Coursework 2: MATH 3018/6141 - Numerical methods

Due: 7 January 2021

In this coursework you will solve a problem using numerical algorithms. The assessment will be based on

- sensible choice and explanation of the algorithms – **5 marks**
- correct implementation of chosen algorithms – **6 marks**
- construction and explanation of appropriate tests – **3 marks**
- documentation and error checking of code – **3 marks**
- appropriate convergence tests – **3 marks**.

The deadline is noon, Thursday 7 January 2021 (week 11). For late submissions there is a penalty of 10% of the total marks for the assignment per day after the assignment is due, for up to 5 days. No marks will be obtained for submissions that are later than 5 days.

Your work must be submitted electronically via Blackboard. Only the Python files needed to produce the output specified in the tasks below is required.

1 Political pricing

With a massive political upheaval in the next year, a government has told a monopoly company that it must reduce prices by 10% over the next year or receive crippling fines. The company wants to do this as slowly as possible to maximize profits, but realizes that its customers will put off buying its goods, despite the monopoly, if they think they can get a better deal by waiting. The company assumes that it therefore wants to set its prices (in dimensionless units) so that $y(0) = 1$, $y(1) = 0.9$ (a 10% decrease over one year), and so that their profit (which is modelled as the charge y minus a penalty due to customers waiting for the future) is as large as possible.

Assuming the penalty is

$$P(t; \dot{y}) = \alpha \left(\dot{y}\right)^2 + \beta(t^2 - 1) \left(\dot{y}\right)^3,$$

your task is to find the solution $y(t)$ that maximizes the profit by minimizing

$$S = \int_0^1 (P - y) \, \mathrm{d}t \tag{1}$$

subject to the boundary conditions $y(0) = 1, y(1) = 0.9$.

This is in the form of a variational problem where the task in to minimize the action

$$S = \int_a^b L \left(t, y(t), \dot{y}(t), \ddot{y}(t), \dots\right) \, \mathrm{d}t, \tag{2}$$

where L is some function of t and y and its derivatives, subject to boundary conditions on $y(t)$. This can be done using the *Euler-Lagrange equations*,

$$\frac{\partial L}{\partial y} - \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{y}}\right) = 0, \qquad y(a) = A, \quad y(b) = B. \tag{3}$$

By using the chain rule we can expand this to give

$$\frac{\partial L}{\partial y} - \left(\frac{\partial^2}{\partial t \partial \dot{y}} L + \dot{y} \frac{\partial^2}{\partial y \partial \dot{y}} L + \ddot{y} \frac{\partial^2}{\partial \dot{y} \partial \dot{y}} L\right) = 0, \qquad y(a) = A, \quad y(b) = B. \tag{4}$$

As L is a *known function* of y and its derivatives, this is an *ODE boundary value problem* for $y(t)$. Given values for $y, \dot{y}$, the derivative of L can be computed (e.g., using finite differencing).

2 Task

1. Write an algorithm to solve the boundary value problem given by the Euler-Lagrange equation (4). If your algorithm is not completely general you should state the restrictions required in order for the algorithm to work. Your algorithm should take as input the function L and the boundary values $y(a), y(b)$ but must compute all derivatives in equation (4) numerically. Justify your choice of algorithm.

2. Construct tests where you know the exact solution and show that your algorithm performs as you expect on these tests. Explain what aspects of your algorithm are checked by these tests.

3. Maximize the profit for the company for the two models

   (a) $\alpha = 5 = \beta$;

   (b) $\alpha = \frac{7}{4}, \beta = 5$.

   For the second case, give evidence that your solution is converging in an appropriate sense. Also state if the mathematical problem has been posed correctly to solve the original problem.

2.1 Summary and assessment criteria

You should submit the code electronically as noted above.

2.1.1 Python

As Python allows for many function definitions in one file, you may submit as many Python files as necessary to solve the problem. However, one file should clearly be the master script that when run (eg from the console inside spyder), produces all the plots required.

2.1.2 Plots

The code you submit will, if it answers all the tasks, produce at least 3 (three) figures:

1. For the first ($\alpha = \beta$) case, the price curve $y(t)$ as a function of t;

2. For the second ($\alpha \neq \beta$) case, the price curve $y(t)$ as a function of t;

3. For the second case ($\alpha \neq \beta$), at least one figure showing evidence that your algorithm is converging to some solution as you vary its accuracy.

Plots should be displayed to the screen.

2.1.3 Assessment criteria

The primary assessment criteria will be the

- choice and justification of the algorithm: this must be explained and documented in comments within the code, preferrably at the start of the main file;
- correct implementation and use of your chosen algorithm: this will be shown through the tests above and the figures produced.

The secondary assessment criteria will be the clarity and robustness of your code. Your functions and scripts must be well documented with appropriate internal comments describing inputs, outputs, what the function does and how it does it. The code should also be well structured to make it easy to follow. Input must be checked and sensible error messages given when problems are encountered. Unit tests of individual functions may also be used to show the correctness of the code. The clarity of the output (such as plots or results printed to the command window) is also important.

Code efficiency is not important unless the algorithm takes an exceptional amount of time to run. As an indication, the model implementation runs all tasks in a maximum of 2 minutes.