Umeå University
Department of Computing Science

Fall 2018

Project 2

Approximation of real functions and solution of non-trivial equations

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://webapps.cs.umu.se/labresults/v2/handin.php?courseid=337 page, also available as the

  Submit/Check results

  link at the bottom left of the course home page.
- Furthermore, every scrap of MATLAB code developed by your team should be uploaded in a single zip file, `P1Code.zip`.

Approximation of real functions and solution of non-trivial equations

Carl Christian Kjelgaard Mikkelsen

December 11, 2018

Contents

List of Figures

1 Purpose

Let $I = [a, b] \subset \mathbb{R}$ and let $f : I \to \mathbb{R}$ denote a function. In this project we consider the problem of computing f and solving the equation $f(x) = 0$. Frequently, but not universally, our knowledge of f is limited to a table of function values, say,

x_0	x_1	x_2	$\dots$	x_n
$f(x_0)$	$f(x_1)$	$f(x_2)$	$\dots$	$f(x_n)$

We must construct accurate approximations of $f(t)$ for all $t \in [x_0, x_n]$ from this data. You will develop software which can compute accurate approximations and solve complicated equations accurately.

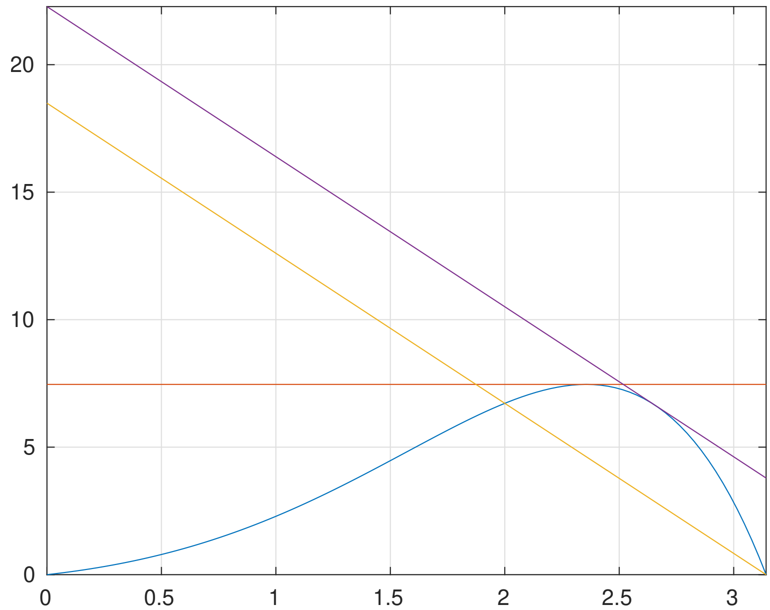

Figure 1: A illustration of Rolle's theorem and the mean value theorem of differentiation.

2 The zero theorems

Much our work hinges on a set of theorems which assert the existence of a zero:

- Robust root finding for a continuous function $f : I \to \mathbb{R}$ requires a bracket (a_0, b_0) such that $f(a_0)f(b_0) < 0$. By the intermediate value theorem there exists at least one zero $r \in (a_0, b_0)$.

- If $a_0 \in I$ and $b_0 \in I$ are zeros of a differentiable function $f : I \to \mathbb{R}$, then by Rolle's theorem. there exists c between a_0 and b_0 such that
$$f'(c) = 0.$$

- If $a_0 \in I$ and $b_0 \in I$ and $f : I \to \mathbb{R}$, is differentiable, then by the mean value theorem there exists c between a_0 and b_0 such that
$$f'(c) = \frac{f(b) - f(a)}{b - a} \tag{1}$$

Assignment 1 ruthlessly exploited the intermediate value theorem. You will now develop a scripts which illustrates both Rolle's theorem and the mean value theorem.

1. Copy `a2f1.m` into `no2/work/MyZeroTheorem.m` and complete the script according to the specifications. The script is finished when it reproduces Figure 1.

3 Approximation of derivatives

Let $f : I \to \mathbb{R}$ be a function which is infinitely often differentiable. Let $h > 0$. By Taylor's theorem we have

$$f(x+h) = f(x) + f'(x)h + \frac{1}{2}f''(x)h^2 + \dots, \frac{1}{p!}f^{(p)}(x)h^p + O(h^{p+1}), \quad h \to 0, \quad h > 0.$$

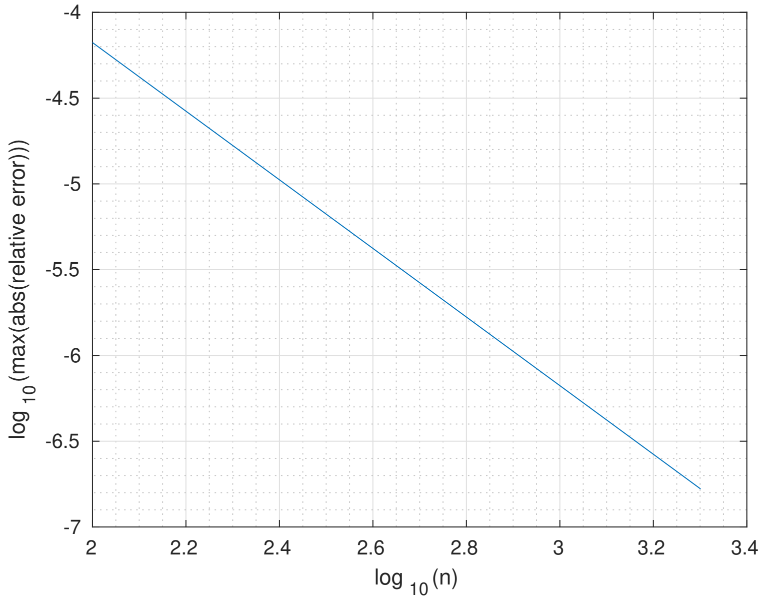

Figure 2: The error of finite difference approximations of the first derivative decay rapidly when decreasing the stepsize h between the sample points.

1. Show that

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} + O(h^2), \quad h \to 0, \quad h > 0. \tag{2}$$

2. Show that

$$f'(x) = \frac{-3f(x) + 4f(x+h) - f(x+2h)}{2h} + O(h^2), \quad h \to 0, \quad h > 0. \tag{3}$$

3. Show that

$$f'(x) = \frac{f(x-2h) - 4f(x-h) + 3f(x)}{2h} + O(h^2), \quad h \to 0, \quad h > 0. \tag{4}$$

4. Copy `a2f2.m` into `no2/work/MyDerivs.m` and complete the function according to the specification. It likely that `MyDerivs` is working correctly when the minimal working example `a2f3.m` regenerates Figure 2.

5. What evidence do you find to support the hypothesis that error committed by `MyDerivs` is $O(h^2)$ as suggested by equations (2), (3), (4)?

4 Hermite's piece-wise approximation

Consider the polynomials p_0, and p_1 given by

$$p_0(t) = (1+2t)(1-t)^2, \qquad p_1(t) = t^2(3-2t) \tag{5}$$

as well as the polynomials q_0 and q_1 given by

$$q_0(t) = t(1-t)^2, \qquad q_1(t) = t^2(t-1). \tag{6}$$

1. Show that
$$p_0(0) = 1, \quad p_0(1) = 0, \quad p_0'(0) = 0, \quad p_0'(1) = 0. \tag{7}$$

2. Show that
$$p_1(0) = 0, \quad p_1(1) = 1, \quad p_1'(0) = 0, \quad p_1'(1) = 0. \tag{8}$$

3. Show that
$$q_0(0) = 0, \quad q_0(1) = 0, \quad q_0'(0) = 1, \quad q_0'(1) = 0. \tag{9}$$

4. Show that
$$q_1(0) = 0, \quad q_1(1) = 0, \quad q_1'(0) = 0, \quad q_1'(1) = 1. \tag{10}$$

Consider a function $f : [a, b] \to \mathbb{R}$ such that f is differentiable and f' is continuous. Let $\phi : [a, b] \to \mathbb{R}$ denote the linear function which maps a into 0 and b into 1, i.e.,

$$\phi(x) = \frac{x - a}{b - a}. \tag{11}$$

Hermite's approximation of $f : [a, b] \to \mathbb{R}$ is the polynomial $p : [a, b] \to \mathbb{R}$ given by

$$p(x) = f(a)p_0(\phi(x)) + f(b)p_1(\phi(x)) + f'(a)(b - a)q_0(\phi(x)) + f'(b)(b - a)q_1(\phi(x)) \tag{12}$$

5. Show that Hermite's approximation satisfies
$$p(a) = f(a), \quad p(b) = f(b), \quad p'(a) = f'(a), \quad p'(b) = f'(b). \tag{13}$$

In short, Hermite's approximation reproduces the values of f and f' at the endpoints of the intervals. If we are given a list of points

$$a = x_0 < x_1 < x_2 < \cdots < x_n = b$$

as well as the function values $f(x_j)$ and the derivatives $f'(x_j)$, then we can approximate f using a piece-wise cubic polynomial $p : [a, b] \to \mathbb{R}$ given by

$$\forall x \in [x_{j-1}, x_j] \ : \ p(x) = p_j(x) \tag{14}$$

where p_j is Hermite's approximation of f corresponding to the sub-interval $[x_{j-1}, x_j]$. By design, the function p is differentiable and p' is continuous.

6. Copy `a2f4.m` into `no2/work/MyPiecewiseHermite` and complete the function according to the specification. It is likely that `MyPiecewiseHermite` is working correctly when the minimal working example `a2f5.m` generates Figure 3.

7. What evidence do you find to support the conjecture that the error $f(x) - p(x)$ decays as $O(h^4)$?

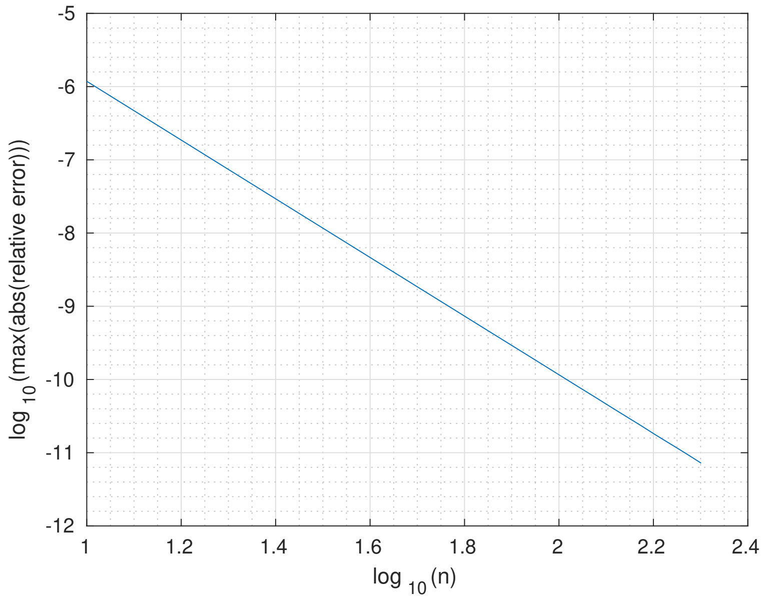

Figure 3: The error of a piecewise Hermite approximation decays rapidly when we decrease the stepsize h between the sample points.

5 Event location for ordinary differential equations

The function `range_rkx` computes the position $(x(t), y(t))$ and velocity $(x'(t), y'(t))$ of an artillery shell by solving an ordinary differential equation

$$\gamma'(t) = f(t, \gamma(t))$$

with respect to the function $t \to \gamma(t)$ which satisfies

$$\gamma(t) = (x(t), y(t), x'(t), y'(t))^T.$$

The specific function f is complicated and describes all the necessary physics. In practice, we are not particularly interested in the trajectory itself, rather we seek to solve nonlinear equation of the form

$$g(\gamma(t)) = 0,$$

where g is a function such that the composition $g \circ \gamma$ is defined. For the sake of simplicity, we assume that g is defined for all $z \in \mathbb{R}^4$, and write

$$g(z) = g(z_1, z_2, z_3, z_4).$$

The function g is called an event function and we say that an event has occurred at time t if $g(\gamma(t)) = 0$. Obviously, some events function are more interesting than others.

- The choice of $g(z) = z_2 - c$ corresponds to solving the nonlinear equation

  $$y(t) = c \tag{15}$$

  This is equivalent to finding the time t where the shell reaches the height c.

- If $x \to h(x)$ is a function which represents the height of the landscape, then the choice of $g(z) = z_2 - h(z_1)$ corresponds to solving the nonlinear equation

  $$y(t) - h(x(t)) = 0.$$

  This is equivalent to computing the time t where the shell hits the ground.

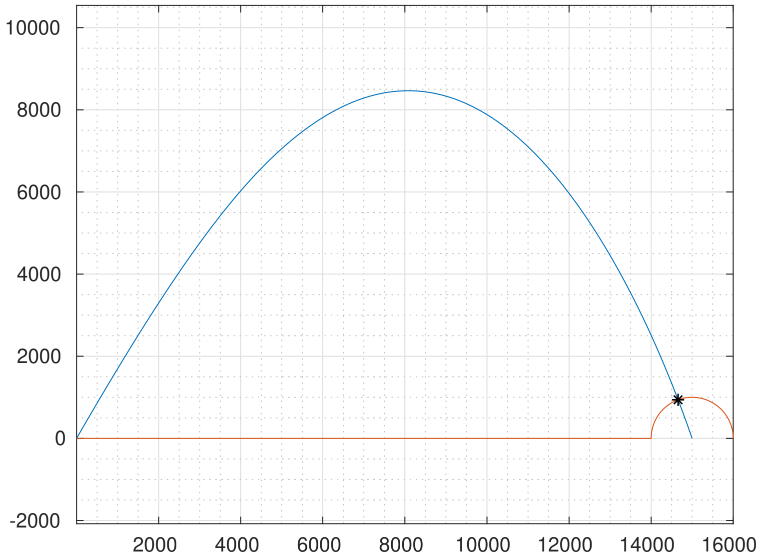

Figure 4: The point of impact between a shell and a simple landscape, i.e. a spherical hill.

By necessity, we are limited to computing a *finite* set of approximations, say, $\gamma(t_j)$ for $j = 0, 1, \ldots, m$. However, the use of Hermite's approximations allows us to extend our approximation to cover the entire interval $[t_0, t_m]$. This allows us to rapidly *approximate* the solution(s) of an event equation.

1. Copy the minimal working example `range_rkx_mwe2` into the file `no2/work/MyEvent.m`. Extend `MyEvent` to the point where it defines Hermite's approximation of both $t \to x(t)$ and $t \to y(t)$ and solves the non-linear equation $y(t) - h(x(t)) = 0$ where $x \to h(x)$ is the small hill defined by the script `a2f6.m`, see Figure 4.

Remark 1 The computed solution of an event equation depends on the size of the time step used to approximate the trajectory in the first place. Estimating the error, i.e. the difference betweeen the computed solution and the real solution is a non-trival problem. It will take us a few weeks to develop the necessary skills.