UNIVERSITY OF BOLOGNA
SCUOLA di Architettura ed Ingegneria

Two year master in

Biomedical Engineering

SALVATORE PINELLI
983685

Project for Numerical Analysis PART B
A.A 2020/21

Project ODE N. 2

PHARMACOKINETIC MODELING OF HEPARINE

1. PROJECT INTRODUCTION

Heparin was discovered in 1916 by J. McLean. In 1939, K. Brinkhous demonstrated that the anticoagulant activity of heparin requires a plasma cofactor. In 1968, U. Abildgaard renamed this cofactor as antithrombin III, and two years after the interaction between this two substances was explained. [1]
Heparin is a mucopolysaccharide extracted from the intestinal mucosa of swine or from the bovine lungs, which has a molecular weight varying between 6000 and 2000.

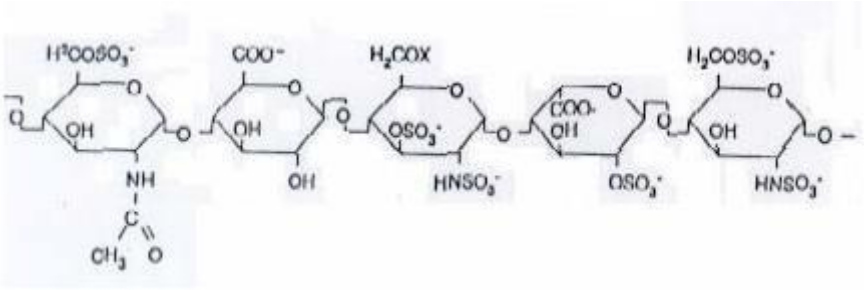


Figure 1: structure of pentasaccharide sequence

The site on the heparin molecule is a pentasaccharide sequence. In the absence of heparin, in contact with foreign surfaces, the blood rapidly coagulates. For this reason, it is used during extracorporeal circulation (dialysis, plasmapheresis, etc.) but it is also administered to prevent thrombosis in patients at risk.
The concentration of heparin is evaluated by measuring its antithrombotic effect, i.e. its ability to prolong the clotting time (Activated Partial Thromboblastin Time, APTT).It is measured in International Units (IU): 1 mg corresponds to about 100-110 IU .
The dose of heparin to be administered during the session must be chosen carefully.
For this purpose it is necessary to know the kinetics of the body's elimination of heparin.
Heparin is poorly absorbed from the gastrointestinal tract and is usually administred by intravenous or subcutaneous injection. After injection, heparin circulates bound to many plasma proteins.[2]
It is not clear what chemical characteristics of heparin molecules are responsible for its anticoagulant effect. The pharmacokinetics of heparin are complicated and, in general, previous studies have revealed that heparin kinetics are nonlinear, characterized by a dose-dependent increase in the apparent biological half-life in plasma.[3]

2. DESCRIPTION OF THE MATHEMATICAL MODEL

Therefore it is important to characterize the elimination kinetics in order to define the mathematical model. For the examined system there are no incoming flows, nor terms of surface accumulation, but only terms of elimination.

The temporal trend of the heparin concentration in blood decays as a negative exponential , and is due to two main sources: renal and metabolic elimination.
Renal elimination is predominant for high concentrations of heparin, expecially in the first hours following the administration of the bolus. A strongly non-linear trend can be seen for these concentrations.
Instead, metabolic degradation begins to predominate over renal elimination at low concentrations. This threshold value is reached with good approximation for concentrations which are lower than 1 IU / ml of blood.
For the first section (high heparin concentrations) the curve is described by the following Non-Linear ODE:

$$\frac{dC}{dt} = -LC - \frac{v_{Max} C}{K_M + C}$$

Eliminaz.renale *Eliminaz. metabolica*

Figure 2. High heparin concentration model

For low heparin concentrations it is C << Km. Therefore, it is not possible to consider the concentration in the denominator of the generative term linked to metabolic degradation. The kinetics is described through the following Linear ODE:

$$\frac{dC}{dt} = -LC - \frac{v_{Max}}{K_M} C$$

Figure 3. Simplified model for low heparin concentration

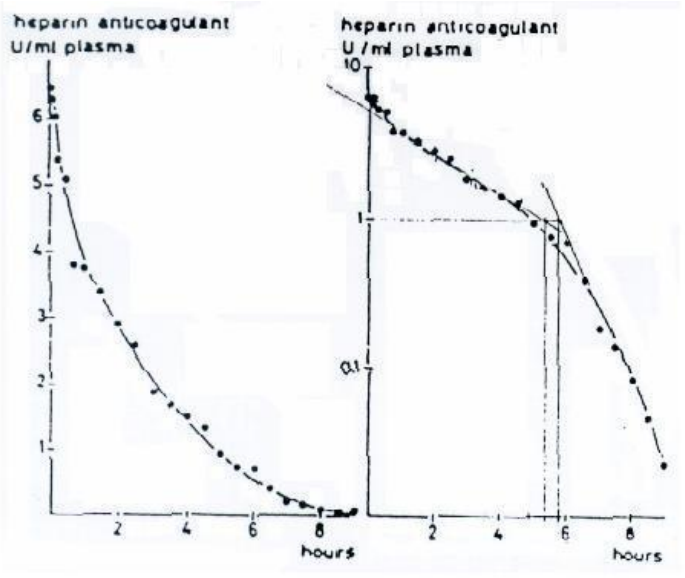


Figure 4. Kinetics of the heparine concentration in blood in linear scale (left) and in semi-log scale (right)

3. REALIZATION OF THE PROJECT IN MATLAB

The mathematical formulation described above constitutes a piecewise first-order ODE. Since it depends on the dose of the bolus injected, this problem can be defined as an Initial Value Problem (IVP).

3.1 Choice of parameters.

As regards the parameters of the model, the following values can be found in the literature:

- the renal kinetic constant (L), from the slope of the initial part of the semilogarithmic diagram, is L = 0.35 h^{-1} (approximately).
- starting from appropriate experiments on subjects affected by renal insufficiency and on healthy subjects, the value of K_m varies on the the range of 0.4-1.6 h^{-1}. [4]
- as regards the V_{max}, its value can be defined by considering the simplified version of the model: $(L+V_{max}/K_m)$= [1- 1.2] h^{-1}. Starting from this relation the values of V_{MAX} can be defined once the values of L and K_m have been fixed.

$$K_m \in [0.4 - 1.6]\,[h^{-1}] \;;\; L = 0.35\,[h^{-1}] \;;\; (L + \frac{V_{max}}{K_m}) \in [1 - 1.2]\,[h^{-1}]$$

Figure.5 Model parameters

3.2 Evaluation of ODEs. (well posed and conditioned problem)

The function that defines our problem is differentiable for any value of t and C. Consequently, it represents a Lipschitz function.

$$f_c = -L - \frac{(K_m * V_{max})}{(K_m + C)^2} \qquad f \in \mathbb{R} \;\; \forall C, t$$

$$f_c = -L - \frac{V_{max}}{K_m} \qquad f \in \mathbb{R} \;\; \forall C, t$$

Figure.6 Jacobian function of piecewise ODE

For this reason, the solution has existence uniqueness and has a continuous dependence on the initial data.
Since the function verifies the Lipschitz condition, IVP is a well posed problem.
From the study of the Jacobian we note how:

- The first part of the function is always negative for each value of the variables and parameters. (As shown in the next figure, via the custom script stability.m)

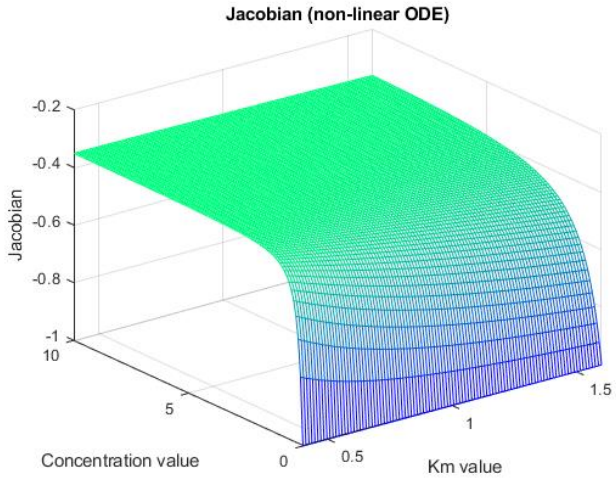


Figure.7 Jacobian function of non-linear part with all value of Km

- The second part of the function is constant, with a value of -1.

The necessary condition for stability is met in both parts of our function. For this reason, it has been demonstrated the well condition of the problem.
The methods used to solve the ODE are:
Euler Explicit (forward), Euler Implicit (backward), Heun and ode45 (considered as the exact solution).
In the stability.m script we are going to evaluate the maximum value of the Jacobian (λ) to identify the stability region of the methods used. Since λ=-1 the choice of the discretization step (h) falls on positive values and less than 2, in order to make the methods of Heun and Explicit Euler convergent (Absolute Stability).
Since we have chosen only real values of h, the stability regions of Heun and EE coincide. The other two methods, being unconditionally stable, do not require major restrictions on the value of h.
As shown in the next figure, choosing values of h> 2, the solutions reported by the explicit methods diverge.

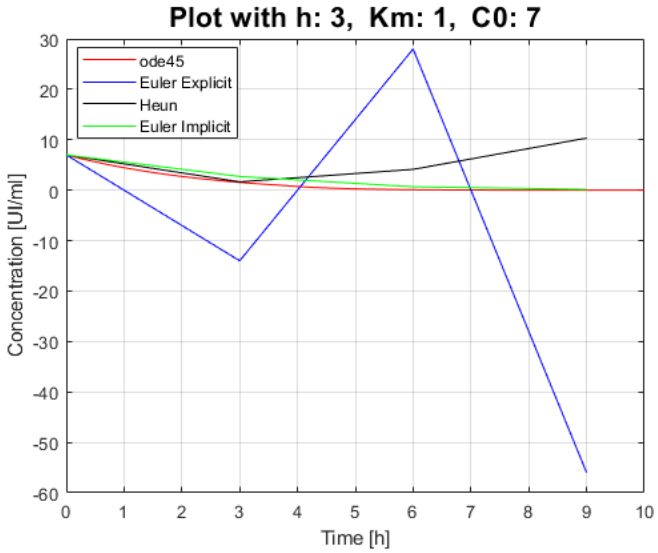


Figure.8 Divergence of explicit method (h>2)

3.3 Explanation of the Project.m script.

First, discretization steps and constants (K_m and C(0)) are requested from the user.
The costants L and K_m are global, so that the functions that describe the ODE and its Jacobian (heparin.m, heparin2.m, jac_hep.m and jac_hep2.m) can also be parameterized. The time interval for the calculation is defined by tspan.
At this point, after the constants L and K_m have been defined, the constant V_{max} is calculated considering the following equation (using the average value of the interval [1-1.2]):

$$(L + \frac{V_{max}}{K_m}) = 1.1\ [h^{-1}]$$

Figure.9 Definition of Vmax.

```
      %variable declaration
global Km L;
disp('In order to reach the stability of explicit method, choose h<2')
disp('                  ------------------------                    ');
h=input('Choose discretization step size (h): ');
Km=input('Choose the value of Km included in [0.4-1.6]: ');
C0=input('Choose volume of heparin (C0): ');
tspan = [0 8];
L=0.35;
Vmax=(1.1-L)*Km;
```

Figure.10 Declaration variable

First, I solved the ODE via the ode45 in order to get the exact solution.
I set the relative error of 10 ^ (-10) and, when the concentration has reached the threshold of 1, I stopped the calculation via the myEvent.m function. In this way I could calculate the function for the nonlinear section defined by the heparin.m function. Then I calculated the solution of the linear ODE starting from t1(end) with initial condition C1(end).
At this point I combined the two solutions by eliminating the repeating component.

```
        %solve with ode45
Opt=odeset('RelTol',1e-10,'events', @myEvent);
[t1,C1] = ode45(@ heparin, tspan, C0,Opt);
[t2,C2] = ode45(@ heparin2, [t1(end) 8], C1(end));
t=[t1(1:end-1);t2];
C=[C1(1:end-1);C2];
```

Figure.11 Code solve with ode 45.

The second step is to calculate the solution using the functions eul_exp.m, eul_imp.m, heun.m.
Since it is not possible to stop integration for C = 1 (as with the ode45), I wrote the following code to achieve the same result.

```
20 -    [teel,Ceel]=eul_exp(@ heparin,tspan,C0,h);
21 -    Cee=zeros(length(Ceel),1);
22 -    for i=1:length(Ceel)
23 -        if Ceel(i)>=1
24 -            Cee(i)=Ceel(i);
25 -        else
26 -            Cee(i)=0;
27 -        end
28 -    end
29 -    Cee=Cee(1:nnz(Cee),1);
30 -    teel=teel(1,1:nnz(Cee));
31 -    [tee2,Cee2]=eul_exp(@ heparin2,[teel(end) 8],Cee(end),h);
32 -    tee=[teel(1:end-1),tee2];
33 -    Cee=[Cee(1:end-1);Cee2];
```

1 2 3

Figure.12 Code solve with Euler Explicit

The previous figure describes the process of eul_exp, but this happens in the same way for the other two methods.
First of all I calculated the solution on the whole tspan, then I created an empty vector of the same length as the solution obtained.

1. With the for loop, I copied all the concentration values higher than 1 into the new vector, while all the others remained null.
2. I removed all null elements from the vector and reduced the size of the time vector (tee). By doing so, I obtained the solution for concentrations higher than 1.
3. Finally, I computed the solution of the linear ODE starting from tee1 (end) with initial condition Cee (end). At this point, I joined the two solutions by eliminating the repeating component (as I did previously with ode 45).

After obtaining all the solutions of the methods, I evaluated the relative error of the methods used (EE, EI, Heun) with respect to the exact solution (ode45), using the error_evaluation.m function.
Since the solutions have different dimensions, I used the spline function to evaluate the solution of the ode45 (not equidistant) in the time values obtained by the methods we implemented (equidistant with step h).
After that, I calculated the relative error for all three methods.

Project.m | error_evaluation.m

```
function [err_ee,err_ei,err_h]=error_evaluation(C,t,Cee,tee,Ch,Cei)
C_45=spline(t,C,tee);
err_ee=abs((Cee'-C_45))./abs(C_45);
err_ei=abs((Cei'-C_45))./abs(C_45);
err_h=abs((Ch'-C_45))./abs(C_45);
```

Figure.13 Code error_evaluation.m

At the end, two figures are shown on the screen: the first contains the linear and semilogarithmic plots of the solutions,while the second contains the relative errors plots.

4. NUMERICAL EXPERIMENTATION and CONCLUSIVE CONSIDERATIONS

4.1 Exact solution with different Km values.

First, I evaluated how the Km parameter affected the curve.
Using the script Var_Km.m I plotted on the same graph the solutions for the two extreme values and for the average value.
It can be seen that for Km max the half-life is shorter.
The greatest difference is noted by evaluating the change point of the solution, i.e. when the concentration reaches the threshold of 1. For Km = 1.6, the threshold is reached after about 3 hours, while for Km = 0.4 this threshold is reached about an hour later. This is also confirmed by the articles found in the literature. [4]

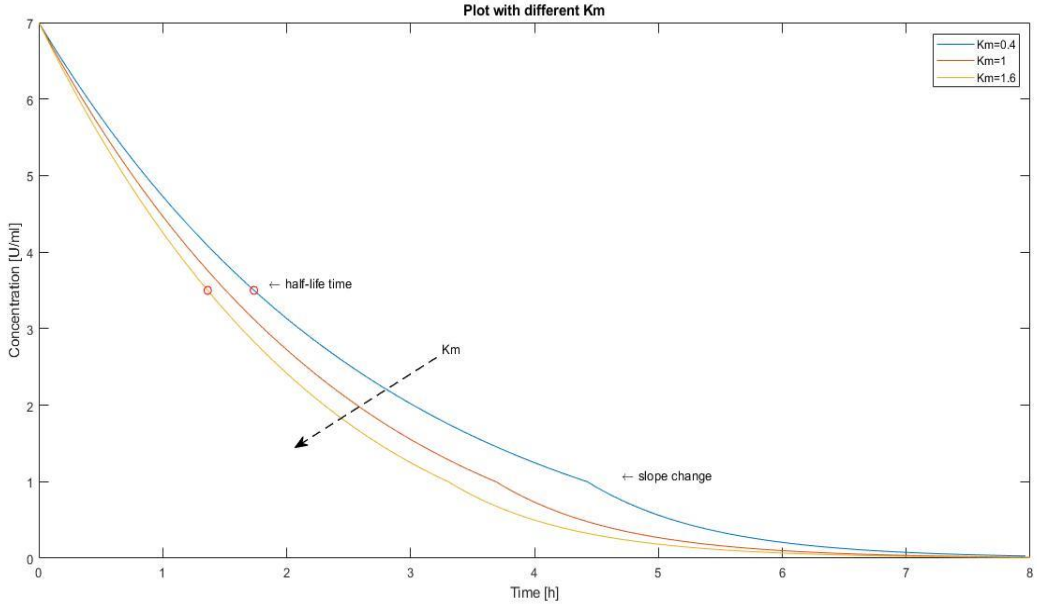


Figure.14 Relationship between concentration and Km value

4.2 Results with different method

For my simulations, I considered a standard male (70 kg) non-smoker (Km = 0.77) [4], who is given a 7 ml intravenous dose of clexane (standard dosage proposed in the Summary of Product Characteristics of the AIFA (Italian Medicines Agency)). [5]
The graphs for different discretization step values (0.5 / 0.1 / 0.01) are shown below.

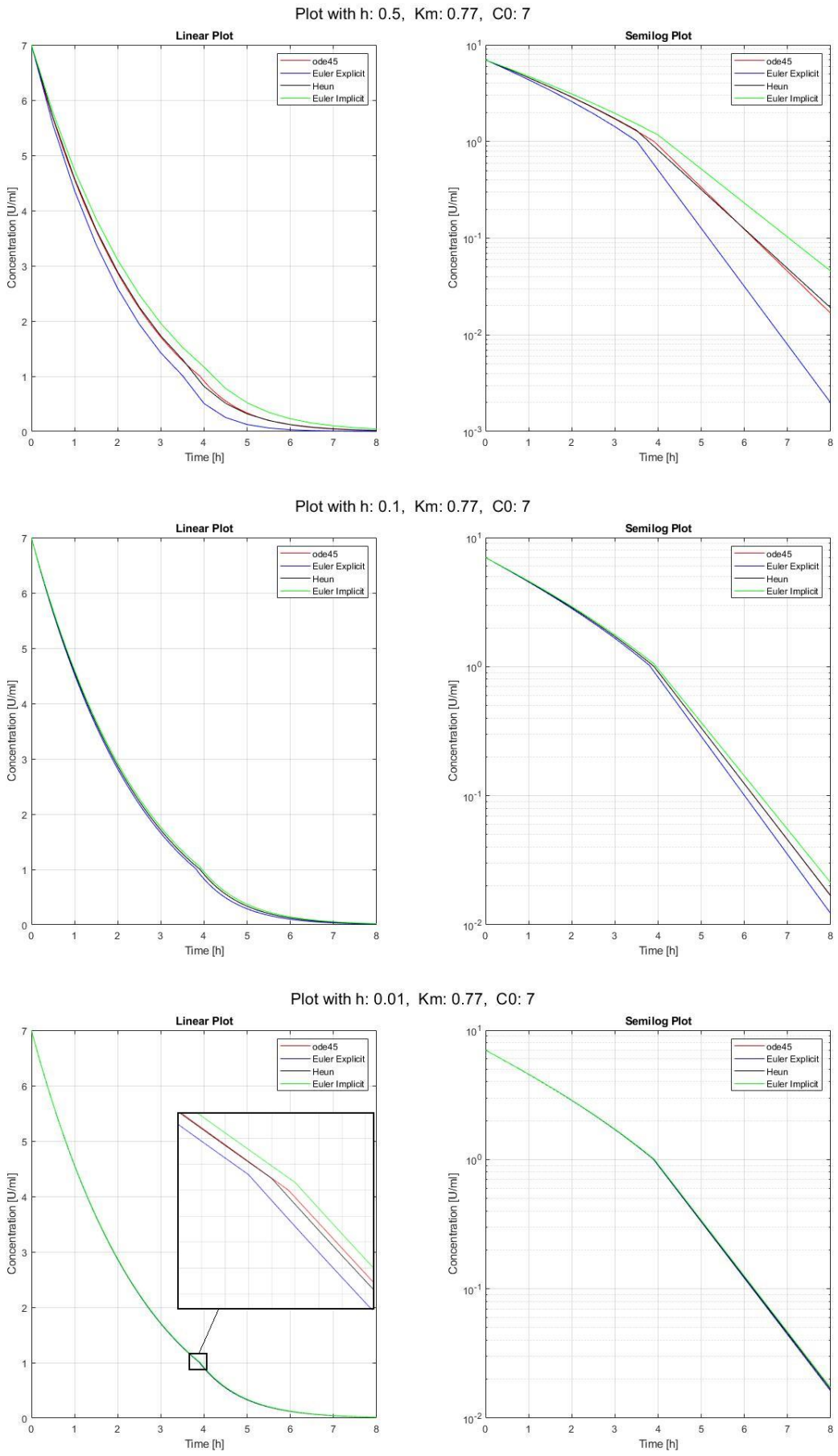


Figure.15 Different solution with different discretization step

As expected, for very low discretization steps, the solutions obtained are very similar to the exact solution. It is clear from the relative error values (following figure) that the methods reach the solution with an excellent approximation. For these values of h it can be noticed how Heun is a second order method. For this reason, its relative error is one order lower than the values of the two Euler methods.

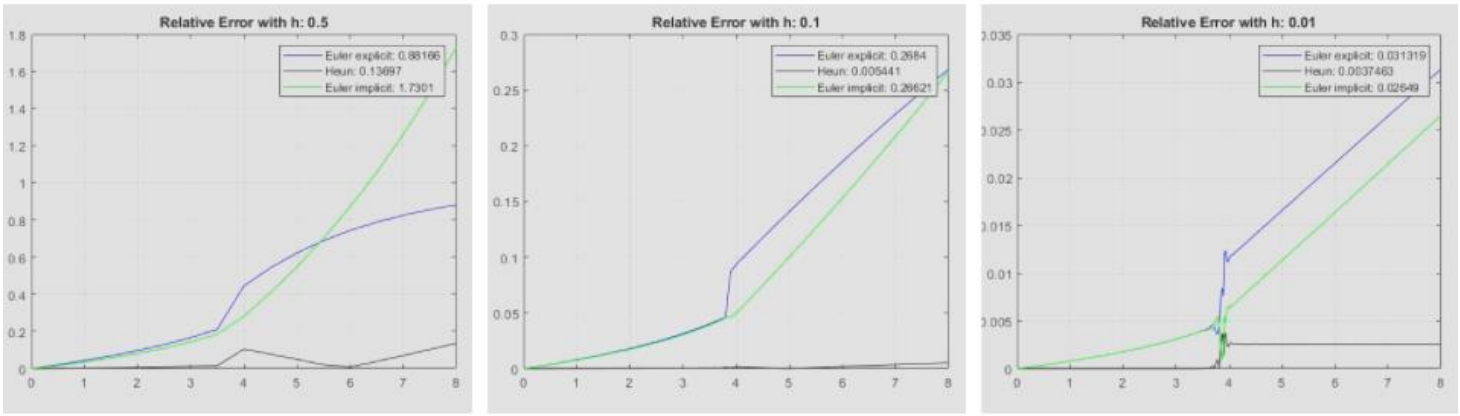


Figure.16 Different Relative Error

4.3 Max relative errore evaluation

Subsequently, I evaluated the maximum relative error, as h. For discretization step values less than 1, explicit methods have a lower maximum relative error than Euler Implicit. (Next fig.)

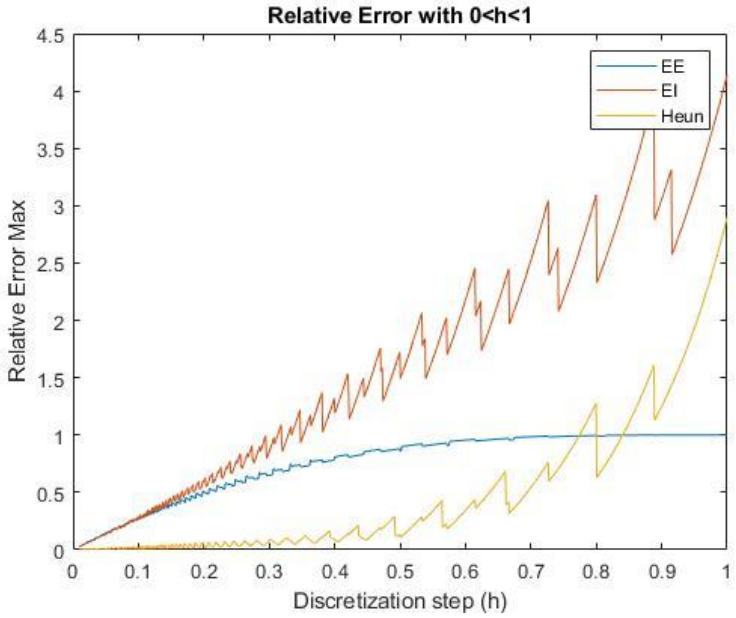
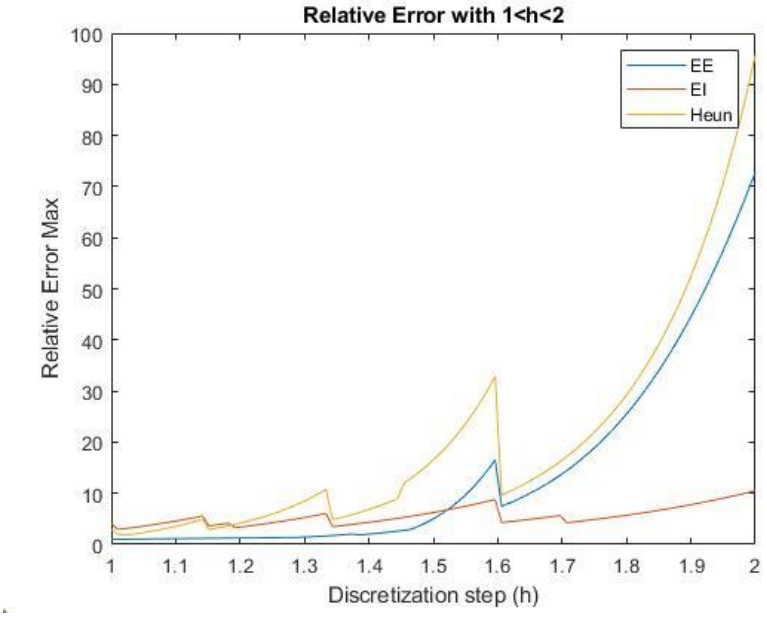


Figure.17 Relative error with 0<h<1

As the value of h increases, the trend changes. In fact, the Explicit methods converge (h <2) but the solution obtained significantly differs from the exact solution. The following graph shows that the relative error of the two methods increases exponentially as the value of h increases.



Step (h)	Eul Exp	Eul Imp	Heun
1,00	1,00	4,14	2,90
1,05	1,07	3,63	2,37
1,11	1,14	4,69	3,68
1,16	1,20	3,75	3,07
1,21	1,27	3,56	4,49
1,26	1,33	4,49	6,53
1,32	1,54	5,63	9,50
1,37	1,97	3,84	5,66
1,42	2,27	4,68	7,77
1,47	3,37	5,68	13,71
1,53	7,18	6,87	19,93
1,58	13,66	8,28	29,06
1,63	8,84	4,64	11,18
1,68	12,47	5,44	15,08
1,74	17,41	4,68	20,35
1,79	23,97	5,53	27,53
1,84	32,51	6,51	37,34
1,89	43,35	7,65	50,85
1,95	56,75	8,98	69,57
2,00	72,78	10,51	95,75

Figure.18 Relative error with 1<h<2 (Var_h.m script)

4.4 Elapsed time

Through the Matlab tic toc function, I measured the execution time of the different codes. As expected, the implicit method, which requires solving a linear system, takes more time than the explicit methods. The difference with the default matlab method is also clear: it has a slower execution time since it chooses non-equidistant points for the resolution of the ODE.
As for the size of the solution, the ode45 gives us a vector with 85 elements, but this solution does not depend on the discretization step.
The size of the other methods depends on the tspan and the value of h.
In general, the following relationship holds true:

$$Dim = \frac{Tspan}{h} + 1$$

So the three methods give us three vectors of dimensions:
801 for h = 0.01;
81 for h = 0.1;
17 for h = 0.5.

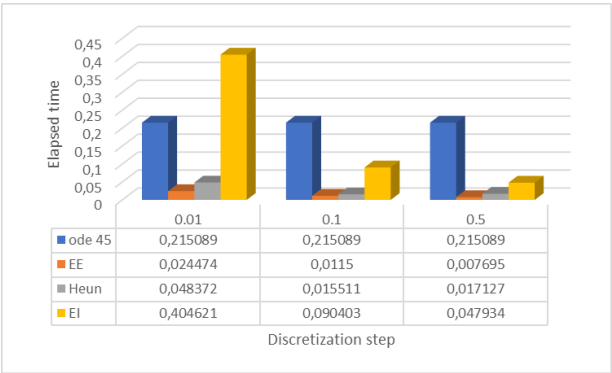


Figure.19 Elasped time

4.5. Exceptions: study of Euler Explicit solutions

Finally with Case_study_EE.m script, I evaluated the explicit methods for discretization step values at the convergence limit.
As mentioned above, the methods converge for values of h <2.
I plotted the results obtained through the implementation of Euler Explicit with three different values of h.
The solutions obtained converge but show negative concentration values (next Fig. With h = 1.5 and h = 2).
These values are unacceptable for the problem proposed, do to the fact that in physiological terms, the human body can not eliminate more heparin than it was injected.

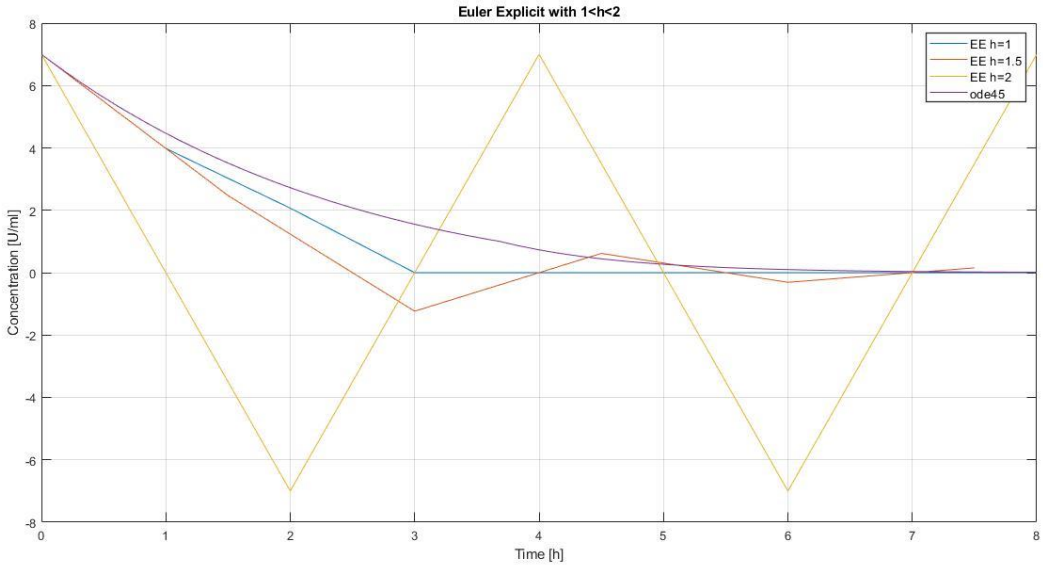


Figure.20 Different solution of Euler Explicit method

4.6. Conclusions

In conclusion, it can be argued that the implemented methods fit the proposed physiological model very well. Mathematically, Explicit methods converge for h <2 values. However, for values comprised between 1 and 2, although they are convergent, solutions are not acceptable.
The Heun method is the best method in terms of relative error for values of h <0.5, as it reaches the solution quickly and with excellent approximation. As a first approximation, on the other hand, the implicit method can be used, which has fewer restrictions on discretization step (h) values. For values of h greater than 0.5, this method can be said to have an acceptable execution time and low dimension of the solution.

Bibliography

[1] A. W. J Oates, “Heparin,” 1991.

[2] D. R. Jobes, A. J. Schwartz, N. Ellison, R. Andrews, R. A. Ruffini, and J. J. Ruffini, “Monitoring Heparin Anticoagulation and Its Neutralization,” *Ann. Thorac. Surg.*, vol. 31, no. 2, pp. 161–166, 1980.

[3] T. D. Bjornsson, K. M. Wolfram, and B. B. Kitchell, “Heparin kinetics determined by three assay methods.”

[4] A. Neilan and C. Pharmacology, “Heparin kinetics : Variables related to disposition and dosage.”

[5] “Riassunto Caratteristiche Prodotto (RCP) Clexane,” pp. 1–53, 2020.