Verkfræðilegar Bestunaraðferðir

Day 2

T-423-ENOP

Arnar Gylfi Haraldsson
arnarh23@ru.is

Hafþór Árni Hermannsson
hafthorh20@ru.is

Ragnheiður Gná Gústafsdóttir
ragnheidurg@ru.is

April 30, 2024

Exercise 1

Gram-Schmidt Orthogonalization

Overview

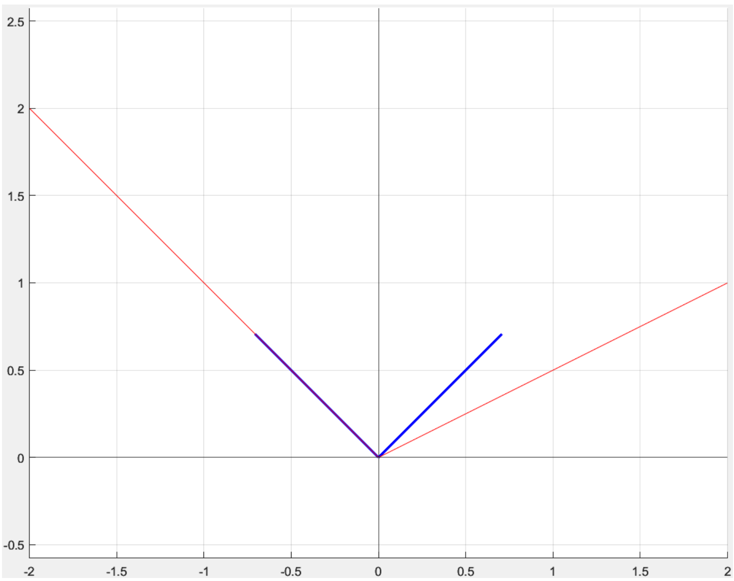
A matlab function was implemented that carries out the Gram-Schmidt orthogonalization procedure. The input is a matrix with column vectors to be orthogonalized and the function ouptputs a matrix with orthogonal column vectors, unless the column vectors are linearly dependant in which case the function returns an error message. The vectors are computed simply by applying the equations given in the problem description.

Results

Input:

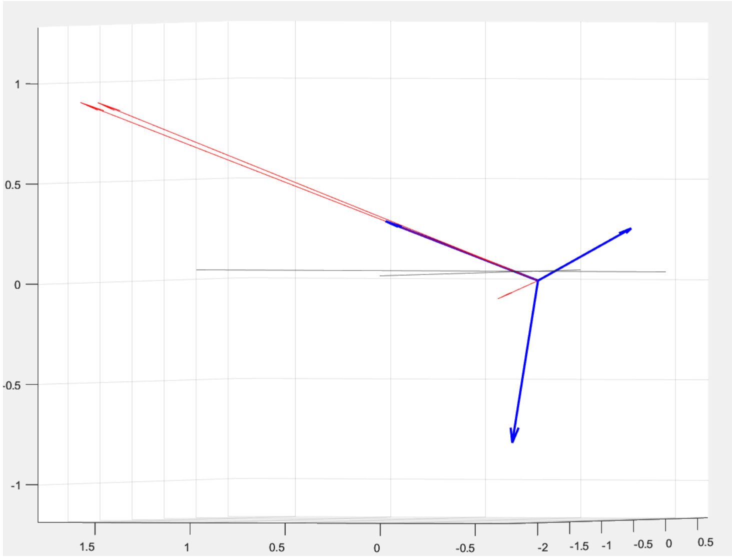
```
v1 = [-2 2]'; v2 = [2 1]'; A = [v1 v2];
GramSchimidt(A)
```

Yields 3D input




```
v1 = [-2 2 1]'; v2 = [-2 1.9 1]'; v3 = [-0.1 0.2 -0.1]'; A = [v1 v2 v3];
GramSchimidt(A)
```

Yields the following graphic



Exercise 2

Overview

A Matlab function was implemented that plots a paraboloid $x^2 + y^2$ as well as a tangent plane established at point $[0.5 - 0.5]^T$.

Functionality

In this exercise, the code provided on slide 35 in lecture 01 on day 1 was used as a base code. When the appropriate parameters were changed in that code the following calculations were made to determine z0 and zi_tangent:

$$z_0 = x_0^2 + y_0^2 \tag{1}$$

$$z_{i\_tangent} = z_0 + 1 \cdot (x_i - x_0) - 1 \cdot (y_i - y_0) \tag{2}$$

Then the plane was plotted the same way the parabola was, except with zi_tangent instead of zi and the point plotted with plot3(xi,yi,zi).

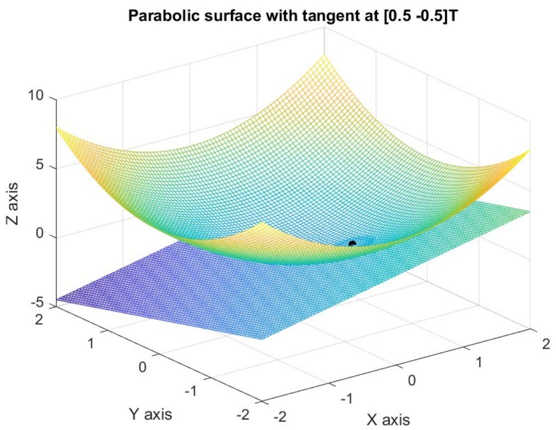


Figure 1: The parabola and the tangent plane.

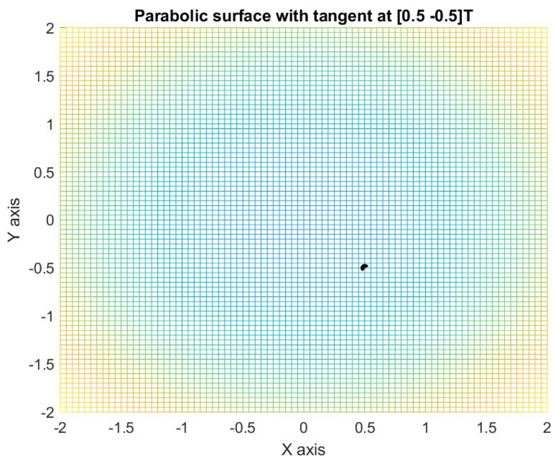


Figure 2: The position of the point on the XY-plane

Exercise 3

Overview

Here we are to write a Matlab function that finds a polynomial interpolating a given dataset that finds all local minima and maxima of this polynomial and plots both the polynomial, its first-order derivative, input data set as well as locations of the minima/maxima.

Functionality

Firstly we split the data into x-coordinates and y coordinates, the we used the polyfit() Matlab function to find the polynomial, using the x and y coordinates and specified the order of the polynomial to be the $(length(x - coordinates) - 1)$. Then we used the polyder() function to find the derivative of the polynomial, defined the x_fit and used polyval() and the x_fit to define the y_fit and the dy_fit using the corresponding polynomials.
To find where the derivative crosses over the x-axis we have to find the two indices where the values turn from positive to negative or negative to positive. This we do by multiplying all values with the next value in the dyvector. If the result is negative then we know the line crosses over the zero between these two values. With all those indices in hand we can plug them into equation (3) and get the values of x where dy_fit crosses the x-axis. Finally we find the y_minmax with polyval(pol,x_minmax). Then it is all plotted up for the results.

$$x_{\text{minmax}} = x_i + \frac{(x_{i+1} - x_i) \cdot (-y_i)}{y_{i+1} - y_i} \tag{3}$$

Testing

We used three additional data sets of varying lengths and varying points, shown below:

$$\text{data} = \begin{bmatrix} 0 & 0 \\ 2 & -1 \\ 2.8 & 5 \\ 4 & 2 \\ 5 & -1 \\ 6 & 5 \\ 7 & 8 \end{bmatrix} \quad \text{data1} = \begin{bmatrix} 0.5 & 1.5 \\ 1.0 & -2.0 \\ 1.5 & 0.5 \\ 2.0 & -3.0 \\ 2.5 & 2.5 \end{bmatrix} \quad \text{data2} = \begin{bmatrix} -1 & -3 \\ 0 & 1 \\ 1 & -2.5 \\ 2 & 5 \\ 3 & 6 \\ 4 & -4 \\ 5 & 3 \end{bmatrix} \quad \text{data3} = \begin{bmatrix} -2 & 2 \\ -1.5 & 3.5 \\ -1 & 4 \\ -0.5 & 5 \\ 0 & 5 \\ 0.5 & 4.5 \\ 1 & 4 \\ 1.5 & 3 \\ 2 & 2 \\ 2.5 & 1.5 \end{bmatrix}$$

The plotted results of feeding these data sets into our function are displayed in figures 3 to 6 and the coordinates of the maxima/minima found for each data sets are:

Coordinates of the maxima/minima:							
Data		Data1		Data2		Data3	
x	y	x	y	x	y	x	y
0.7455	-13.8121	0.8025	-2.8994	-0.5241	4.4578	-1.6810	3.5870
3.0524	5.3517	1.4547	0.5340	0.7360	-3.0502	-0.2668	5.1271
4.8973	-1.0545	2.1594	-3.6870	2.5848	7.3255	2.2574	1.9686
6.7073	9.6216			4.3714	-5.8644	-1.4038	3.4733
						2.1314	1.9557

Table 1: Coordinates of the maxima/minima for each of the data sets tested

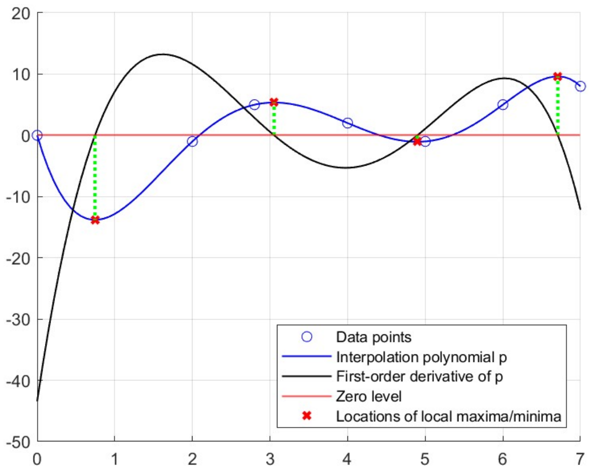


Figure 3: Result plot of the given data set

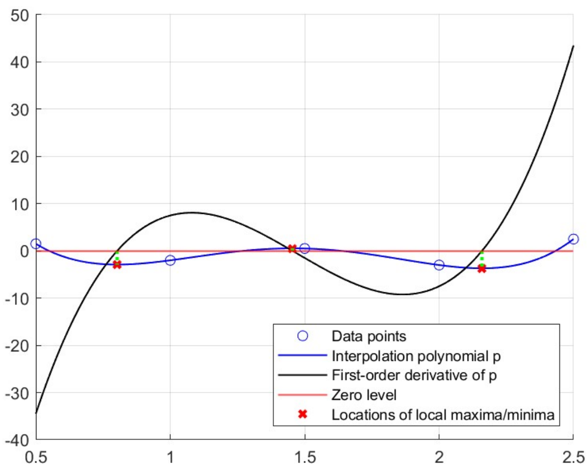


Figure 4: Result plot of data set Data1

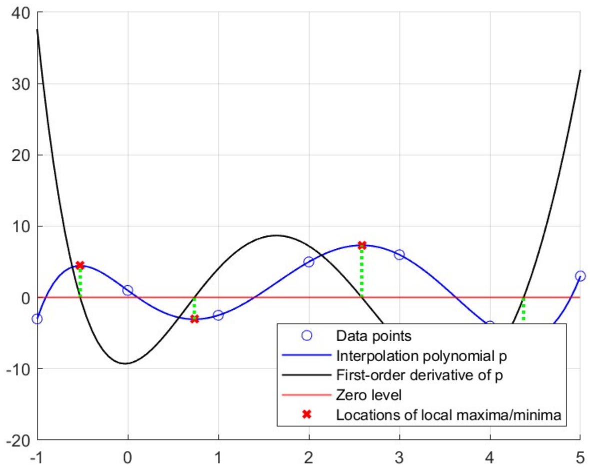


Figure 5: Result plot of data set Data2

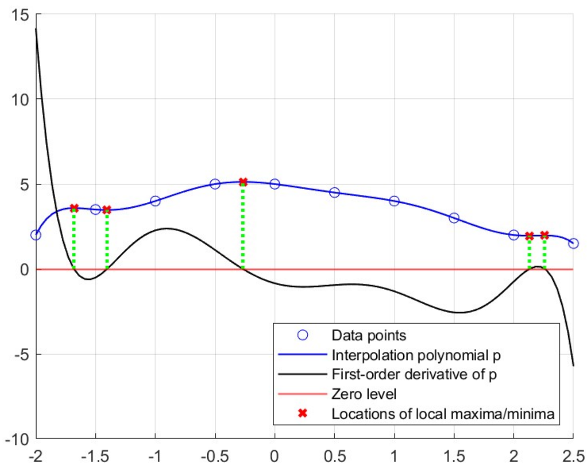


Figure 6: Result plot of data set Data3

Exercise 4

In Matlab, the two ODEs:

$$y'(t) = -y + 3\cos(3t)\exp(-t), \qquad y(0) = 0$$
$$y'(t) = y, \qquad y(0) = 1$$

Were each estimated with both Euler's method and Adams-Bashforth method using different step sizes and compared to the exact solution.

- Euler's method: $y_{n+1} = y_n + h \cdot f(t_n, y_n)$
- Adams-Bashforth method: $y_{n+2} = y_{n+1} + 1.5 \cdot h \cdot f(t_{n+1}, y_{n+1}) - 0.5 \cdot h \cdot f(t_n, y_n)$

Result

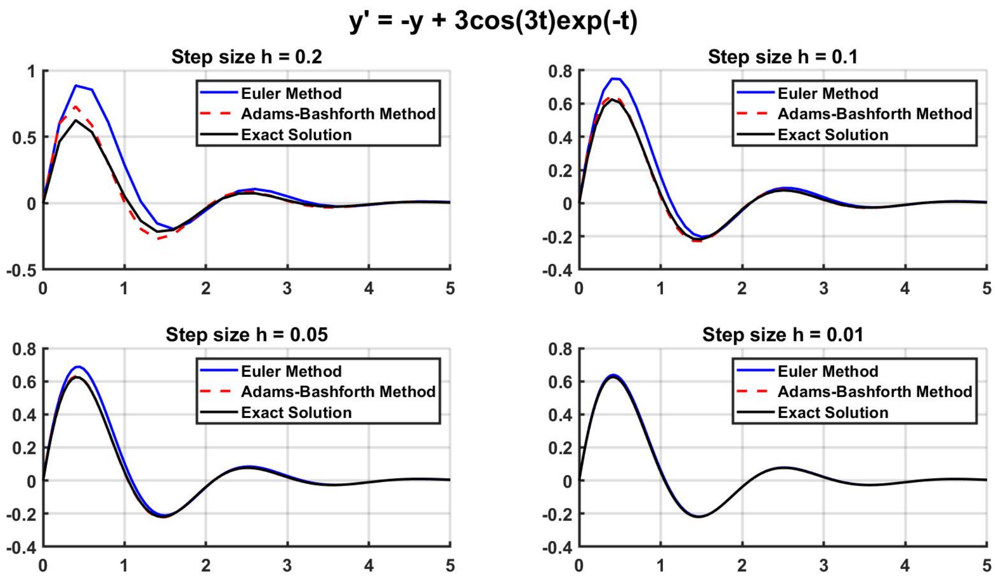


Figure 7: First equation solved with both methods using different step sizes

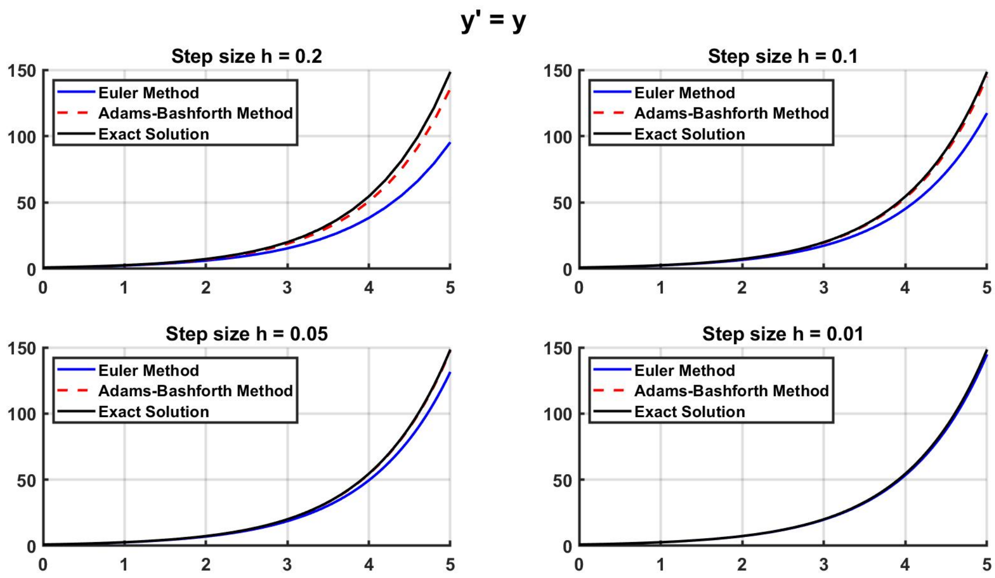


Figure 8: Second equation solved with both methods using different step sizes