ADVANCED PHOTON SOURCE

Beamline Controls & Data Acquisition

An algorithm for efficient optimization directed by noisy data

ESRF Workshop on Automatic Beamline Alignment
Tim Mooney
March 12, 1999

ABSTRACT

At the heart of several efficient nonlinear-optimization algorithms is a parabolic fit in which the function (or a one-dimensional "slice" of its hypersurface) is represented by three evaluations in the neighborhood of a local extremum. (The extremum of the parabola through those points is assumed to approximate the function's true extremum.) The three-point solution fails (optimization converges slowly or not at all) when function evaluations contain enough noise, granularity, etc. The algorithm presented here solves analytically for the best-fit parabola and line(s) to four or more data points, calculates the uncertainties of fit parameters, compares chi-square values of the fits, and uses various heuristic information to direct the search for an optimal value.

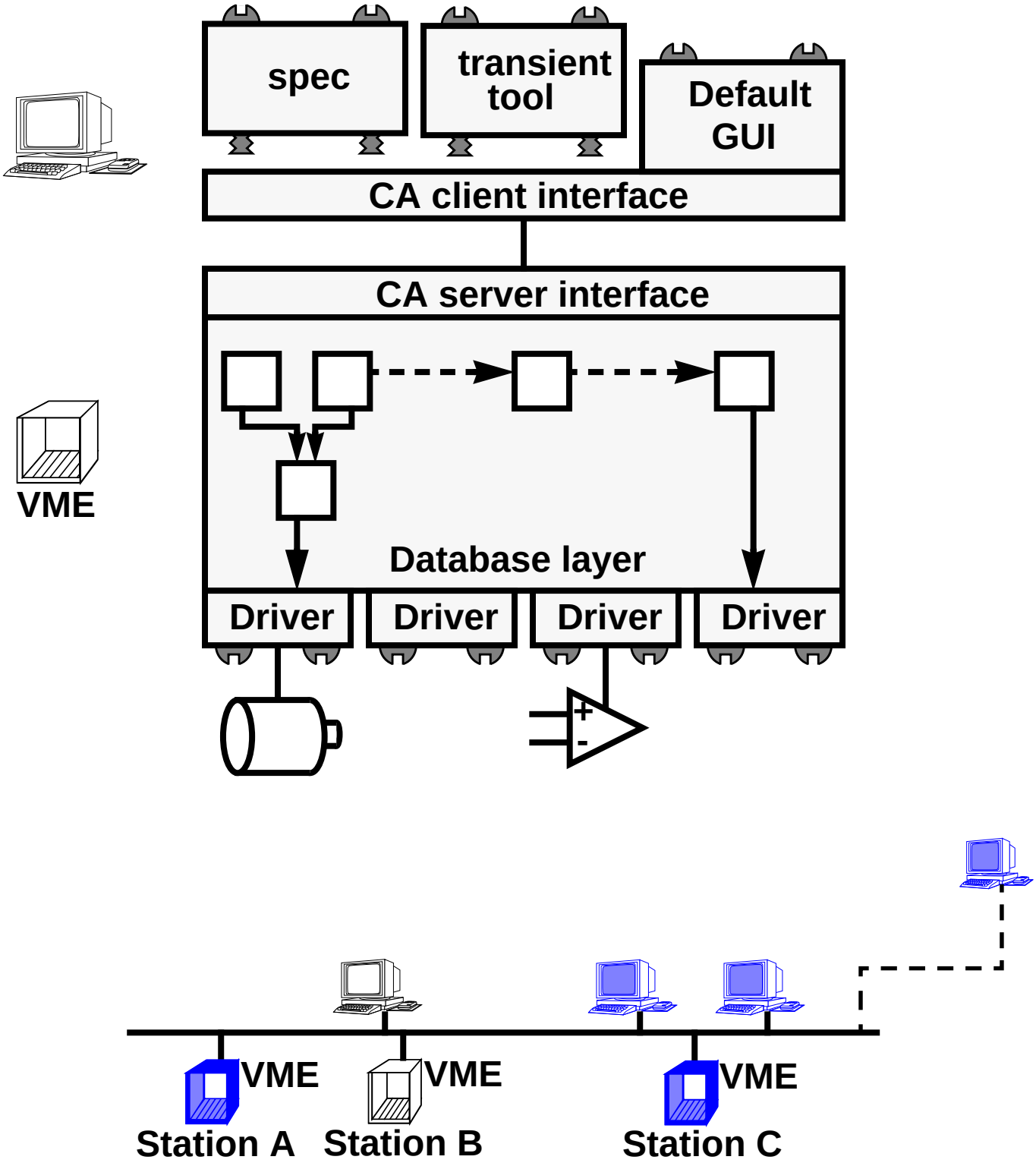
Objective: N-dimensional optimization tool for online use

- User specifies:
  - Positioners, limits, step sizes, convergence criteria
  - Detector trigger(s)
  - Signals to be acquired
  - Function to be optimized
  - (Looks pretty much like a scan specification)

- Algorithm must be efficient and robust
  - Function evaluations are many orders of magnitude more costly than in typical data-analysis applications
  - Can't waste beam time
  - Can't challenge equipment-protection system

- Noise, granularity, etc. limit algorithm options:
  - Can't use derivatives
  - Can't base any major decision on a single data point
  - Strongly correlated parameters are indistinguishable

Context: EPICS-based beamline software architecture



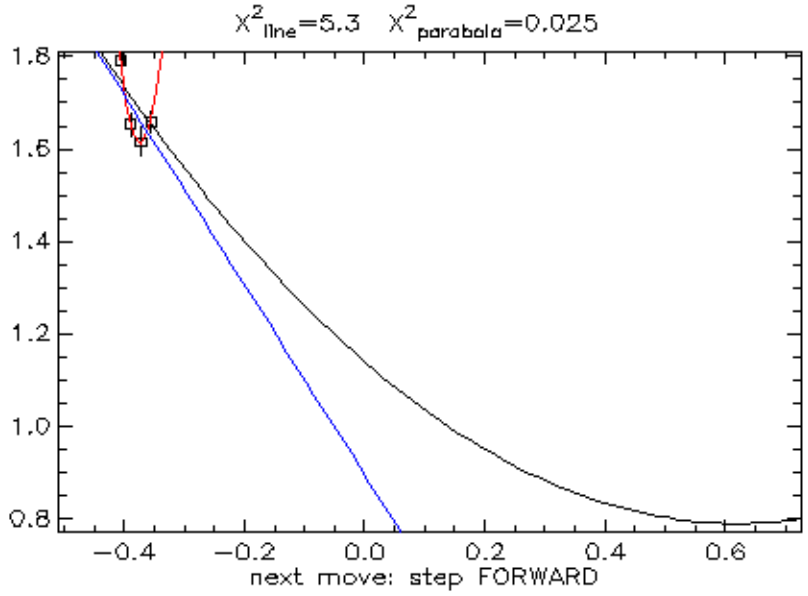
Algorithm

- Starting points
  - Powell's conjugate gradient method (*e.g., Numerical Recipes*)
    - Turns **n**-dimensional search into **m** 1-dimensional searches
    - **n**D decisions based on **1**D results (i.e., *several* data points)
  - Analytical fit of polynomial to data (*e.g., Bevington 2nd Ed.*)
    - Build line and parabola fits
    - Gives error estimate for fit parameters

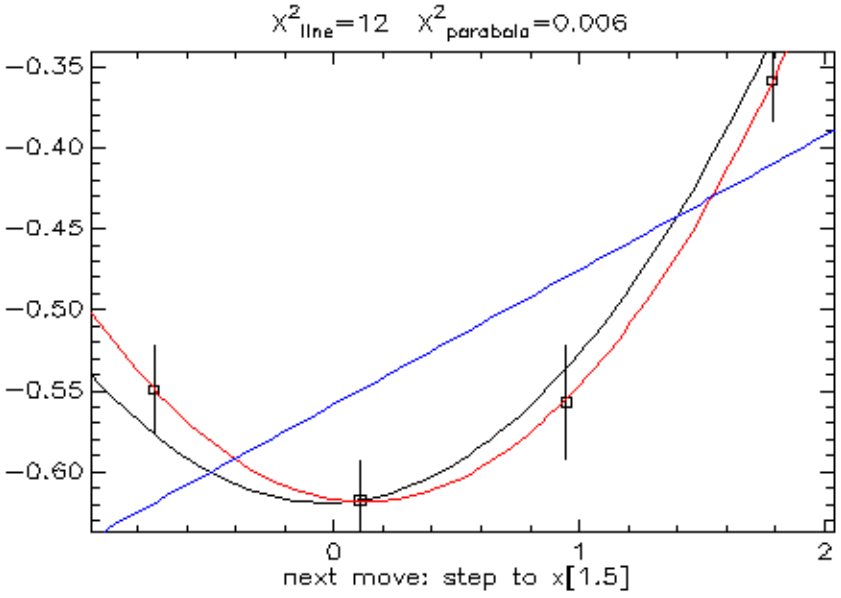
- Search strategy
  - Compare χ^2_{parabola} to χ^2_{line} (*i.e.,* χ^2 *per degree of freedom*)
  - Require $0.5 < \chi^2_{\text{parabola}} < 2$ before using result to direct search. (Exact criteria depends on functional form of extremum)
  - Require roughly symmetrical distribution of >5 data points about extremum before using error estimate for χ^2_{parabola}Must have a reasonable estimate of detector noise to use χ^2 in this way. (Could use the ratio $\chi^2_{\text{line}}/\chi^2_{\text{parabola}}$ without a noise estimate.)

Examples

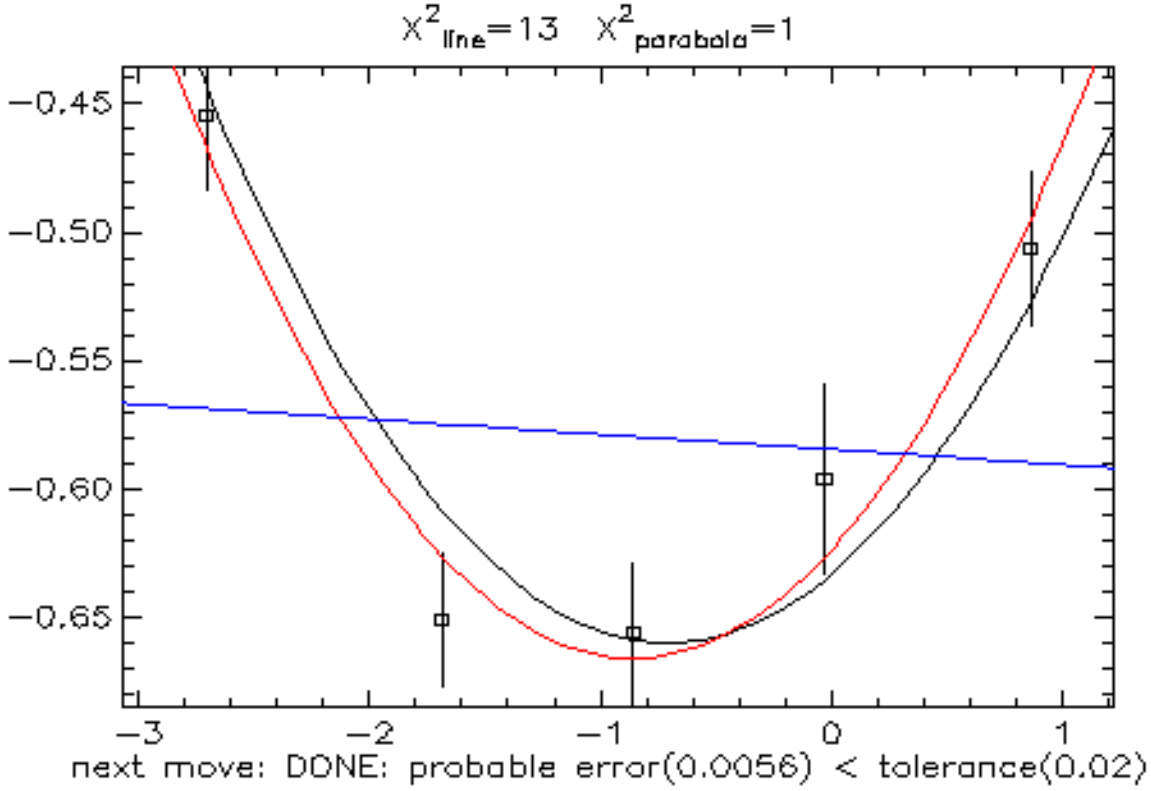
- missing the point (χ^2too small)



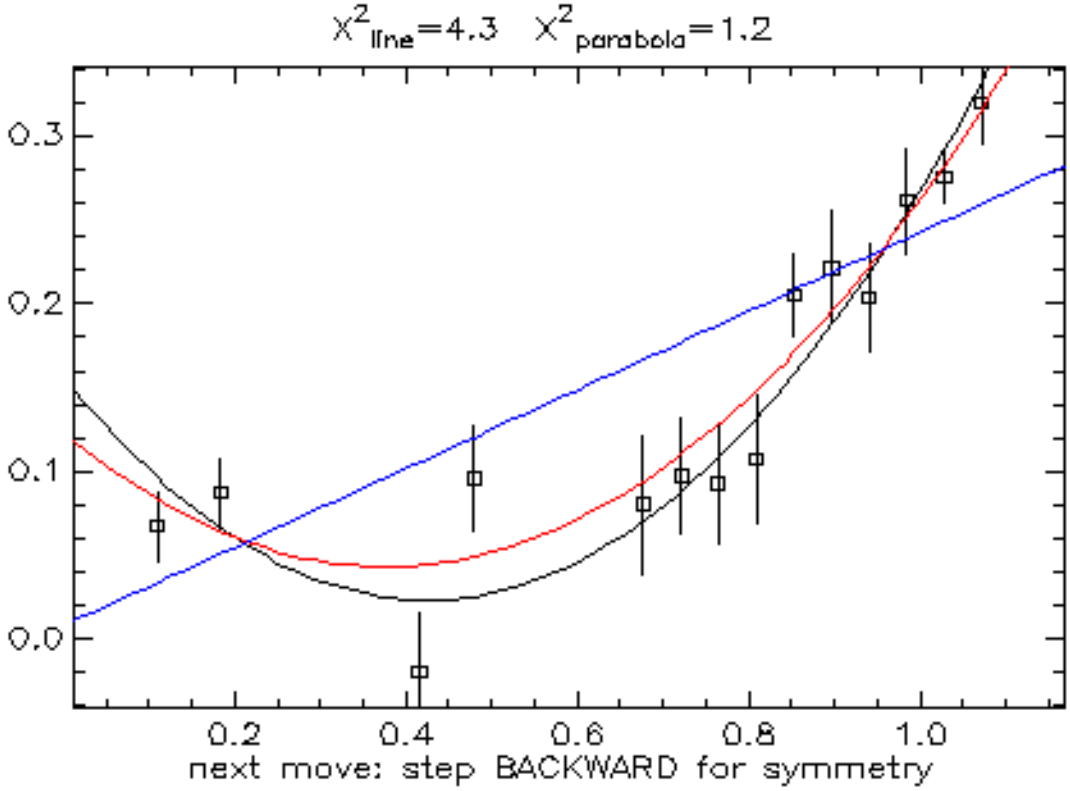
- ditto



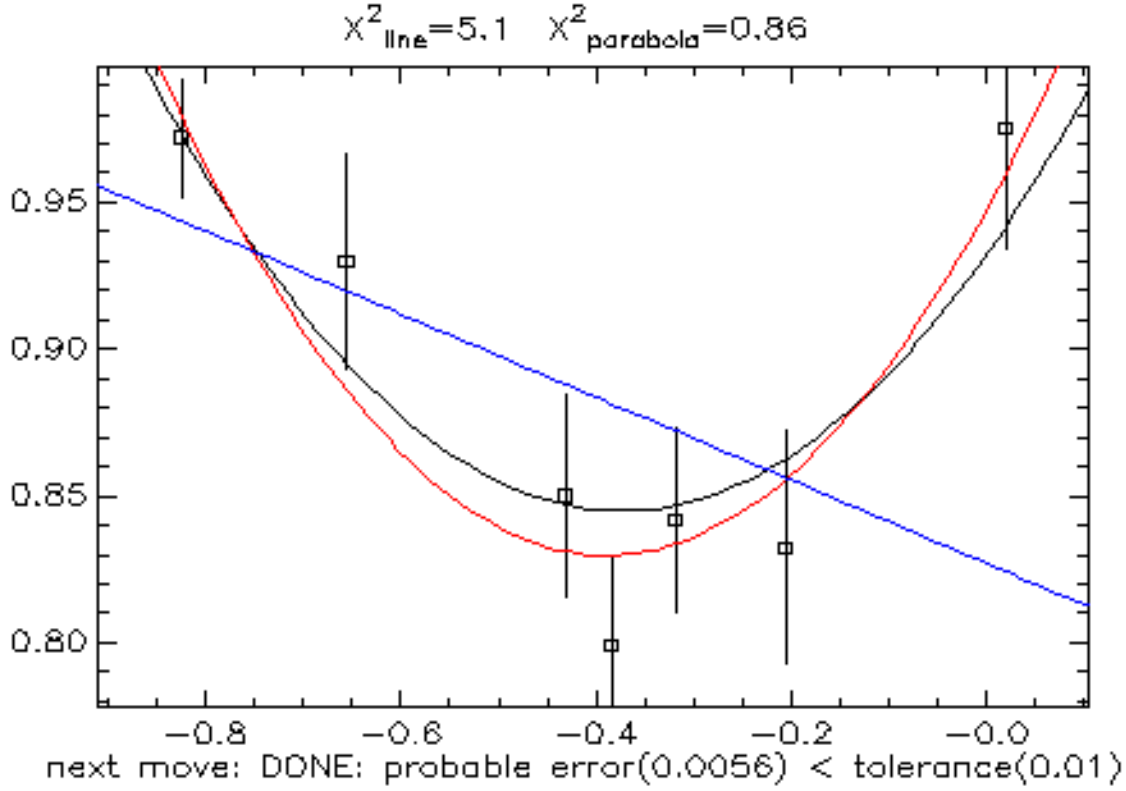
Examples: fooled



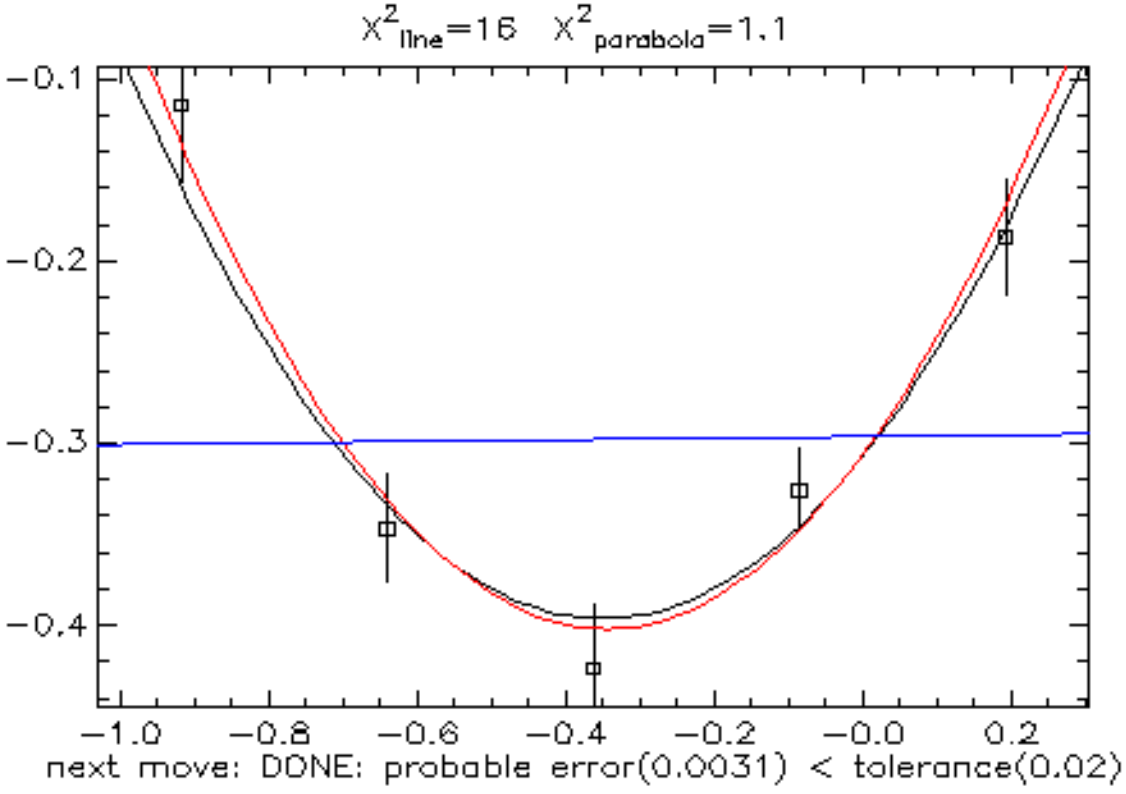
- need symmetry



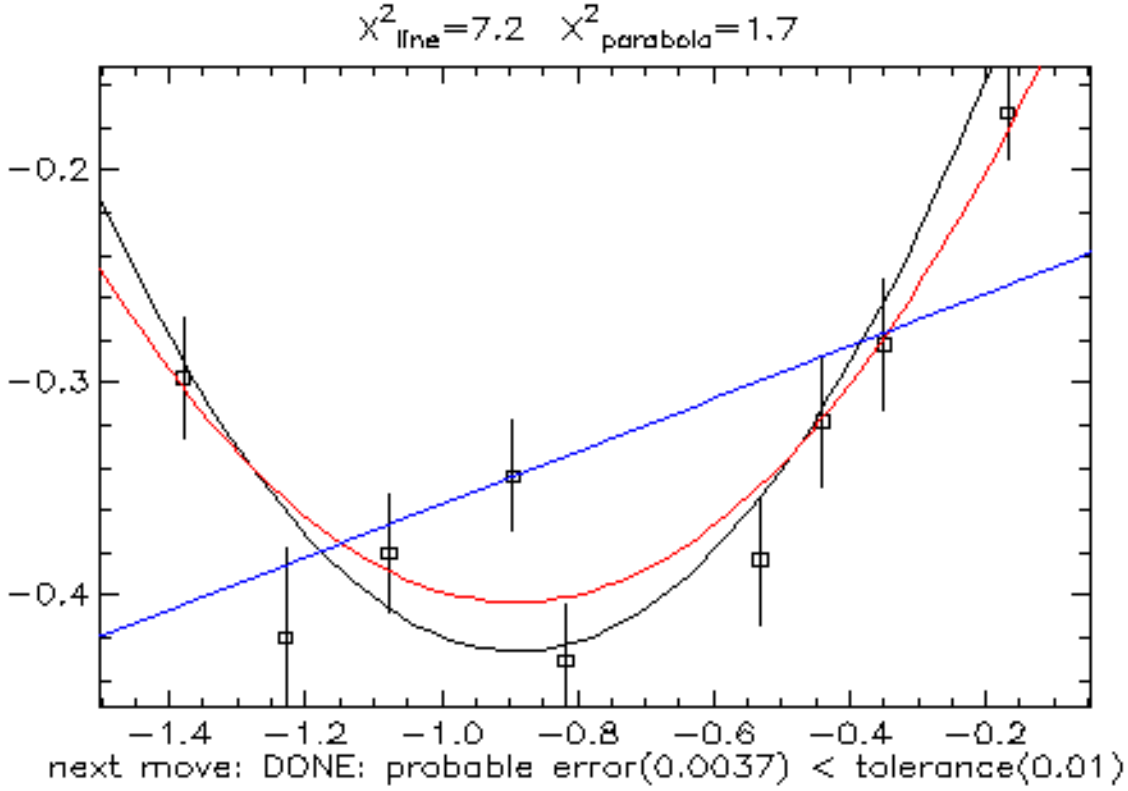
- shallow minimum



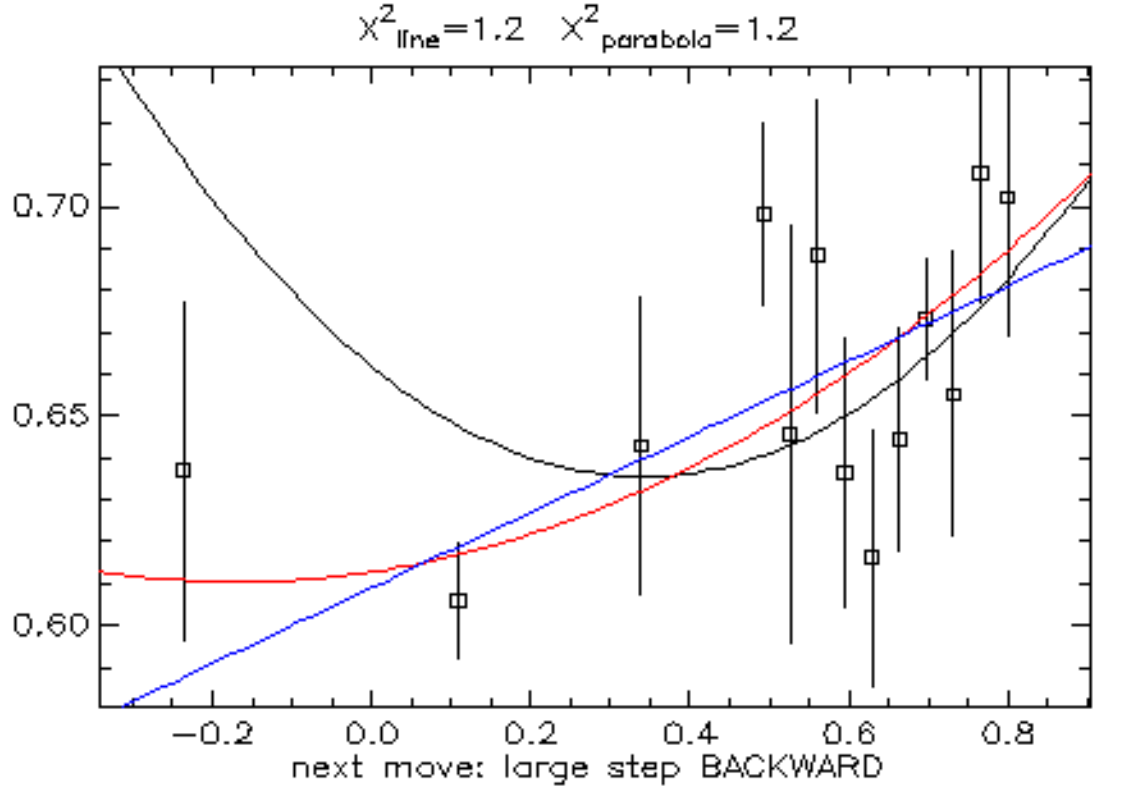
- good start point and step size



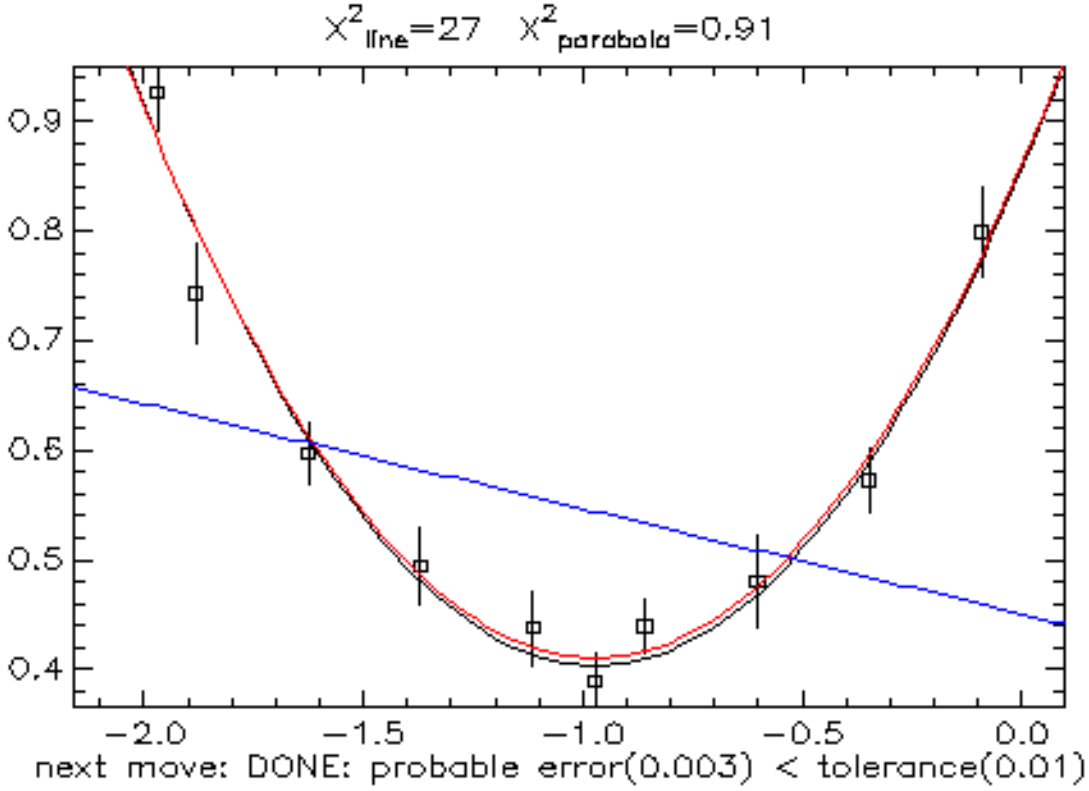
- got lucky



- recovering from bad step size



- evenly distributed data-point errors



- very shallow minimum