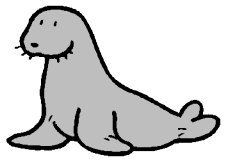
THE MINUIT PROJECT

" object-oriented re-implementation of MINUIT in C++"



Status report

- **Motivation**
  - **why, what, who, how, ...**
- **State**
  - **available prototypes**
  - **technology base**
- **Outlook**
  - **goals, improvments**

Previous presentation:

November 27, 2002
by Fred James
wenaus.home.cern.ch/wenaus/peb-app/

Note: MathLib project became part of SEAL!

Preliminary version released with SEAL_0_2_0

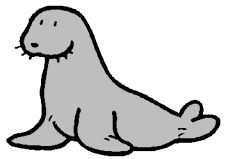
Why re-engineer MINUIT?

- **Why gsl, nag-c etc. do not do the job:**
  - **errors:**
    - **individual parameter errors**
    - **correlations (error matrix V)**
    - **" minos" non-linear errors**
    - **contours (parameter space / corrected for correlations)**
    - **change of V if one parameter is fixed**
  - **robustness:** bounds on parameters
  - **stopping criterion:** minimize number of function evaluations
  - **not object-oriented**

difficult to find
unique to MINUIT

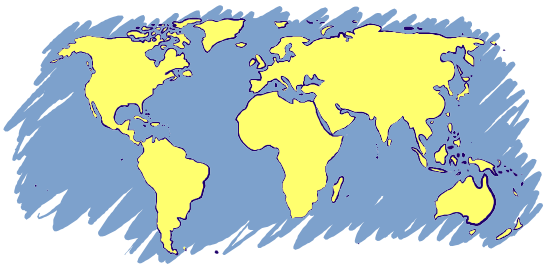
THE MINUIT PROJECT

" object-oriented re-implementation of MINUIT in C++"



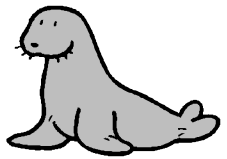
MINUIT (Fortran version):

- **physics analysis tool for function minimization (χ^2, log-likel., user-def.)**
  - **framework for set of tools for minimization**
- **© by Fred James, written in Fortran**
- **contains a collection of tools for minimization and error analysis:**
  - **migrad, simplex, scan, seek**
  - **minos (asym. errors), hesse**
- **used from main-program, interactively, in batch mode, inside PAW, ROOT**
- **used also outside CERN/HEP**



THE MINUIT PROJECT

" object-oriented re-implementation of MINUIT in C++"



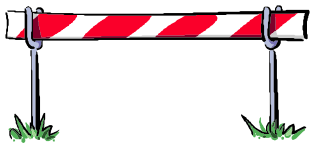
What do we expect from new one?

- **making it modern:** lifetime of MINUIT long (~25 years) w.r. to computers/languages
  - **better maintainability**
- **choosing OO-technology:**
  - **profit from increased flexibility and functionality**
  - **make it extendable (recursiveness)**
- **make it open for new algorithms:**
  - **FUMILI (F(α) = $\Sigma_i \chi_i^2(\alpha)$), ...**

What MINUIT did not do, does not do and will never do:

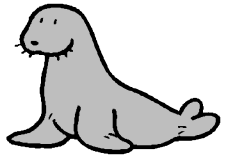
histogramming, data handling, graphics, ...

(keep it low-level, but best of all)

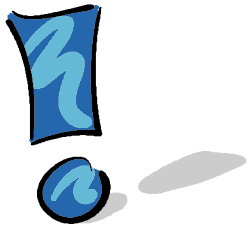


THE MINUIT PROJECT

" object-oriented re-implementation of MINUIT in C++"



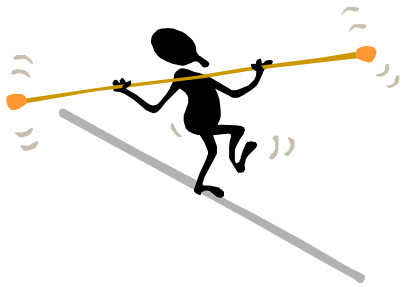
The Constraints:



- **support existing usages (main program, interactive, etc)**
- **for the "old" physicists: provide interface to Fortran function ("FCN")**
  - **minimize changes in the end-user interface**
- **for the young ones: create an C++ API (allows Python, Java, Cint, ... interface)**
- **fully independent version which runs stand-alone (users outside HEP)**
- **usable from ROOT, HippoDraw, etc ...**

In general: Get the API correct ...

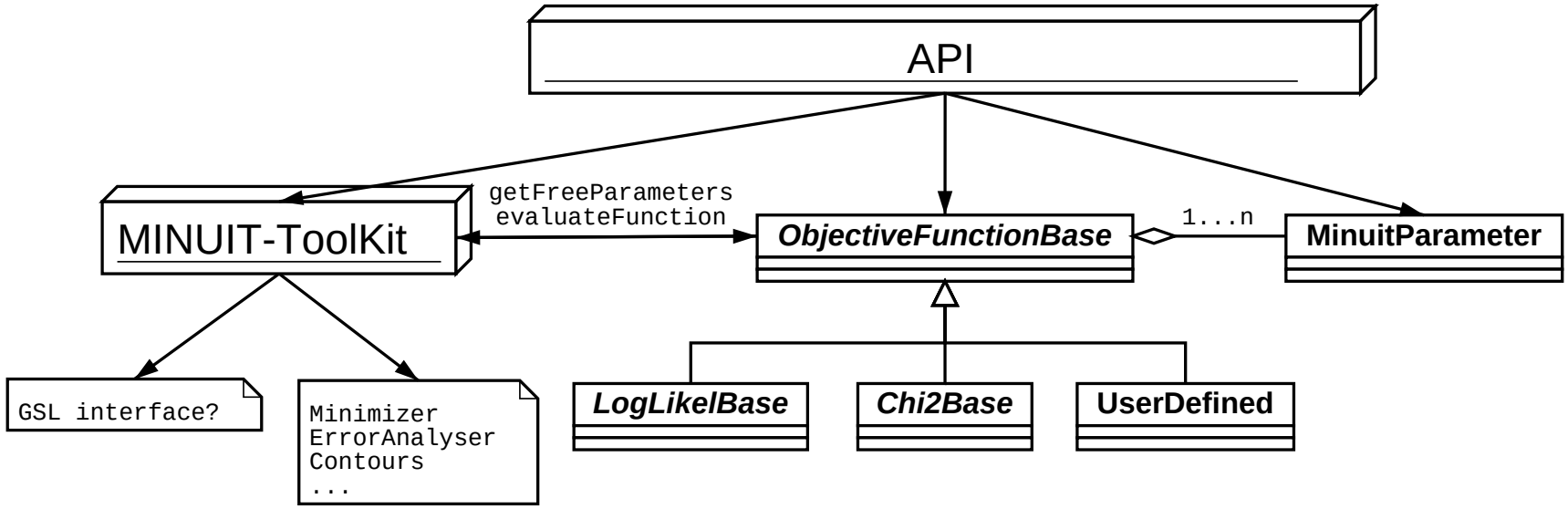
(absolut minimal required interface)



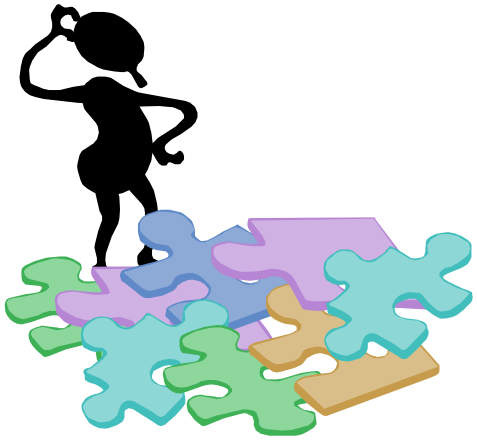
THE MINUIT PROJECT

" object-oriented re-implementation of MINUIT in C++"

SEAL

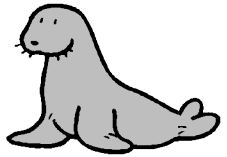


Design in evolution...

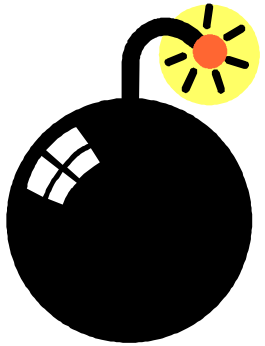


THE MINUIT PROJECT

" object-oriented re-implementation of MINUIT in C++"



Risks in undertaking such a project:



- **technical risks: wrong choice of technology**
  - **MINUIT becomes too slow due to C++**
  - **qualitative degradation w.r. to Fortran version**
- **building the wrong thing**
  - **take care of the user interfaces (human interaction)**
  - **incompatible requirements**

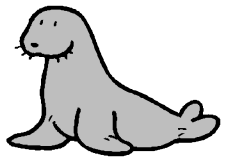
In general: We have to meet the physicists expectations in order to succeed.

The Strategy:

- **regular meetings between Fred and myself:**
  - **twice per week**
- **meetings "on demand" with experts and users:**
  - **Pere, Vincenzo, Rene, Andreas, Paul, Teddy...**
  - **people from LEP analysis groups**
- **limit number of new components:**
  - **only if current methods work properly, implement new functionality**

THE MINUIT PROJECT

" object-oriented re-implementation of MINUIT in C++"

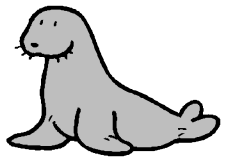


The Status:

- **prototypes:**
  - **" migrad" (Variable Metric Methods): Jan 28, 2003**
  - **" minos" (asym. errors): Mar 6, 2003**
- version for development **scram-ified**
- available from **cvs repository**
- **use gsl-wrapper for linear algebra**

THE MINUIT PROJECT

" object-oriented re-implementation of MINUIT in C++"



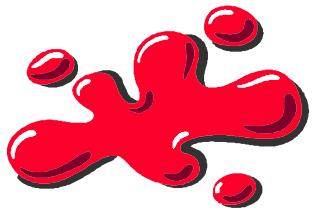
SEAL

```
MINUIT TASK: Time Distribution of Leptonic K0 Decays

FCN=   33.43457     FROM MINOS     STATUS=SUCCESSFUL   473 CALLS      555 TOTAL
                    EDM=  0.32E-06    STRATEGY= 1      ERROR MATRIX ACCURATE

 EXT PARAMETER                    PARABOLIC         MINOS ERRORS
 NO.   NAME         VALUE            ERROR       NEGATIVE       POSITIVE
  1    Re(X)       0.55653E-01   0.88802E-01  -0.73139E-01    0.94472E-01
  2    Im(X)       0.27634E-01   0.12524      -0.17853        0.12326
  5    Delta M     0.32672       0.24295      -0.85932        0.20588
 10    T Kshort    0.89200       constant
 11    T Klong      518.30       constant
ENTER MINUIT COMMAND:
```

Fortran



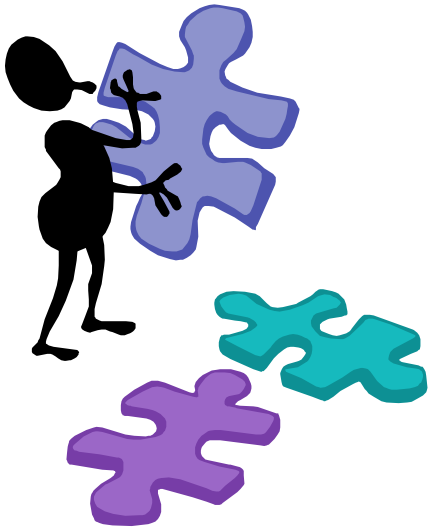
- **good agreement for prototype**
- **no detailed comparison**

New

```
minimum function value: 33.4346
minimum edm: 2.73772e-09
....
....
start MINOS
print MINOS result:
# 0 Re(X)       value: 0.0556273 parabol: 0.0862927 low: -0.0731132 up: 0.0944975
# 1 Im(X)       value: 0.0276876 parabol: 0.124027 low: -0.176853 up: 0.123211
# 4 Delta M     value: 0.326752 parabol: 0.243421 low: -0.840622 up: 0.205851
```

At the moment:

- **discuss the C++ API:**
  - **parameters**
  - **function ("FCN")**
  - **MINUIT:**
    - numerics (migrad, minos, etc.)
    - parameters (fix, release, set, etc.)
    - control (tolerance, iterations, precision, etc)
- **integration with SPI:**
  - moving to **SEAL CVS** repository (done)
  - adding MathLib to **Savannah**
  - documentation using **Doxygen**
  - configuration for development: **SCRAM** (done)
  - **testing,** using **Oval** and **CppUnit**
  - **...**



The Road-Map:

- **Version 0** (absolute minimal functionality of OO-MINUIT)**:**
  - **fully independent** as Fortran version
  - **same functionality** as Fortran version
  - **similar performance** as Fortran version
  - **same numerical quality** as Fortran version
  - **C++ API**
- **future versions:**
  - **single-sided bounds of parameters**
  - **correlated multi-parameter limits/constraints**
  - **FUMILI**
  - **recursivness**
  - **API interfaced via Python, Java, Cint (MINUIT-server?)**
  - **new minimization methods**
  - **...**
- **improve testing**
  - fully controlled **simulation** environment

What is available when?

- **by end of June 2003:**
  - **C++ API to FCN, parameters, Migrad, Minos**
  - **Migrad, Minos at same numerical quality as Fortran**
  - **usable from main-program**
  - **fully integrated into SPI and SEAL**

Summary:

- **no equivalent minimization package w.r. to MINUIT**
  - **what physicists want (errors etc.)**
- **first protoypes:**
  - **learn** from it
  - think about **C++ API**
  - come up with an **overall design**
- **define next steps**
  - **functionality vs. release**