Subject: Re: [DISCUSSION] Add 'complex' datatype, to Value too
Posted by kohait00 on Mon, 20 Jun 2011 07:36:36 GMT
View Forum Message <> Reply to Message

Quote:
I know it is possible in most cases... But the wrapper has more pros that I forgot to mention. E.g. it is a clean way to "move" complex to Upp namespace... I am too lazy person to type "using std::complex;" every time I want to use it (And hardcoding the using clause into U++ doesn't feel right...)

this is again a base principles decision. why not thinking of complex as 'just another logical type like int or double'. this would relief the namespace boundaries..

i'm not quite familiar with the std::complex implementation, which states to be optimized (through template specialization) for double, int and float..

but as mirek said, i dont mind to use std::complex at all.

again, to make std::complex usable in Value, we'd need to have those (or some related) changes in SetNull behaviour or extend Nuller (which probably is a lot esier.

as of fft: doubtlessly fftw is the fastest implementation and the most versatile. but the api sucks definitely. so it'd be a must to have it as plugin to be able to expose it's c api as C++, thus avoiding LGPL issues. ofcorse static linkage isnt possible then.

EDIT: in case std::complex<>: to write complex<double> each time is cumbersome. a \typedef complex<double> cdouble' and respectively for the others would be good..
typedef

EDIT: no need to derive from complex to pull it into upp namespace. just place the 'typedef std::complex<double> cdouble' in the upp namespace..cdouble is then Upp::cdouble. just tested it.