Overview over the Capabilities (and missing pieces?) of the Fittino Code

Philip Bechtle

October 29th 2009

1. What Fittino is

2. What do we have?
   - Theory
   - Observables
   - Technologies

3. What's missing?

Aims of Fittino

- Fit the (N)MSSM parameters to the observables from all possible sources: LHC, ILC, Tevatron, B-factories, etc.
- Fit high-scale (mSUGRA, GMSB, AMSB) and low-scale models (MSSM,NMSSM)
- Bottom-up approach
- If the user wishes to test unbiased measurement: no prior knowledge of the parameters at any step
- Provide easy user interface for measurements, parameter definitions and output
- Goals:
  - Show that unambiguous parameter determination without human bias is possible
  - Determine precision of parameter measurements
  - Test the necessary experimental and theoretical precision
  - Study comparisons of models: MSSM vs. NMSSM etc.

1. What Fittino is

2. What do we have?
   - Theory
   - Observables
   - Technologies

3. What's missing?

Models

- mSUGRA, GMSB
- MSSM24
- MSSM24 with arbitrary other assumptions
- NMSSM
- No CMSSM, no models with partial unification (e.g. fit $\tan\beta$, m_0, A_0, μ, M_1, M_2, M_3) $\longrightarrow$ need codes
- everything can be fitted wit hand without an arbitrary number of SM parameters

Calculators for Predictions

- Everything which works with SLHA
- Experience with
  SPheno, micromegas, "mastercode", SuSpect, SoftSUSY
- Some requirements:
  - Code should not write any SLHA output in case of internal error
  - Code should, at best, always exit with retuurn value $\neq 0$ in case of internal error
  - For spectrum calculators: Please return every relevant particle mass, also the likes of m_W etc.
- can use different calculators in parallel for different observables, hard-coded...
- Missing: Select different calculators in the input file for each observable

Observables and other Inputs

Observables:

- Fittino can fit to any combination of the following observables:
  - Masses of SM and MSSM particles
  - Edges in mass spectra
  - Particle widths
  - Branching fractions
  - Cross-sections
  - Any product of cross-sections and/or branching fractions
  - Any ratio of the above
  - LE-Observables: $b \to s\gamma$, $(g-2)_\mu$, relic density, $\Delta m_s/\Delta m_d$, etc.
- Correlations among observables can be specified
- Limits on masses and BR of unobserved particles/decays can be specified
- HiggsBounds for model-independent limits on Higgs bosons from Tevatron and LEP

Very Short Example Input File

```
################################################
###           Fittino input file           ###
################################################
massh0                    115.237 GeV +- 0.05 GeV +- 0.5 GeV # comment
massTop                   178.0   GeV +- 0.3  GeV
correlationCoefficient    massh0 massTop 0.05                # quatsch
# etc
edge  1 massNeutralino1 massNeutralino2      263.5 GeV +- 1.2 GeV alias 1
sigma ( ee -> Neutralino1 Neutralino2, 1000.,0.8,0.6 ) 7.678 fb +- 2.0 fb
BR ( h0 -> Bottom Bottom~ )   0.8033 +- 0.01 +-(lumiErr) 0.05
BR ( h0 -> Charm Charm~   )   0.05   +- 0.02 +-(lumiErr) 0.01
nofit cos2PhiL                                       0.62865 +- 0.0005
# etc
fitParameter  TanBeta           10.0
fixParameter  Mu                358.6 GeV
universality  MSelectronR MSmuR
# etc
LoopCorrections       on
CalcPullDist          off
```

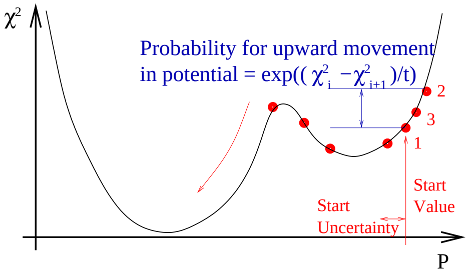
Technologies

- To find the χ^2 minimum
  - Tree-level estimate + MIGRAD in MINUIT:
    fast but unreliable, if not started very close to the true minimum
  - Tree-level estimate + Simulated Annealing:
    not too slow, very reliable detection of the global minimum
- To map the parameter space
  - 1D or 2D scans:
    fast, but no correct treatment of the correlations to fixed parameters
  - Markov Chain:
    n-dim. Probability map of the available parameter space
- To determine the uncertainties
  - MINOS in MINUIT:
    Slow and not very reliable
  - Automatic generation of pull fits:
    Very reliable and (at least on a farm) not slower than MINOS
- To get a feeling for the parameters and observables
  - Visualize the effect of variations of each parameter on observable χ^2's

Markov Chains

- Build a chain of parameter points $\vec{x}_i$ using tMetropolis-Hastings
- Calculate a likelihood $\mathcal{L}(\vec{x}_i)$ for each point. Here: $\mathcal{L}(\vec{x}_i) = e^{-\chi^2}$
- Randomly pick a new point $\vec{x}_{i+1}$ near $\vec{x}_i$, using a proposal distribution $Q(\vec{x}_{i+1};\vec{x}_i)$
- Calculate $\rho = \mathcal{L}(\vec{x}_{i+1})/\mathcal{L}(\vec{x}_i)$ for symmetrical $Q(\vec{x}_{i+1};\vec{x}_i)$
- If $\rho > 1$, accept the new point and add it to the end of the chain
- If $\rho < 1$, accept it with probability ρ, else, add $\vec{x}_i$ again to the chain
- For the efficiency of the algorithm: Optimize $Q(\vec{x}_{i+1};\vec{x}_i)$, e.g. gaussian distribution around $\vec{x}_i$ with widthes $\vec{\sigma}_i$
- Bayesian: Point density is proportional to the probability distribution
- Frequentist: Look for minimal χ^2 and $\Delta\chi^2 < 1, 4, 5.99, \ldots$
- Extremely effective sampling: needed number of steps scales with the number of dimensions D instead of n^D

Simulated Annealing



- Works as Markov Chains, but with different potential (Boltzmann) and with reducing the temperature with time
- Aim is not sampling of the parameter space, but reliable convergence towards the global minimum
- Use this together with Toy Fits: If theory code is stable (not always the case), this is free of any asumptions and renders all (non-linear) correlations and all uncertainties (without assumptions on gaussianities or else)

1. What Fittino is
2. What do we have?
   - Theory
   - Observables
   - Technologies
3. What's missing?

What's missing?

- Observables:
  - Shapes. maybe not necessary at LHC, but who knows
  - A general framework to include asymmetries
  - Systematic Uncertainties: Other shapes than gaussian?
- Models:
  - CMSSM (no fundamental problem, easy to implement if available by spectrum calculator etc)
  - Complex MSSM (too far away?)
  - R-parity violating models
  - No fundamental develpment necessary for the above, just need/use theory codes
  - More problematic: SM alone → no SLHA!
  - Everything beyond SUSY is difficult simply because of the lack of SLHA. Need an input framework capable of handling SLHA and other formats?
- Minimization:
  - Quite well equipped, but other developments would be great to test, mainly for potential speedup, such as
  - Genetic Algorithms, Swarms, ...