Math Plans for 2010

- Plans for next production release 5.28
- Priority on stability and finalizing the current packages
  - need to cope with the reduction in man power
- Focus on test, validation and improvement of performances
- Some new features expected mainly in histogram library
  - efficiency class and kernel density estimator
- Working plans for RooFit and TMVA will be presented separately

Histogram Library

- Histogram efficiency class (*THEfficiency*)
  - new keep per bin the events passing a cut and the total (n1 and n2)
    - *THEfficiency::Fill(double x, bool selection);*
  - Implement different statistical methods for error calculations of binomial interval
    - see for example http://en.wikipedia.org/wiki/Binomial_proportion_confidence_interval
    - normal approximation (Wald interval, current one in *TH1::Divide*) not recommended
    - frequentist Clopper-Pearson interval (probably the default)
    - Bayesian interval
  - Errors are asymmetric, need to store lower and upper error
    - it would be nice to have also in TH1 possibility to have asymmetric error (e.g. Poisson errors)
  - *THEfficiency::Fit* will use the correct maximum likelihood method
    - no need to have anymore the *TBinomialEfficiencyFitter* class
  - User deal with only one object (for I/O or when merging results)
    - no need to create and manage two histogram classes
  - Plan to have the class ready for June release

Histogram Library (2)

- New kernel density estimation classes (in one or multi-dimensions)
    - Roofit and TMVA have already these classes but they will be useful in general
    - Bartolomeu started working on this, should be available for next release
- TF1 improvements:
    - have functions creating cdf functions, derivatives and normalizing the TF1
        - normalization will be needed for un-binned fits
    - implement possibility for analytical calculations of derivatives and integrals
        - need to store in the TF1 a list of functions which will be used in the *TF1::Integral* and *TF1::Derivative* methods when they are available
    - add possibility to change numerical algorithms via a static method (for integration, root-finding, etc..)
- TFormula improvements
    - when LLVM will be available one should be able to use it for parsing the expression
    - extend TFormula to use RooFit factory syntax ?

      example: *"gaus"* written as *"Gaussian:g(x[-10,10],mu[0,0,10],s[1,0,10])"*

Other Improvements

- Mathcore:
  - add an interface for generating random numbers for arbitrary distributions
    - use UNURAN or FOAM with the plug-in manager
  - can be used from Roofit (it will reduce its dependency)
- Minimization:
  - improve debugging in Minuit2
  - include in trunk the genetic minimizer available based on TMVA (available already in dev branch)
  - investigate new minimization algorithms (e.g. LVMIN from Blobel)
- MathMore (low priority)
  - wavelet transforms
  - multi-dimensional root-finders
  - quasi-random numbers
- Bayesian tools
  - evaluate Bayesian tool kit (BAT) to have it eventually distributed with ROOT
  - can be useful for RooStats
- Fit Panel GUI
  - add interface to Roofit to use RooFit pdf when fitting (work started by David)
- Vc (parallel vector based on intrinsic)
  - evaluate inclusion in ROOT (works only with g++ >= 4.4)
  - could be used together with SMatrix and GenVector's classes (as template arguments)

Tests and Validation

- Math
  - bring in stress suite or in rootests many of the tests available currently in the math directory (mathcore/test, mathmore/test, smatrix/test, etc...)
  - add a new stress test for TF1
  - add a new stress test for random number generation (using Unuran, Foam, etc..)
- RooStats
  - add stress test suite
  - perform validation studies with more realistic examples from experiments
    - use old data (LEP or Tevatron) ?
  - work on improving performances
    - some of the work will be mainly in RooFit
      - more efficiency usage of data sets (not always going through a TTree), faster toy generation, etc...
      - faster pdf evaluation
  - add support for parallelization