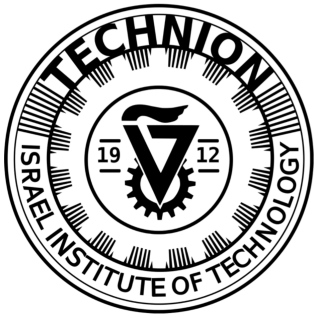
Technion - Israel Institute of Technology

Numerical Methods in Aeronautical Engineering (086172)

GRADE	OUT OF	CHAPTER
	2	ABSTRACT
	2	CONTENTS, STYLE &C.
	4	PHYSICAL PROBLEM
	4	MATHEMATICAL MODEL
	26	NUMERICAL METHODS
	20	INFLUENCE OF NUMERICAL METHODS
	20	RESULTS
	2	SUMMARY & CONCLUSIONS
	20	COMPUTER PROGRAM
	100	**TOTAL**



Daniel Engelsman, ID 300546173 @ March 23, 2019

= Homework Assignment 1 =

Contents

List of Figures

¡ – Consider attaching list of figures

1 Abstract

In this assignment we were asked to solve a diffusion and a chemical reaction which are described by a coupled of non-linear equations. To that end we'll use the studied tools at class and compare their performances under different parameters. The main **computational** goal is to portray the temperature and the concentration distribution along the x domain [0, 1], using the Shooting method vs. finite differences, which will be further elaborated in the coming chapters.

2 The physical problem

A diffusion and a chemical reaction are given as a coupled 2nd order ODE problem. Both are defined as functions of *Concentration* (C) and *Temperature* (T), alongside constants : *Activation energy* (β), *Thermicity* (γ) and *Thiele modulus* (ϕ).

3 The mathematical model

We'll denote $d/dx = (\quad)'$ such that :

$$T'' + a(x_i)T' + b(x_i)T = -\phi^2 \beta C e^{\gamma(1-\frac{1}{T})} \tag{3.1}$$

$$C'' + a(x_i)C' + b(x_i)C = \phi^2 C e^{\gamma(1-\frac{1}{T})} \tag{3.2}$$

Boundary and Initial conditions of the problem :

$$T(x) \quad : \quad T'(0) = 0, \quad T(1) = 1 \tag{3.3}$$

$$C(x) \quad : \quad C'(0) = 0, \quad C(1) = 1 \tag{3.4}$$

Using central differences, the derivative's initial condition requires the following treatment :

$$y'(0) = \frac{y_1 - y_{-1}}{2h} = \alpha \quad \rightarrow \quad y_{-1} = y_1 - 2\alpha h, \qquad i \in [0,\ N] \tag{3.5}$$

Such that the x domain is initialized as : $[x_{n-1}^{(i=0)} = x_{-1}, \quad x_{n+1}^{(i=N)} = x_{N+1}]$.

Using the course's model equation :

$$y'' + a(x_i, y_i)y' + b(x)y^2 = q(x) \qquad 0 \leq x \leq 1 \tag{3.6}$$

In case of L'hopital theorem, put the relevant initialization :

$$y'' + a(x_i, y_i)y' + b(x)y^2 = q(x) \qquad \rightarrow L'hop... \tag{3.7}$$

4 The numerical method

(A) Shooting method

An iterative approach where we substitute $y' = u$ and plug an initial guess $y'(0)$ or $u(0)$ into a *Runge-Kutta* algorithm so we can solve the ODE system. Every iteration we check whether our solution gets closer to the given boundary value β :

$$f(u(0)) \equiv y(1, u^n(0)) \simeq \beta \tag{4.1}$$

The "optimization process" is followed step by step by using the *Secant* method, which is suitable for the non-linear case, and takes only very few iterations :

$$u^{n+1}(0) = u^n(0) - \frac{f(u^n(0))(u^n(0) - u^{n-1}(0))}{f(u^n(0) - f(u^{n-1}(0))}$$

(B) Finite differences

Another iterative approach where the following ODE ($i \in [1, N]$) :

$$y_{i-1}\Big[1 - \frac{a(x_i, y_i)h}{2}\Big] + y_i\Big[-2 + h^2 b(x_i, y_i)y_i\Big] + y_{i+1}\Big[1 + \frac{a(x_i, y_i)h}{2}\Big] = h^2 q(x_i, y_i) \tag{4.2}$$

(O) Auxiliary methods

∘ Euler method - (?? necessary ??) a numerical method to solve 1st order ODE with a given initial value, with a local error proportional to a square step size $\mathcal{O}(h^2)$:

$$y_{n+1} = y_n + hf(t, y(t)) = y_n + hy'_n \tag{4.3}$$

∘ Runge-Kutta ($RK4$) - a numerical method to approximate a solution of an ODE with a local accumulated error of $\mathcal{O}(h^5)$. It is used as the integrator of the initial value problem posed in the shooting method every iteration.

* Check whether a higher order approximation is also acceptable when using Runge-Kutta $\mathcal{O}(h^3)$.

Where the x domain is: $[x_{n-1}^{(i=1)} = x_0, \quad x_{n+1}^{(i=N)} = x_f]$, and is calculated iteratively until reaching convergence criterion. However, there are several methods we could use to implement it, exhibiting different level of efficient performance. Ideally, the easiest way to sort an ODE would be using *Jacobi* method, such that all entries occupy a tri-diagonal from in $A(\underline{y})_i^{n+1} = \underline{b}$ form $(... \equiv 0)$:

$$\begin{bmatrix} C_2 & C_3 & 0 & ... & ... & 0 \\ C_1 & C_2 & C_3 & ... & ... & ... \\ ... & C_1 & C_2 & C_3 & ... & ... \\ ... & ... & ... & ... & ... & ... \\ ... & ... & ... & ... & ... & C_3 \\ 0 & ... & ... & ... & C_1 & C_2 \end{bmatrix} \begin{bmatrix} y_1^{n+1} \\ y_2^{n+1} \\ ... \\ ... \\ ... \\ y_N^{n+1} \end{bmatrix} = \begin{bmatrix} h^2 q_1 - y_0^n \cdot C_1 \\ h^2 q_2 \\ ... \\ ... \\ h^2 q_{N-1} \\ h^2 q_N - y_f^n \cdot C_3 \end{bmatrix} \tag{4.4}$$

However, I chose to use *SUR*, which is based on *Gauss-Seidel* method, but performs better due to the non-linearity nature of the problem :

$$y_i^{n+1} = \frac{q(x_i, y_i) - y_{i-1}^{n+1}\left[1 - \frac{a(x_i,y_i)h}{2}\right] - y_{i+1}^{n}\left[1 + \frac{a(x_i,y_i)h}{2}\right]}{\left[-2 + h^2 b(x_i, y_i) \cdot y_i^n\right]} \tag{4.5}$$

For convenience, we'll substitute the brackets with the following coefficients :

$$y_{i-1}^{n+1} \cdot C_1 + y_i^{n+1} \cdot C_2(y_i) + y_{i+1}^{n} \cdot C_3 = h^2 q_i \quad \rightarrow \quad y_i^{n+1} = \frac{h^2 q_i - y_{i-1}^{n+1} \cdot C_1 - y_{i+1}^{n} \cdot C_3}{y_i^n \cdot C_2} \tag{4.6}$$

Due to its implicit nature, we'll separate y_i^{n+1}, y_i^n such that y_i^{n+1} can be extracted :

$$\begin{bmatrix} C_2 & 0 & 0 & ... & ... & 0 \\ C_1 & C_2 & 0 & ... & ... & ... \\ ... & C_1 & C_2 & 0 & ... & ... \\ ... & ... & ... & ... & ... & ... \\ ... & ... & ... & ... & ... & 0 \\ 0 & ... & ... & ... & C_1 & C_2 \end{bmatrix} \begin{bmatrix} y_1^{n+1} \\ y_2^{n+1} \\ ... \\ ... \\ ... \\ y_N^{n+1} \end{bmatrix} + \begin{bmatrix} 0 & C_3 & 0 & ... & ... & 0 \\ 0 & 0 & C_3 & ... & ... & ... \\ ... & 0 & 0 & C_3 & ... & ... \\ ... & ... & ... & ... & ... & ... \\ ... & ... & ... & ... & ... & C_3 \\ 0 & ... & ... & ... & 0 & 0 \end{bmatrix} \begin{bmatrix} y_1^n \\ y_2^n \\ ... \\ ... \\ ... \\ y_N^n \end{bmatrix} = \begin{bmatrix} h^2 q_1 - y_0^n \cdot C_1 \\ h^2 q_2 \\ ... \\ ... \\ h^2 q_{N-1} \\ h^2 q_N - y_f^n \cdot C_3 \end{bmatrix}$$

$\circ$ $L^* \equiv$ Lower triangular (dominant) $\quad \circ$ $U \equiv$ Upper triangular

Therefore, we get :

$$L^*(\underline{y})_i^{n+1} + U(\underline{y})_i^n = \underline{b}_i \quad \rightarrow \quad (\underline{y})_i^{n+1} = (L^*)^{-1}(\underline{b}_i - U(\underline{y})_i^n) \tag{4.7}$$

Trial and Error

At both methods few parameters remained unknown such that an empiric method should have been used. Finding compatible step size, range for initial guesses in shooting method ... Include graph

5 Influence of the numerical methods

The graphs themselves of both methods, better off 3D : Convergence vs. h vs. convergence criterion.

6 Results

You should give details of all the results computed and try to give a physical description of the behavior of what you found. **Number graphs correctly and refer them later**.

7 Summary and conclusion

A brief summary of the main results, numerical and qualitative, should be provided, as well as conclusions about your choice of numerical methods, comments about the work, suggestions for future work.

8 Computer program

The program should be written modularly. Each section of the program should contain a comment about what that sections function is. The more comments you include the easier it is for the reader to follow the program. You can write the program in whatever computing language you want.