Re: A question on Newton's Method

Source: http://sci.tech−archive.net/Archive/sci.math.num−analysis/2005−04/msg00063.html

- *From*: Jon Harrop <usenet@xxxxxxxxxxxxx>
- *Date*: Mon, 04 Apr 2005 01:55:28 +0100

James Van Buskirk wrote:
> "Jon Harrop" <usenet@xxxxxxxxxxxxx> wrote in message
> news:42503d87$0$42333$ed2619ec@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx
>> Really? Have you considered the Fortran equivalent of my numerical
> analysis
>> example? How about some other examples? Can you design a single example
>> of numerical analysis which is more comprehensible in Fortran?
>
> Please submit your equivalent of
>
> http://home.comcast.net/~kmbtib/conv2b.f90
>
> in the non−Fortran language of your choice!

That is a 2,199−line Fortran program. I believe the equivalent in
Mathematica is the built−in ListConvolve function which is 12 letters of
code. Of course, the Mathematica implementation is much more powerful as it
works for any "n".

An OCaml implementation would use FFTW and would probably be <100 lines and
also be more powerful than the Fortran you've given.

So the Fortran is several orders of magnitude more verbose than it need be.
That's hardly what I call a comprehensible example. Any other examples of
numerical analysis code which is actually simpler in Fortran?

>> Plot[f'[x], {x, −1, 2}]
>
> No, I don't want this! In ordinary everyday mathematical usage,
> you can define a new function to be the derivative of an old
> function. For example, psi(x) = (d/dx)(ln(Gamma(x))). We want
> to be able to do this in Mathematica; I don't want to keep
> referring to the original function (as in your f'[x] above.)

If you want to explicitly define a function, say "df", as the derivative of
"f" then just do it the obvious way instead of messing about with delayed
evaluation, the "D" function and unnecessarily−reused definitions:

df[x_] = f'[x]

or even:

df = f'

> So the origi