DigiPen Institute of Technology, Bilbao

MAT300 Curves & Surfaces
Spring, 2022
Project 2. Deadline: 22/3/2022

1 Main description of the project

The project consists of coding routines to obtain cubic spline curves in 2D and 3D.

Given $n+1$ points (t_0, P_0), (t_1, P_1), $\ldots$, (t_n, P_n) with $t_i < t_{i+1}$, there is a unique cubic spline $p \in P^n_{3,2}[t_0, t_1, \ldots, t_n]$ through the given points satisfying $p''(t_0) = p''(t_n) = 0$.

The idea is to select $n+1$ points in $\mathbb{R}^2$ or $\mathbb{R}^3$. Next select a mesh of nodes over a closed interval $[a, b] \in \mathbb{R}$. Finally, construct and evaluate a cubic spline, which graph will be given as output.

1.1 Input data

The input data will be the following:

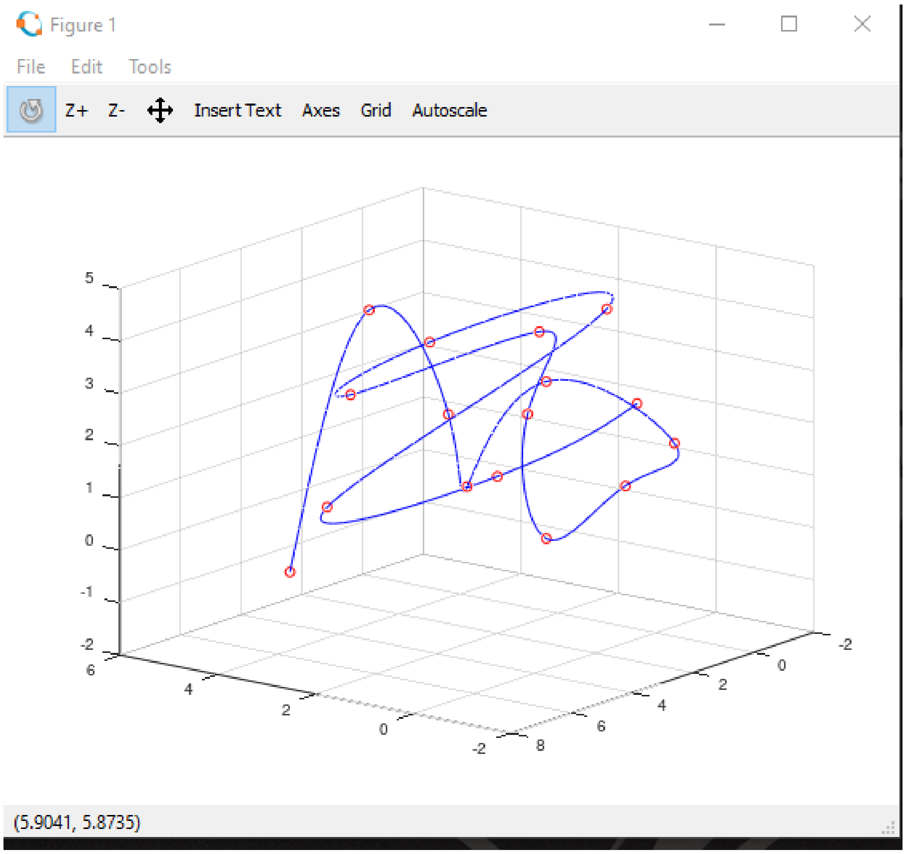
- Interpolation points P_0, P_1, $\ldots$, P_n.
- Dimension (2 or 3).
- Number of nodes for the output mesh.

The data will be introduced in a script file like this:

```
%------------------------------------------------------------
% MAT300 Curves and surfaces
% Julia Sánchez Sanz
% julia.sanchez@digipen.edu
% 15/5/2019
%
% Script input data project 2
%-------------------------------------------------------------

PX=[1 2 3 1 4 5 3 2 1 0 -1 1 2 3 4 5];  % x-coordinate points
PY=[-1 2 6 0 1 2 -1 1 2 1 1 2 3 2 3 4]; % y-coordinate points
PZ=[2 1 1 4 3 2 3 2 0 1 2 3 1 2 4 -1]; % z-coordinate points (un-comment)

Dimension=3; % dimension 2 or 3 (consistent with PZ)

outputnodes=250; % number of nodes for the output mesh
```

1.2 Expected output

When the user runs the program, a figure should appear with the graph of the cubic spline curve and the interpolation points. The plot will be a 2D or 3D plot depending on the input data.



1.3 Main function

The main function of the project will be **cubicspline.m**

The function will construct the mesh for the t-values. Next, it will construct a system of equation and apply the Gauss-Jordan algorithm for obtaining the coefficients of the cubic spline in the standard right-shifted basis. Finally, it will evaluate the spline at the output mesh and plot the result.

1.4 The mesh of nodes

The mesh for this assignment is the regular mesh in $[0, n]$ satisfying $0 = t_0 < t_1 < \ldots < t_n = n$ and same distance among nodes.

1.5 Cubic splines

A cubic spline is a piecewise polynomial $p \in P^n_{3,2[t_0,t_1,\ldots,t_n]}$. Let

$$p(t) = \begin{cases} p_1(t) & t \in [t_0, t_1), \\ p_2(t) & t \in [t_1, t_2), \\ \vdots & \\ p_n(t) & t \in [t_{n-1}, t_n], \end{cases}$$

where $p_i(t) \in \mathbb{P}_3,\ i = 1, \ldots, n$. Moreover we assume that p is continuous with first and second derivatives continuous in $[t_0, t_n]$. A basis for the space $P^n_{3,2[t_0,t_1,\ldots,t_n]}$ is

$$B = \{1, t, t^2, t^3, (t - t_1)^3_+, (t - t_2)^3_+, \ldots, (t - t_{n-1})^3_+\},$$

therefore p can be expressed as a linear combination of elements of B in the following way

$$p(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4(t - t_1)^3_+ + a_5(t - t_2)^3_+ + \ldots + a_{n+2}(t - t_{n-1})^3_+,$$

Given $n+1$ points $(t_0, P_0),\ (t_1, P_1),\ \ldots,\ (t_n, P_n)$ with $t_i < t_{i+1}$, the cubic spline is determined by the solution in $(a_0, a_1, a_2, \ldots, a_{n+2})$ of the system of equations

$$\begin{cases} P_0 & = a_0 + a_1 t_0 + a_2 t_0^2 + a_3 t_0^3 + a_4(t_0 - t_1)^3_+ + a_5(t_0 - t_2)^3_+ + \ldots + a_{n+2}(t_0 - t_{n-1})^3_+, \\ P_1 & = a_0 + a_1 t_1 + a_2 t_1^2 + a_3 t_1^3 + a_4(t_1 - t_1)^3_+ + a_5(t_1 - t_2)^3_+ + \ldots + a_{n+2}(t_1 - t_{n-1})^3_+, \\ P_2 & = a_0 + a_1 t_2 + a_2 t_2^2 + a_3 t_2^3 + a_4(t_2 - t_1)^3_+ + a_5(t_2 - t_2)^3_+ + \ldots + a_{n+2}(t_2 - t_{n-1})^3_+, \\ & \vdots \\ P_n & = a_0 + a_1 t_n + a_2 t_n^2 + a_3 t_n^3 + a_4(t_n - t_1)^3_+ + a_5(t_n - t_2)^3_+ + \ldots + a_{n+2}(t_n - t_{n-1})^3_+. \end{cases}$$

In order to impose a unique solution we need two more constrains, so we will impose the conditions $p''(t_0) = p''(t_n) = 0$ as equations. Notice that we have to solve the above system for each coordinate x, y and z.

1.6 Graphs

In octave 2D plots are computed with the command **plot**. 3D plots are obtained with **plot3**, sometimes they appear planar plots that you have to rotate.

2 Submission instructions

The programming projects will be done in groups of at most three people.

Please, submit all project parts on the Moodle page for MAT300 in a zip. The name of the zip will be

MAT300_ project2_ "surnames of all members of group"

You should include in the zip:

- Octave codes for each part of the project.
- A **readme.text** file specifying which programs the zip contains, in which program is each task of the project, the authors and instructions for running the code.
- A **explanations.pdf** file with the mathematical background. This file will contain at least 3 and at most 5 pages. In this pdf you should present:
  - **the description of the problem:** what is the mathematical problem that the code solves (one sentence or two), what are the inputs and the expected outputs.
  - **the explanation of the numerical methods that you use to tackle the problem:** here you have to present the theory of every method you use, step by step, using the proper terminology, a correct notation and understandable mathematical formulation (you can use latex). The explanations should be carried out for the general case, and not for particular examples. You can use the slides, the books of reference in the syllabus, or internet for information, but do not forget to include the references in the bibliography. If you look at Wikipedia as a source of information, be sure you understand the material, notation and formulation before doing copy-paste, and be sure that material is consistent with the contents of the course. The notation has to be consistent along the whole document, and if possible the same as the used in class. Please, use a common form along the document (same font, same size and so on).
  - **the relation between the above methods and your code:** you should link the above descriptions to the files and lines of code (what the code does and in which order, what is done-where is done).
  - **examples showing that your code indeed works:** you should compare your obtained results with analytical solutions. It worth to compute for a short number of nodes cubic splines by hand and evaluate them in some points and compare with the computations.
  - **observations:** behavior of curves, problems of computations if any, time of computation, comparison among cubic spline and interpolation, and other additional information that you may consider relevant.
  - **bibliography:** include the references that you used as literature.

3 Grading policy

The total grade of this programming assignment will be computed as follows:

- Codes 60%:

  - Cubic spline algorithm works 35%
  - Meshes are correct 5%
  - Outputs in 2D and 3D are correct 5%
  - Codes are structured and clean. They implement Octave matrix and vector algebra 5%
  - Codes are well commented with headers and mathematical explanations with correct terminology in each line 10%
- Readme 4%
- Document 36%:
  - Description of the problem 3%
  - Mathematical explanation of the numerical methods that you use to tackle the problem: 20%
  - The relation between the above methods and your code: 5%
  - Examples: 5%
  - Observations: 3%

Concerning late submissions I will restrict to the late policy included in the syllabus.