Project: Numerical Integration with Multi-Language Experience

Instructor: Seongjai Kim*

1 Introduction

In the project, we will study some numerical techniques for the numerical integration for given realistic data; in particular, we will consider the Trapezode and Simpson's quadrature rules. For the project, you will implement computational routines for the algorithms and compare the numerical solutions for various data. Even though MATLAB is a convenient computational language, we will try to utilize computational languages such as C++, C, and FORTRAN. In fact, we will experience the three languages in a program; the purpose of such a multi-language implementation is for you to be better prepared for both the computational languages and very heterogeneous simulation environments in industry. I know some of you may feel uncomfortable in the beginning; however, you will feel okay soon. Do not hesitate for questions and suggestions.

2 The objects

For the project, you will use the modelcode provided:

1. Download the modelcode, `cstart` and `mkmk`:
   `www.ms.uky.edu/~skim/GRADE/TAR/Numer_Integration.CF.tar`
   `www.ms.uky.edu/~skim/Shell/cstart+mkmk.v1.tar`
2. Untar them in a **Unix** machine, using e.g.
   `<tar xvf Numer_Integration.CF.tar>`

If you are interested in, try to visit www.modelcode.net, where various modelcodes are posted for education and research purposes.

*Department of Mathematics, University of Kentucky, Lexington, Kentucky 40506-0027 USA
Email: skim@ms.uky.edu Phone: 859-257-8681

Here is your job for the project:

- Implement routines for the Trapezoid and Simpson's rules for numerical integration, one in C and the other in FORTRAN (f77, f90, or f95), and save them into separate files.
- For Simpson's rules, you have to combine two different Simpson's rules to incorporate general numbers of subintervals (`nx-1` in the modelcode).
- Call the routines from `integration.f`, a F77 subroutine in the modelcode `Numer_Integration.CF`. (You can use <`mkmk`> to update `Makefile` automatically. `Makefile` is the source for the Unix command <`make`>.)
- Test if your routines are correct using the given data, `DATA1`, which is made out of $y = x^2$.
- Compose your own data, having at least 10 points, and test the routine for for data. (You may use `cstart` to update `ReadDATA.c` automatically; try <`cstart SET_file`>.)
- You can discuss the problem and difficulties with your classmates. However, you cannot share the data; make your own unique data.
- Your report must include hard copies of
  1. a one-page note (expressing your experiments),
  2. `integration.f` (revised),
  3. the two routines for the Trapezoid and Simpson's rules,
  4. your data, and
  5. numerical results from the two data sets

The modelcode is written in a combination of C++, C, and F77. FORTRAN shows advantages in scientific computation. In particular, if you are dealing with complex-valued solutions, FORTRAN is the language. Also we know that C/C++ can easily incorporate other libraries, packages, and GUIs; they are so powerful! I hope you can get an idea for modern high-performance computation environments, through the project. I believe such an experience of multi-language computation will help your future research and job search.

Makefile: Multi-Language computation is not easy for some cases. However, I believe there is always a way to link objects constructed from different compilers. I assume the computer you are using is installed with C++, C, and F77. (You can find such a computer in campus. If it is not true for you, please contact me as soon as possible.) Once the compiling is finished, the next problem is linking. The makefile generator (`mkmk`) will produce `Makefile` to link all. If the linking is not successful, email me the result of the Unix command <`echo $HOST $OSTYPE`>

Good Luck!!