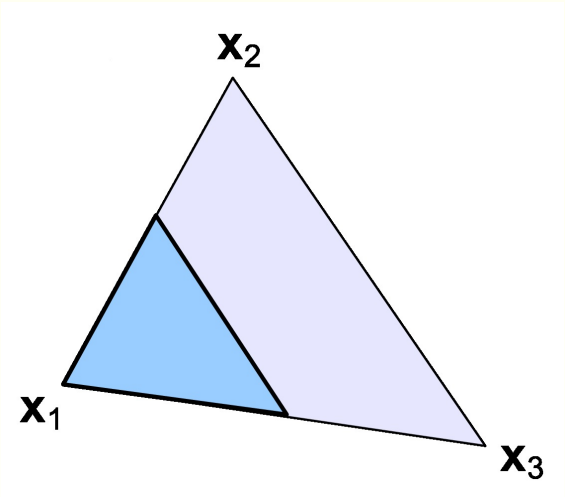
Else perform a shrinkage of the simplex.

The Nelder-Mead method: Shrinkage

If all else fails,...

...we perform a shrinkage of the simplex **towards the best point** x_1**.**

$$x_i \longrightarrow x_1 + \sigma(x_i - x_1).$$



$0 < \sigma < 1$ is the shrinkage parameter (standard choice: $\sigma = \frac{1}{2}$).

The Nelder-Mead method: Stopping criterion

Up to now: No stopping criterion for the algorithm. $\Rightarrow$ would run forever.

Criterion suggested by Nelder and Mead:

$$\frac{1}{n+1}\sum_{i=1}^{n+1}(f_i - \bar{f})^2 < \epsilon.$$

$$\bar{f} = \frac{1}{n+1}\sum_{i=1}^{n+1} f_i.$$

In words:

Stop when values of f on the vertices are close enough to each other.

Alternative: Stop when the vertices are close enough to each other, i.e. when the volume of the simplex is small enough.

The Nelder-Mead method: Convergence

Important question: **Convergence properties** of the NMM.

$\rightarrow$ A major problem of the NMM. Only few theorems on convergence known.

Lagarias et al.[2] (1998):

Convergence of the NMM

The NMM in one dimension converges $\Leftrightarrow \rho\chi \geq 1$.

According to this paper it is even unknown whether there exists *any* function $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ for which the NMM always converges to a minimum.

$\Rightarrow$ We need an **"emergency exit"**

If too many iterations:
Discard results and start with a new random simplex.

[2] SIAM J. Optim. Vol. 9, No. 1 (1998) 112-147

The Nelder-Mead method: Convergence

By construction the simplex always moves to smaller values of f.

$$\Rightarrow \textbf{Downhill simplex method.}$$

$\longrightarrow$ Even if NMM converges, it is only suited to find local minima.

$\longrightarrow$ Several possibilities for a way out of this dilemma.

1. Repeat the algorithm with many random initial simplices.
2. If local minimum found: perturb current simplex and start again (NM + perturbations).
3. Allow uphill moves, e.g.: NM + simulated annealing.

The Nelder-Mead method: Performance

How many iterations are usually needed to find the local minimum of a function f?

$\longrightarrow$ We need a test function, e.g.

$$T_n(x_1, \ldots, x_n) := \sum_{i=1}^{n} x_i^4$$

As initial simplices we use $n+1$ vertices whose coordinates are random numbers in $(-1, 1)$.

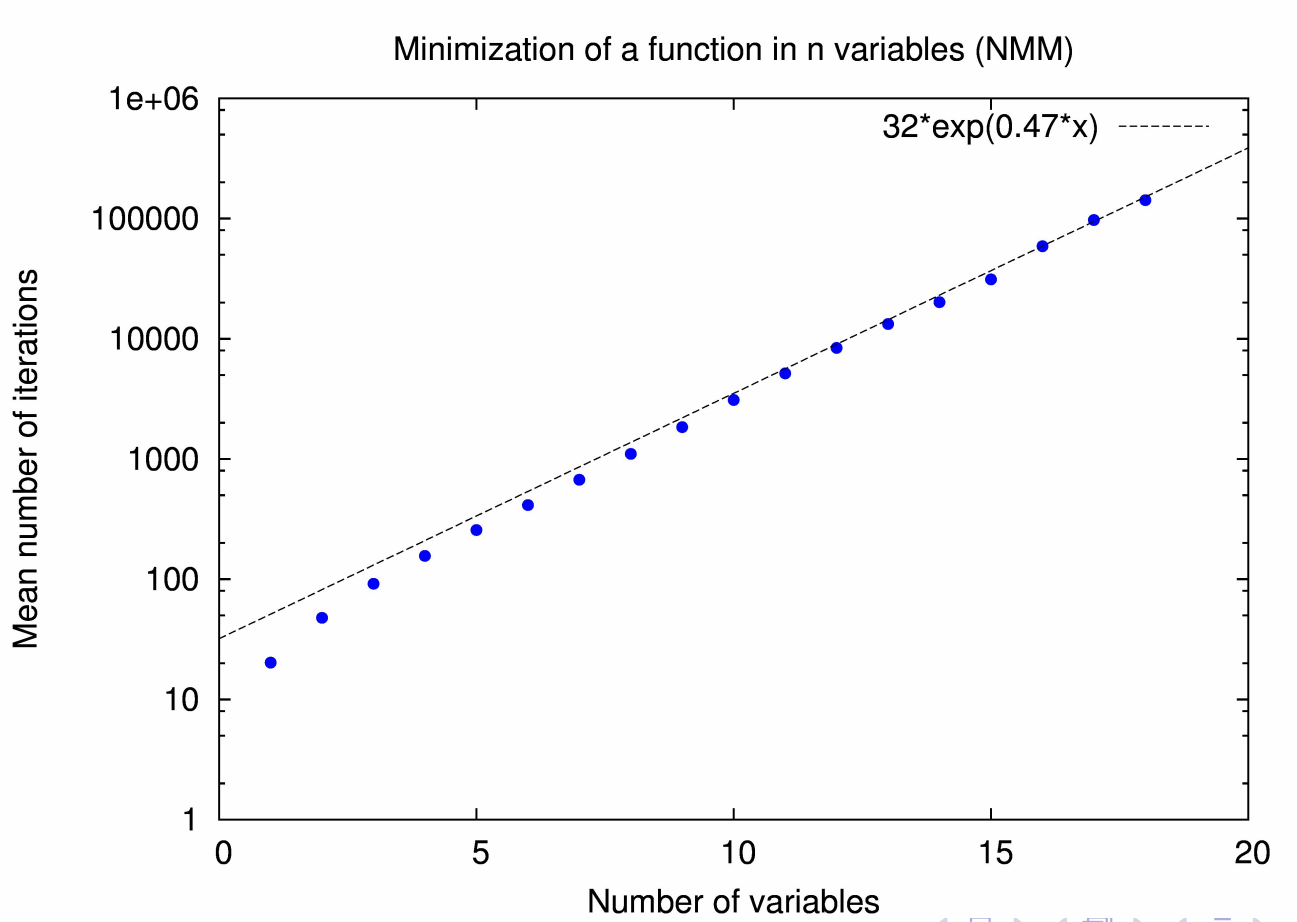
In order to achieve reasonable accuracy also for large n we set $\epsilon = 10^{-50}$ (Nelder, Mead (1965): 10^{-16}).

Found minima:
$$n = 1: \quad |x_i| \sim 10^{-7}$$
$$n = 5: \quad |x_i| \sim 10^{-7}$$
$$n = 10: \quad |x_i| \sim 10^{-6}$$
$$n = 15: \quad |x_i| \sim 10^{-6}$$

The Nelder-Mead method: Performance

How many iterations are usually needed to find a local minimum?

We minimize T_n (10000 times for each n) and determine the mean number of iterations needed to find a local minimum.

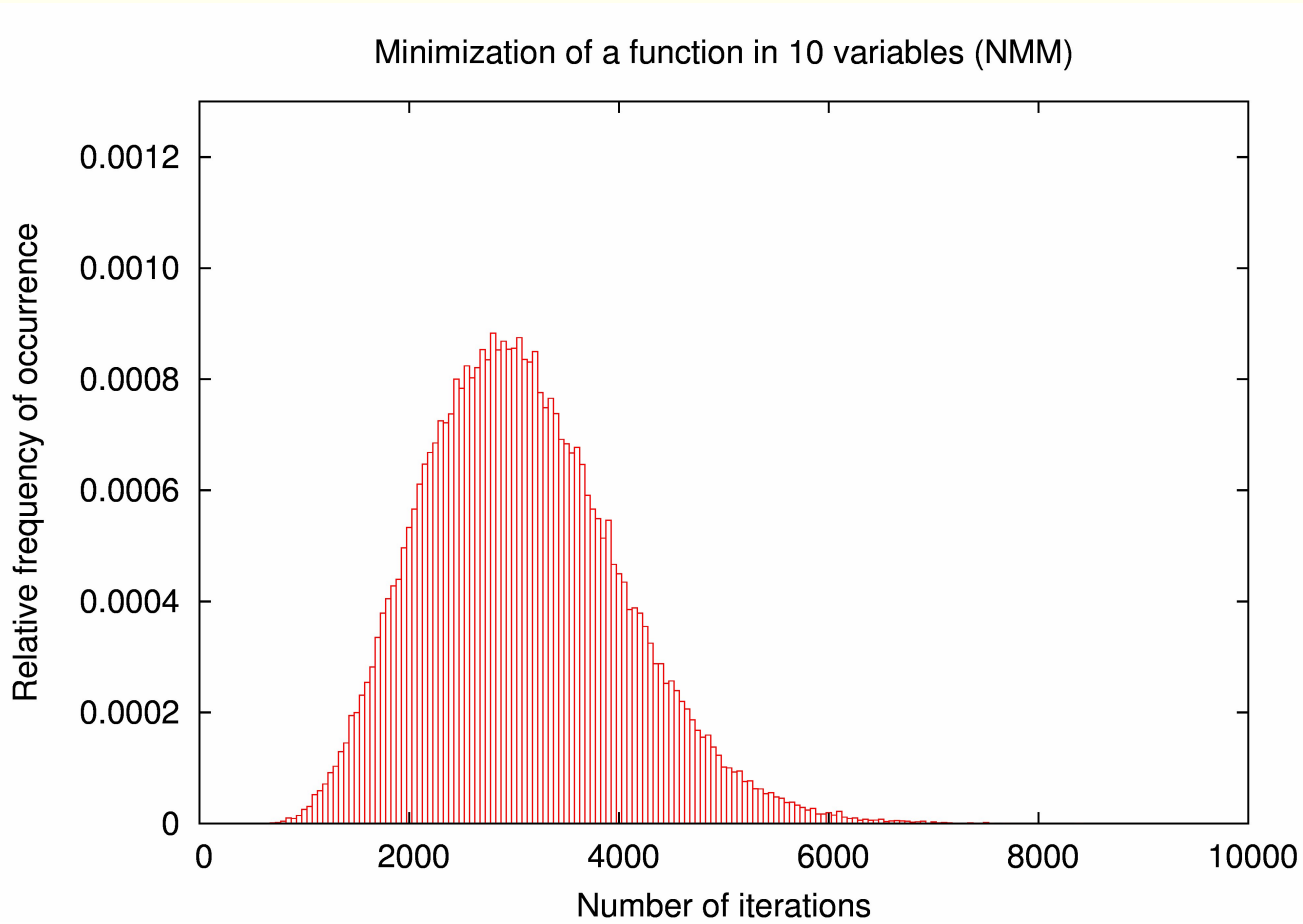


The Nelder-Mead method: Performance

How large is the influence of the initial simplex?

$\rightarrow$ We minimize T_{10} with 50000 random initial simplices $(x_i)_j \in (-1, 1)$.

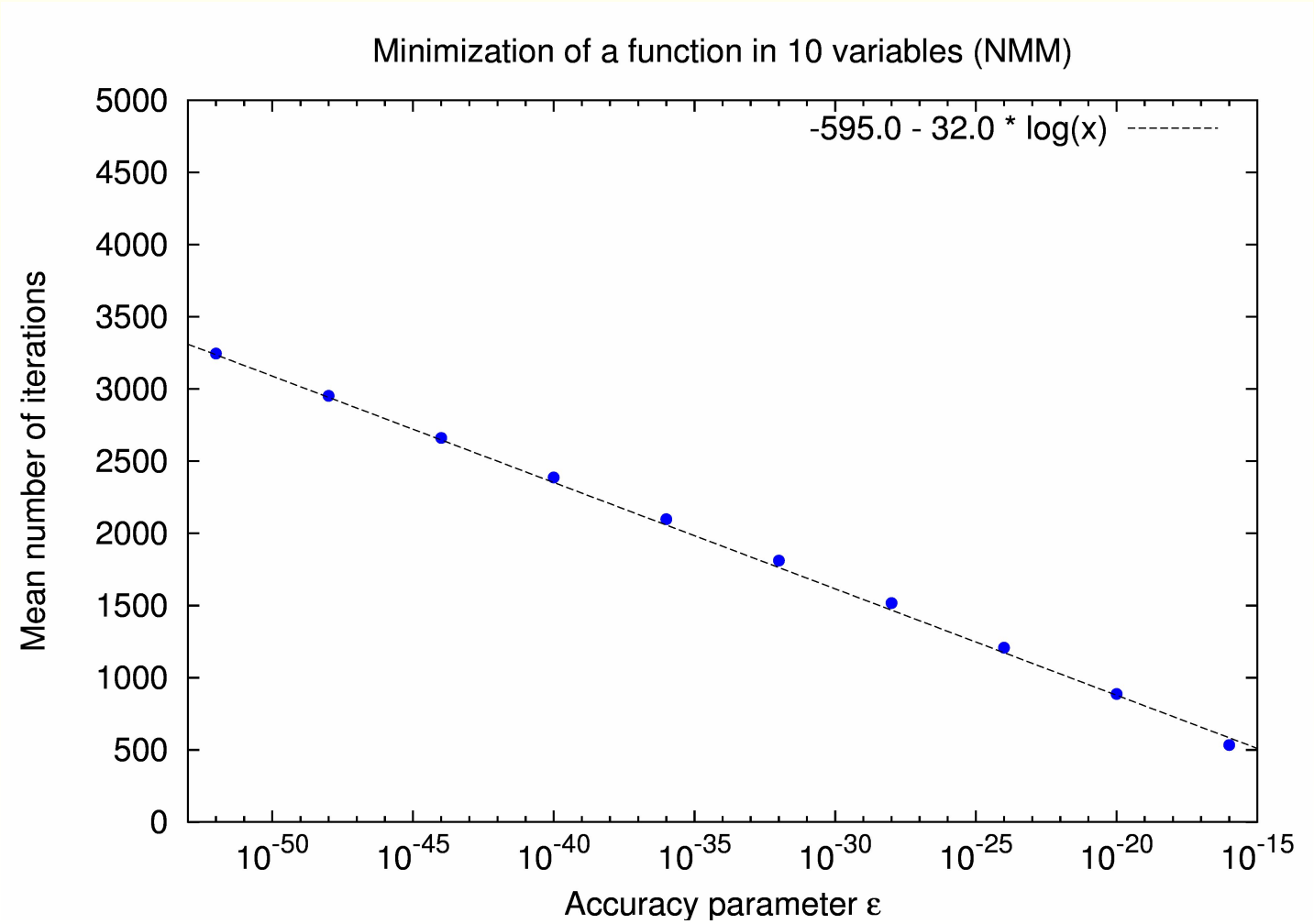
$\Rightarrow \approx$ 3000 iterations needed ($\epsilon = 10^{-50}$).



The Nelder-Mead method: Performance

If we use $\epsilon = 10^{-16}$ (as Nelder and Mead did): ≈ 500 iterations needed.

$\Rightarrow$ Accuracy parameter ϵ has a large influence on the computational effort.



Summary: Advantages and disadvantages of the NMM

Advantages:

- It is suited to deal with functions of many variables.
- It is easy to implement.
- It needs only function evaluations $\Rightarrow$ Suited to minimize non-differentiable functions.
- It usually needs only ≈ 2 function evaluations per iteration (exception: shrinkage).
- It has proven to work well in practice.

Summary: Advantages and disadvantages of the NMM

Disadvantages:

- The NMM can be very slow compared to other minimization routines (based on estimation of gradients).
- Little is known on its convergence properties for $n > 1$.
- There are situations where the algorithm fails[3] or converges extremely slowly.
- The simplex moves strictly downhill $\Rightarrow$ not suited to find global minimum.

[3] K.I.M. McKinnon, SIAM J. Optim. 9 (1998) 148-158

Towards application - The pinning term method

Restricting variables to a domain

Nelder-Mead algorithm "*lives*" on the whole space $\mathbb{R}^n$.

$\longrightarrow$ By the algorithm itself: No restriction on the variables possible.

$\longrightarrow$ We have to modify the function to implement additional restrictions.

E.g.: Restriction of the variables to a domain $D \subset \mathbb{R}^n$ can be done via replacing

$$f(\vec{x}) \mapsto \tilde{f}(\vec{x}) := \begin{cases} f(\vec{x}) & \text{for } \vec{x} \in D, \\ \infty & \text{for } \vec{x} \notin D. \end{cases}$$

The pinning term method

In the case of a χ^2-minimization one can also modify χ^2 to **pin down** an observable to a desired value.

$$\chi^2(\vec{x}) = \sum_{i=1}^{q} \left(\frac{P_i(\vec{x}) - \bar{\mathcal{O}}_i}{\sigma_i} \right)^2 .$$

can be replaced by

$$\tilde{\chi}^2(\vec{x}) = \sum_{i \neq j} \left(\frac{P_i(\vec{x}) - \bar{\mathcal{O}}_i}{\sigma_i} \right)^2 \underbrace{+ \left(\frac{P_j(\vec{x}) - \lambda}{0.01\lambda} \right)^2}_{\text{pinning term}} .$$

$\Rightarrow \tilde{\chi}^2$ becomes large, if $P_j(\vec{x})$ is not within a small 1%-region around λ.

$\Rightarrow P_j(\vec{x})$ effectively pinned down to λ.

$\rightarrow$ Enables to answer the question: How good can the fit get if an observable is restricted to a special value?

Towards application - An illustrative example

Preliminaries: Lepton mixing in a nutshell

Fermion mass terms in the Lagrangian (formulated as flavour eigenfields):

$$-\bar{\ell}_L \mathcal{M}_\ell \ell_R + \text{H.c.} \quad \text{(Dirac fermions)}$$

or

$$\frac{1}{2} \nu_L^T C^{-1} \mathcal{M}_\nu \nu_L + \text{H.c.} \quad \text{(Majorana fermions)}; \quad \mathcal{M}_\nu^T = \mathcal{M}_\nu.$$

$\mathcal{M}_\ell$ and $\mathcal{M}_\nu$ are (in the simplest case) 3×3-matrices. Transformation to mass eigenfields corresponds to diagonalization of the mass matrices:

$$U^\dagger \mathcal{M}_\ell V = \hat{m}_\ell, \quad W^T \mathcal{M}_\nu W = \hat{m}_\nu.$$

Lepton mixing matrix:

$$U_{\text{PMNS}} = U^\dagger W \quad \text{(unitary)}.$$

U_{PMNS} parameterized by three mixing angles θ_{12}, θ_{13}, θ_{23} and six phases.

Example: Texture zeros

Assumptions:

- Majorana neutrinos $\Rightarrow \mathcal{M}_\nu$ symmetric.
- Charged lepton mass matrix $\mathcal{M}_\ell$ diagonal.
  $\Rightarrow U = \mathbb{1} \Rightarrow U_{PMNS} = W$.
- We assume **texture zeros** in $\mathcal{M}_\nu$.

$$\mathcal{M}_\nu = \begin{pmatrix} a_1 & 0 & a_2 \\ 0 & 0 & a_3 \\ a_2 & a_3 & a_4 \end{pmatrix}$$

$a_j = r_j e^{i\varphi_j}$ are complex parameters. Global phase not relevant.
$\Rightarrow$ We set $\varphi_1 = 0$.

$\Rightarrow \mathcal{M}_\nu$ has 7 real free parameters.

Example: Texture zeros

Our question: Are these assumptions compatible with the experimental data?

$\Rightarrow$ We perform a χ^2-analysis.

Free parameters: $r_1,\ r_2,\ r_3,\ r_4, \varphi_2,\ \varphi_3,\ \varphi_4$.

Observables: $\Delta m^2_{21},\ \Delta m^2_{31}, \sin^2\theta_{12}, \sin^2\theta_{13}, \sin^2\theta_{23}$

$$\chi^2(r_j, \varphi_j) := \sum_{i=1}^{q} \left(\frac{P_i(r_j, \varphi_j) - \bar{\mathcal{O}}_i}{\sigma_i} \right)^2 .$$

Additional constraint: From cosmology: Sum of all three neutrino masses smaller than ~ 1 eV (let's say 2 eV).

$$\chi^2_{\text{cosm.}} := \begin{cases} 0 & \text{if } m_1 + m_2 + m_3 < 2\,\text{eV} \\ \infty & \textit{else} \end{cases}$$

$$\chi^2 \mapsto \chi^2 + \chi^2_{\text{cosm.}}.$$

Example: Texture zeros

$\Rightarrow$ Minimize χ^2!

Steps involved in calculating χ^2:

1. Singular value decomposition of $\mathcal{M}_\nu$ ($\rightarrow$ LAPACK).
   $\Rightarrow m_1$, m_2, m_3, W.
   (Further assumption: normal neutrino mass spectrum:
   $m_1 < m_2 < m_3$.)
2. Calculate mass squared differences $\Delta m_{ij}^2 = m_i^2 - m_j^2$ and $sin^2\,\theta_{ij}$.
3. Calculate χ^2.

- Start values (for random simplices): $r_j \in [0, 5\,\text{eV}]$; $\varphi_j \in [0,\, 2\pi)$.
- Maximal allowed number of Nelder-Mead iterations: 10000.
- Number of random initial simplices: 1000.

After 44.6 seconds (Intel(R) Core(TM) i7 CPU 870 @ 2.93GHz):

$$\chi^2_{\text{min}} = 8.74 \times 10^{-2}.$$

$\Rightarrow$ Assumptions compatible with data.

Example: What else can we learn?

This special set of texture zeros implies that[4]

$$\text{Neutrino mass spectrum quasi-degenerate (i.e. } m_1 \text{ large)} \Rightarrow \sin^2\theta_{23} \approx 1/2.$$

$\Rightarrow$ Strategy: We **pin down** m_1 to a large value (0.2 eV), and $\sin^2\theta_{23}$ to different values between 0 and 1.

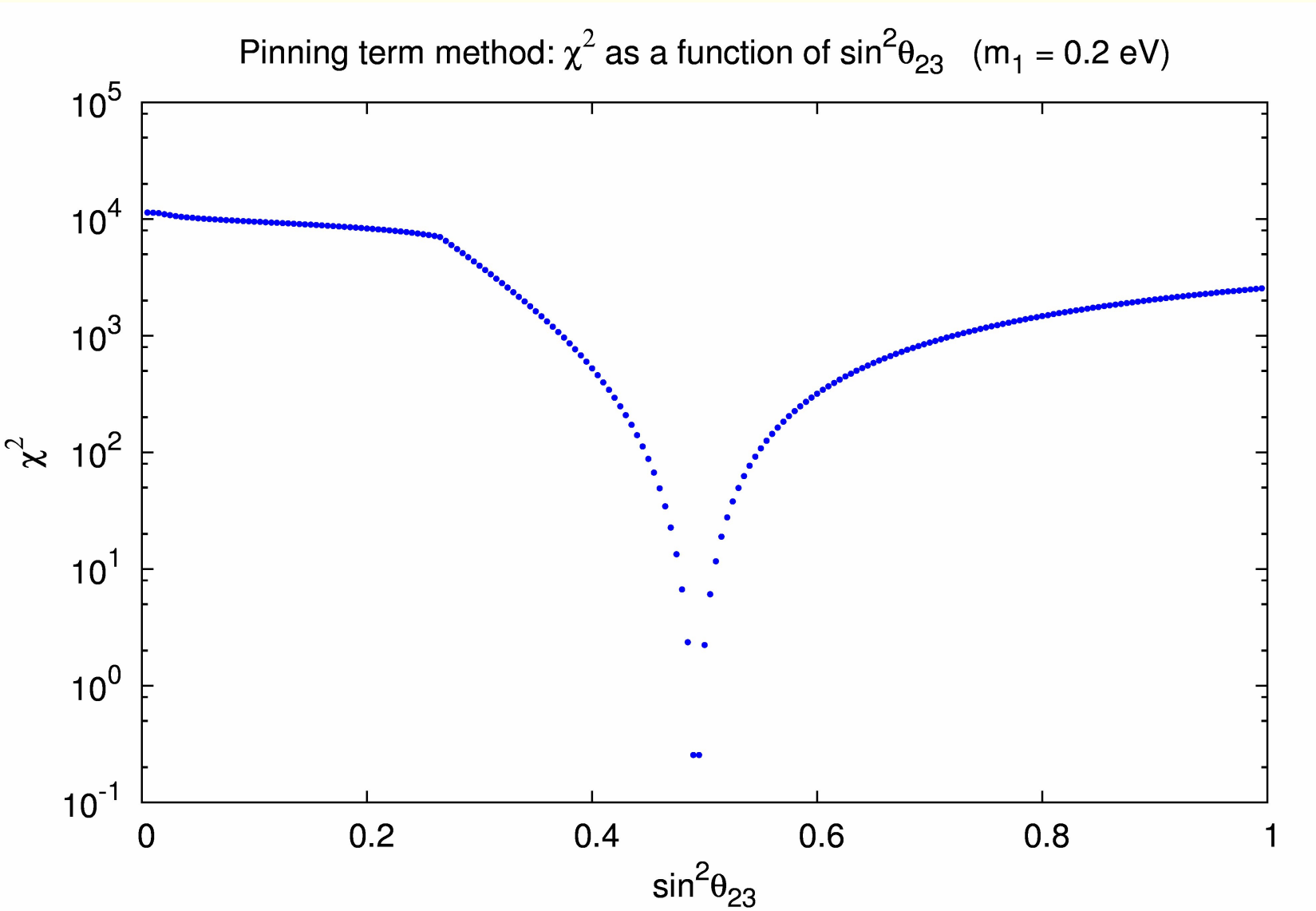
$$\chi^2 := \sum_{\mathcal{O}_i \neq \sin^2\theta_{23}} \left(\frac{P_i - \bar{\mathcal{O}}_i}{\sigma_i}\right)^2 + \chi^2_{\text{cosm.}} + \left(\frac{m_1 - 0.2\,\text{eV}}{0.01 \times 0.2\,\text{eV}}\right)^2 + \left(\frac{\sin^2\theta_{23} - \lambda}{0.01\lambda}\right)^2$$

We minimize this χ^2-function for different $\lambda = \sin^2\theta_{23}$.

$$\Rightarrow \text{Plot } \chi^2(\sin^2\theta_{23}).$$

[4] W. Grimus and P.O. Ludl, Phys.Lett. B700 (2011) 356-361

Example: What else can we learn?



Summary

- Minimization of χ^2-functions is an appropriate tool to link models with data.
- For many realistic applications χ^2 will be a (probably complicated) non-differentiable (or non-continuous) function.
  $\Rightarrow$ Numerical methods which rely on analytic knowledge or numerical approximation of derivatives are not applicable.
  $\Rightarrow$ We need a direct search method.
- The Nelder-Mead method is appropriate, because
  - it can deal with a very high number of variables,
  - it only needs ~ 2 function evaluations per iteration (except shrinkage).
    $\Rightarrow$ For a direct search method it is very fast.
- The pinning-term method allows to put additional constraints on the variables.
  $\Rightarrow$ Possibility to extract physical predictions.

Thank you for your attention!