Subject: Re: Large array memory problem.
Posted by Marius Hagen on Wed, 23 Nov 2005 20:43:08 GMT
View Forum Message <> Reply to Message

Carolina wrote:
> Does anyone know a clever trick to compute inverse of large sparse
> matrices?

While I'm no expert on the subject, I do spend a lot of time working with large matrices, so maybe I can say a few things that might help here. In most instances getting the inverse of a large matrix is a *bad idea*, since, in floating-point math, you can easily wind up amplifying very small errors in the input into HUGE errors in the output. For an array with 10^8 elements, even in double-precision, most direct matrix inverses are so corrupted with these amplified errors as to be almost useless. Instead, what most people try is to use either a direct solver or an indirect solver of the *forward* matrix problem.

That is, if you are trying to calculate Ax = b (A=huge matrix, x and b long vectors), then rather than solving this as x = A^-1 b, you can use QR decomposition to change the form of A such that it is easy to solve by back-substitution. This can be rather slow, so an alternative is to obtain an approximate solution for x by use of an iterative solver. These are much faster, and typically they can obtain as much accuracy as you want in a fraction of the time that a direct solver requires. And these techniques do not have the extreme amplification of small errors.

Having said that, my own work is entirely in matrices which do not have an inverse at all, and which have 10^15 elements or so, and so iterative solvers are my only choice. Choosing which algorithm to use here is an art --- generally each application wants something different. But if you have access to Matlab, its support for sparse matrix routines and for direct or indirect matrix solvers is impressive.

- Marius