Overview of Nashlib and its Implementations

John C Nash, retired professor, University of Ottawa Peter Olsen, retired ??

11/01/2021

Abstract

The repository https://github.com/pcolsen/Nash-Compact-Numerical-Methods is a collection of implementations of the algorithms from Nash (1979), which was written by one of us (JN) in the mid 1970s. The present collection originated with a query from the other author (PO) concerning a possible Python implementation based on the Pascal codes in the second edition of the original book (Nash (1990)).

From email interchanges, the repository and the idea of gathering different implementations and ensuring workable example codes arose. We hope that these will be useful in various ways:

- as a focus for comparison of programming languages and coding styles for numerical computations
- as a source of didactic examples and exercises for students
- as a resource for workers needing to embed numerical computations within their application programs.

History

?? give a history of the codes

Documentation of this project

Github allows for Markdown (http://www.aaronsw.com/weblog/001189 and https://daringfireball.net/projects/markdown/) files to be included in the file repository. We have used this for the `README.md` file and some other introductory or indexing documentation. There is also a wiki associated with each project. We have NOT used the wiki in favour of including a `Documentation` directory within the file repository so that the documentation is carried along with any clone of the repository or download as a Zip archive.

A relatively well-developed resource for documentation is the `knitr` set of tools (Xie (2013)). vastly expanded within the RStudio computational ecosystem. See https://rstudio.com/products/rstudio/. In particular, the **Rmarkdown** flavour of the **markdown** markup language allows for LaTeX mathematical expressions to be included inline, BibTeX references to be cited easily, and code in a number of programming languages to be executed, with the results automatically included in the report. Reports can be output in a variety of ways such as HTML, PDF (our choice), or even as Microsoft Word documents.

Programming language issues in Nashlib implementations

Fortran

BASIC

Pascal

Python

R

Others

Running codes in different programming languages

This part of the Overview document is likely to evolve as we and other workers apply our ideas to different computing platforms. We welcome input and collaboration.

Our principal objective is to be able to take a program in one of the selected programming languages (or dialects) and have it execute correctly and produce acceptable output. In the subsections below, we outline how we do this, possibly with variations for different operating environments.

Fortran

Rmarkdown is able to compiler Fortran subroutines and then call them from R. We leave out the question of timing for which this simple example was devised.

```
C Fortran test
      subroutine fexp(n, x)
      double precision x
C  output
      integer n, i
C  input value
      do 10 i=1,n
         x=dexp(dcos(dsin(dble(float(i)))))
  10  continue
      return
      end
```

Now try running it from R.

```
res = .Fortran("fexp", n=100000L, x=0)
str(res)
## List of 2
##  $ n: int 100000
##  $ x: num 2.72
```

What about complete programs? In this case we need to work through the operating system. For JN this is a Linux Mint MATE distribution.

The following is file `knitrExample.f`. It is in the directory `../fortran/` relative to this documentation file (`NashlibOverview.Rmd`). See the source of this document for the mechanism by which the Fortran code is read into our discussion text.

```
C  Test a computation
       do 10, i=1,100000
        y=exp(sin(cos(dble(i))))
10     continue
```

```
      write(*,100) y
 100  format('0 last value of y = ',1pe16.8)
      end
```

We run the code with a command line script (**bash** script).

```bash
#!/bin/bash
gfortran ../fortran/knitrExample.f
./a.out
```

```
## 0 last value of y =   4.31224912E-01
```

BASIC

Pascal

Python

R

Others

References

Nash, J. C. 1979. *Compact Numerical Methods for Computers : Linear Algebra and Function Minimisation.* Book. Hilger: Bristol.

Nash, John C. 1990. *Compact Numerical Methods for Computers : Linear Algebra and Function Minimisation, Second Edition.* Book. Institute of Physics : Bristol.

Xie, Y. 2013. *Dynamic Documents with R and Knitr.* Chapman & Hall/Crc the R Series. Taylor & Francis. https://books.google.ca/books?id=QZwAAAAAQBAJ.