Umeå University
Department of Computing Science

Fall 2018

Project 3

Error estimation for artillery computations

Scientific Computing

The deadline for this project can be found at:
http://www8.cs.umu.se/kurser/5DV005/HT18/planering.html
(Link *Overview* on the course homepage.)

- The submission should consist of:
  - The complete report, including
    - A front page with the following information:
      1. Your **name**.
      2. The **course name**.
      3. Your **username** at the Department of Computing Science.
      4. The **project number**.
      5. The **version** of the submission (in case of re-submissions).
  - An appendix with the source code.
  - To simplify feedback, the main report (optionally excluding the appendix) must have **numbered sections** and **page numbers**.
- The submitted code must be `MATLAB`-compatible. If you choose to work in Octave, verify that your code is `MATLAB`-compatible before you submit your project.
- If you write your report using LaTeX, double-check that your references have been resolved correctly before you submit. "Figure ??" is useless to any reader.
- Your report should be submitted as a pdf file uploaded via the https://www8.cs.umu.se/~labres/py/handin.cgi page, also available as the

  Submit/Check results

  link at the bottom left of the course home page.
- Furthermore, the source code should be available in a folder called `edu/5dv005/assN` in your home folder, where N is the project number. You will probably have to create the folder yourself.

Accuracy in artillery computations

Carl Christian Kjelgaard Mikkelsen

December 14, 2018

Contents

1 Primary purpose

The primary purpose of this assignment to develop the ability to compute error estimates which are reliable and accurate.

2 Asymptotic error expansions

We have considered the problem of approximating the range of a gun, the flight time of a shell or the elevation necessary to hit a particular target. It remains to compute reliable error estimates for such approximations.

In this project, we view each approximation A as a function of the size of the time step h used when computing the trajectories, i.e., $A = A_h$. We will investigate, if there exists asymptotic error expansions of the form

$$T - A_h = \alpha h^p + \beta h^q + O(h^r), \quad 0 < p < q < r \tag{1}$$

The term αh^p is called the primary error term, while βh^q is the secondary order term. We will obtain reliable error estimates.

The asymptotic error expansion describes the difference between the target value T and the approximation A_h. We can not hope to obtain the exact value of A_h. The very best we can hope for is the floating point representation of A_h, i.e. $\hat{A}_h = \mathrm{fl}(A_h)$, but in general this is not a realistic goal. The difference between A_h and the computed value of $\hat{A}_h$ is the result of many rounding errors. By monitoring the *computed* value of Richardson's fraction we can determine when the rounding error $A_h - \hat{A}_h$ is irrelevant compared with the error $T - A_h$.

k	Approximation A_h	Fraction F_h	Error estimate E_h
1	2.895480163672e+00	0.00000000	0.000000000000e+00
2	2.805025851403e+00	0.00000000	-9.045431226844e-02
3	2.761200888902e+00	2.06399064	-4.382496250165e-02
4	2.739629445828e+00	2.03161941	-2.157144307422e-02
5	2.728927822736e+00	2.01571695	-1.070162309156e-02
6	2.723597892360e+00	2.00783544	-5.329930376490e-03
7	2.720938129638e+00	2.00391198	-2.659762721350e-03
8	2.719609546672e+00	2.00195456	-1.328582965698e-03
9	2.718945579511e+00	2.00097692	-6.639671619268e-04
10	2.718613676976e+00	2.00048837	-3.319025345263e-04
11	2.718447745963e+00	2.00024413	-1.659310128161e-04
12	2.718364785520e+00	2.00012206	-8.296044325107e-05
13	2.718323306559e+00	2.00006078	-4.147896106588e-05
14	2.718302567373e+00	2.00002842	-2.073918585666e-05
15	2.718292197853e+00	2.00001403	-1.036952016875e-05
16	2.718287013122e+00	2.00001123	-5.184730980545e-06
17	2.718284420669e+00	1.99993264	-2.592452801764e-06
18	2.718283124268e+00	1.99973060	-1.296401023865e-06
19	2.718282476068e+00	2.00000000	-6.482005119324e-07
20	2.718282151967e+00	2.00000000	-3.241002559662e-07

Figure 1: The output of `a3f2.m` after the completion of `MyRichardson.m`

3 Software

The function `range_rkx` moves the shell from point to point using a time step which is fixed except for the very last time step. Here a non-linear solver is used to compute the time step which will place the shell on the ground. This equation is solved to the limit of machine precision, specifically the tolerance passed to the underlying bisection routine is $\text{tol} = 2^{-53}$. The function `range_rkx_sabotage` is identical to `range_rkx` except that the tolerance is much larger, specifically, $\text{tol} = 2^{-3}$. The function `a3int` computes the integral of a given function along a trajectory computed by `range_rkx`.

4 Questions

1. Copy `scripts/a3f1.m` in `work/MyRichardson.m` and complete the function according to the the specification. It is likely, that the function is working correctly when the corresponding minimal working example `scripts/a3f2.m` returns the output given by Figure 1.

2. Develop a script `a3range.m` which apply Richardson's techniques to compute the range of the shot whose physical parameters are given by `a3f3.m`

   - Use methods `'rk1', 'rk2', 'rk3', 'rk4'` to compute the trajectories
   - Use time steps $h_k = 2^{-k}$ seconds for $k = 0, 1, 2, \ldots$.

   For each of the four methods:

(a) Determine the power of the primary error term.

(b) Determine the power of the secondary error term.

(c) Determine when the computed value of Richardson's fractions behaving in a manner consistent with an asymptotic error expansion of the type given by equation (1).

(d) Identify the best approximation of the range and explain why the error estimate is reliable.

3. Develop a script `a3time` which compute the flight time of the shot given by `a3f3` using your method of choice.

   (a) You must include an error estimate.

   (b) You must explain why your error estimate can be trusted.

   (c) You must discuss the behavior of Richardson's fractions.

4. Develop a script `a3low` which computes the low firing solution for a target located at 15000 meters to the right of the gun given by `a3f3.m`.

   - You must include an error estimate.
   - You must explain why your error estimate can be trusted.
   - You must compute Richardson's fractions and discuss their behavior.

   Warning: Richardson's fraction will not behave correctly unless the elevations are computed with what would appear to be excessive accuracy! Expect to use residuals which are as small as 10^{-10} meters. **Warning:** This is not a fast calculation, so begin with a small number of rows, say, 6 rows, in Richardson's table and see if this is enough.

5. Develop a script `a3range_g7` which computes the range of the shot defined by `a3f4`.

   (a) For which methods do you retain the ability to estimate the range accurately?

6. Develop a script `a3range_sabotage` which uses `range_rkx_sabotage` to compute the range of the shot given by `a3f3`.

   (a) For which methods do you retain the ability to estimate the range accurately?

7. The script `a3length` computes the length of the trajectory of the shot given by `a3f3.m`.

   - It uses methods `'rk1', 'rk2', 'rk3', 'rk4'` to compute the trajectories
   - It uses time steps $h_k = 2^{-k}$ seconds for $k = 0, 1, 2, \ldots$.
   - It uses the trapezoidal rule to compute the length of each trajectory.

   (a) For each method what is the order of the primary error term?

5 Concluding remarks

Our entire analysis hinges on the existence of an asymptotic error expansion (AEE), see equation (1). It can be difficult to prove the existence of an AEE, but close observation of the Richardson's fraction will allow you to determine when Richardson's error estimate is reliable, i.e., Questions 2, 3, 4.

Mathematically, the existence of an AEE requires a certain degree of differentiability to sustain the necessary Taylor expansions. In particular, higher order methods such as 'rk3' and 'rk4' require more differentiability than lower order methods such as 'rk1' and 'rk2'. The drag function used by `a3f4` is a piecewise cubic polynomial which is two, but not three times differentiable. This significantly reduces the amount of differentiability available and causes the problems which you detected in Question 5.

Moreover, all central equations must solved accurately or you loose the ability to estimate the error accurately. The error made by `range_rkx_sabotage` when computing the final time-step destroyed you ability to estimate the difference between the true range r and the approximation r_h, see Question 6. Similarly, it was necessary to use a very small residual when computing the elevation θ_h, see Question 4.

The calculation of the length of the trajectory of the shell in Question 7 illustrates a fundamental principle of scientific computing. In general, a calculation is only as a accurate as its least accurate component. If we use a first order method to compute the trajectory, then we gain nothing from using a second order method to compute the arc length. Similarly, if we use a third or fourth order method to compute the trajectory, then a second order accurate calculation of the arc length will reduce the overall accuracy to second order.