Utiliser une bibliothèque

Beaucoup ne connaissent pas les libs de calculs. Elles sont nombreuses et pas forcement accessible pour les débutants. Le but ici est de présenter une méthode pratique et empirique, pour que l'utilisateur lambda puisse tester différentes méthodes et choisir facilement celle qui sera le plus adapté. Cette approche ne remplacera pas un expert mais pourra être utilisé en première intention.

comment tester les performances

- boost.timer
- faire des séries suffisament longue
- faire plusieurs séries
- faire varier les paramètres (taille des données)
- génération des données de tests (données identiques, données triées, données aléatoires)

Vue d'ensemble des libs

- BLAS : http://en.wikipedia.org/wiki/Basic_Linear_Algebra_Subprograms. plusieurs implémentation (boost, cuda)
- PETSc, ATLAS, BLAS, FLAME, LAPACK, ScaLAPACK, Trilinos, (P)ARPACK, etc.)
- GSL (GNU Scientific Library) :
- nt2
- boost.simd

- LAPACK (Linear Algebra PACKage) : http://en.wikipedia.org/wiki/LAPACK
- Blitz++ : http://en.wikipedia.org/wiki/Blitz%2B%2B
- Armadillo (C++ library)

- boost.math
- boost.uBlas : http://www.boost.org/doc/libs/1_48_0/libs/numeric/ublas/doc/index.htm

- CUDA SDK linear algebra : http://www.nvidia.com/content/cudazone/cuda_sdk/Linear_Algebra.html
- cuBlas : http://developer.nvidia.com/cublas
- MAGMA (Matrix Algebra on GPU and Multicore Architectures) : http://icl.cs.utk.edu/magma/
- cula tools (cuda linear algebra) : http://www.culatools.com/
- GPU-Accelerated Libraries : http://developer.nvidia.com/gpu-accelerated-libraries

- OpenCL Blas : http://sourceforge.net/projects/openclblas/
- calcul matriciel