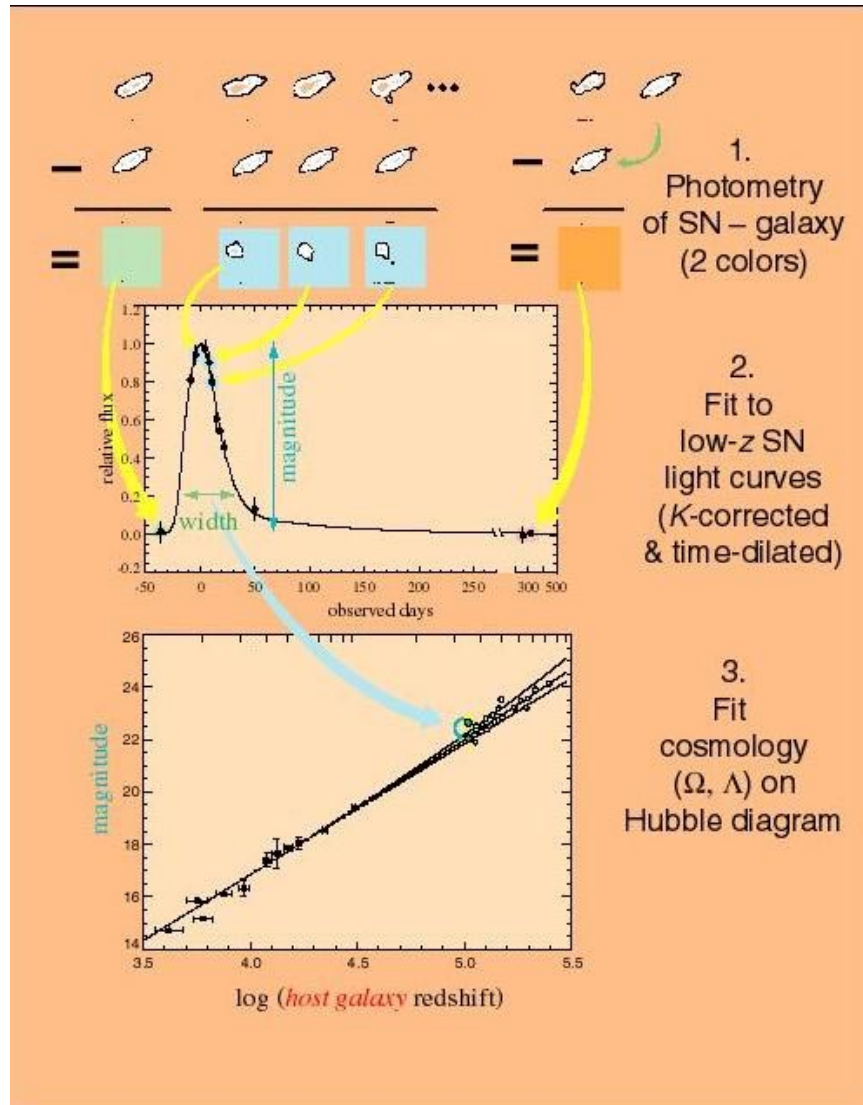


snminuit $\implies$ snmin02

Lightcurve fitting program update plans and progress

Don Groom, LBNL

SCP Meeting, 12–13 June 2002



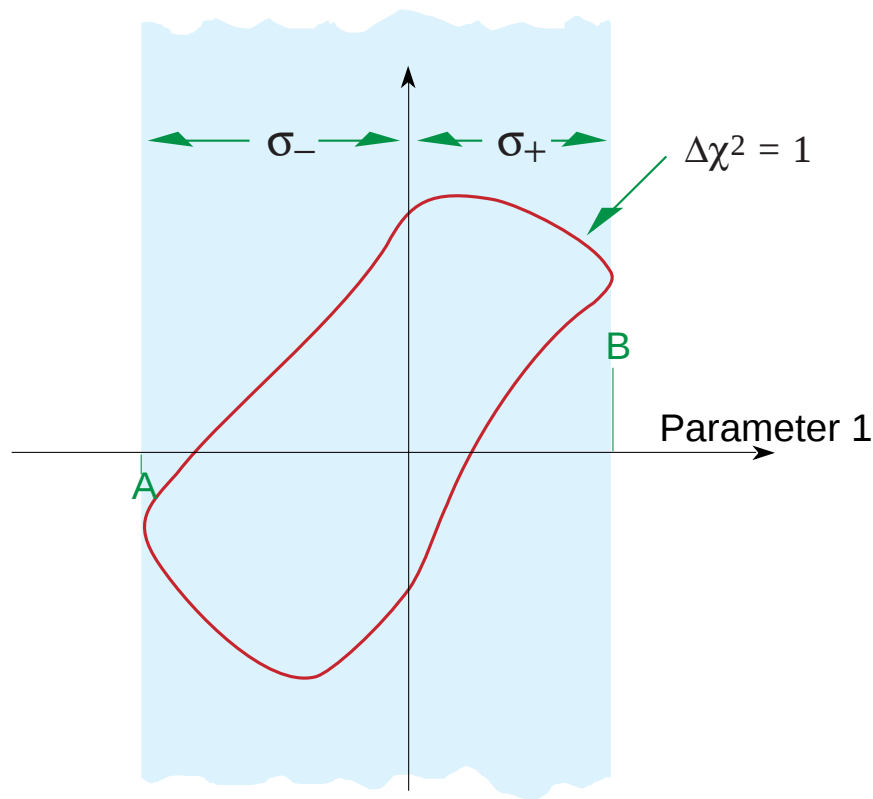
`snminuit.pro` evolved (somewhat painfully) in the 42SNe analysis

- It could fit standard templates to one or two standard lightcurves, using standard K corrections
- It defaulted to the most recent lightcurve version in `roblight` or, after that, in `corrlight` ...but one could override with a user-specified lightcurve file(s)

⇒ Lightcurve is
 $f(t_{max}, stretch, height_{C_1}, height_{C_2}, \dots, base_{C_1}, base_{C_2}, \dots)$
 (but parameters can be fixed)

Why use MINUIT, that old CERN warhorse, written in *shudder* FORTRAN, rather than something simpler like IDL's *curvefit* or a Numerical Recipes steepest descent method?

- It provides the full error matrix in a nonlinear fit.
 $\implies$ *Correlations are important!*
- Contains two quite different minimization algorithms and “changes gears” if it runs into trouble with one of them (look at the output file sometime!). Deals with negative determinants, etc.
- It forces user to calculate χ^2 or a likelihood $\implies$ this makes it a “mathematical minimizer” rather than a “statistical package.” But given a properly scaled function, function differences can be used to do statistics
- Metric-independent “stop” at the minimum, rather than use of first derivative
- MINOS analysis and other tasks use nontrivial finite-difference algorithms to calculate first and second derivatives
- It's been debugged and enhanced for 40 years!

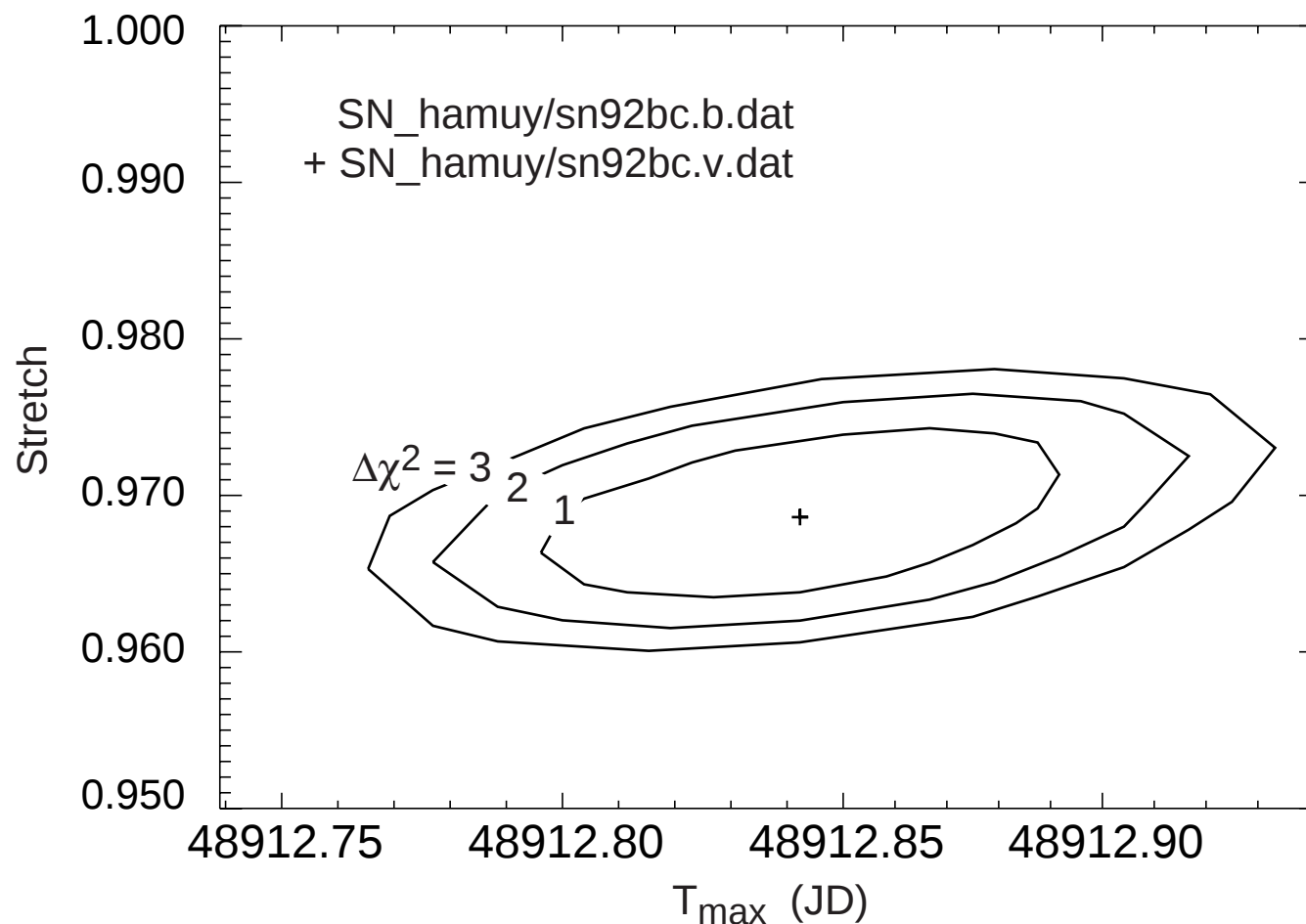


MINOS is probably the first, and is still only one of a few, generally available program to calculate parameter errors taking into account both nonlinearities and parameter correlations

In general MINOS errors are asymmetric and are often larger than (symmetric) parabolic errors

`snminuit` cum `snmin02` uses average of MINOS errors as the assigned parameter error. User should *always* inspect and compare

MINUIT has lots of other options that snminuit hasn't used, such as contour finding



Problems with existing code:

- Call cumbersome, especially if non-default lightcurve files
- Need user-defined K corrections
- Limit of 2 colors simultaneously fit
- Should not make first lightcurve special (“all colors are created equal”)
 - Time should be UT with respect to B max
 - Vertical scales should be given for all colors, not just the first with color differences for the other(s)
 - Time should be UT with respect to B max
- Other enhancements and code cleanup to make Rob happy (he’s now sad)

So far, only the call has been generalized:

```
IDL> snmin02, 'traditional.dat'
```

The data file `EarlScruggs.dat` is of the form:

`EarlScruggs`

```
v , r
```

```
u,r ,kcorr= kcorrQ.dat
```

```
b, i ,file= /home/sierra1/deg/idl/lightfits/sn9621.rdat
```

```
v,hj , file=anything.ext,kcorr = kcorr2file
```

- IDL-like format; comma delimited and spaces don't count
- first two arguments are restframe and observer colors
- keywords 'kcorr' and 'file' override defaults

●●● MY OTHER LOGJAMS ARE PAST, AND I CAN MAKE PROGRESS
ON THE REST DURING THE SUMMER ●●●

Rob will be happy