University of Bath
Faculty of Engineering & Design

Word count: 1584

November 29, 2018

Systems Modelling & Simulation Coursework 1

Supervisor
A. Cookson

Assessor

Author's Candidate Number
10838

Contents

1 Introduction **1**

2 Part 1: Software Verification **1**
2.1 Question 1a 1
2.1.1 Introduction 1
2.2 Investigation of Time Stepping Schemes 1
2.3 Investigating the L2 Norm 3

3 Part 2: Modeling and Simulation Results **4**
3.1 Question 1 4
3.2 Question 2 5

Appendices **9**

A Flow Chart of FEM Solver **9**

B Laplace Element Matrix Code **11**

List of Figures

1 The Effect of Mesh Size on Temperature Resolution 2
2 The Effect of Mesh Size on Temperature Resolution 3
3 4
4 Layout of Skin Layers and the x Domain. XXX REF 4
5 Temperature Profiles as Time Increases. 5
6 Temperature Profiles As Algorithm Converges on $\Gamma = 1$. 7
7 Flow Chart Showing The Process Followed by The XXX BCFinder.m Script 8
8 Flow Chart Showing The Process Followed by The FEM solver StaticReactDiffSolver.m 10

1 Introduction

This paper is based on solving the transient diffusion-reaction equation given by equation 2.1.1.

$$\frac{\partial c}{\partial t} = D\frac{\partial^2 c}{\partial x^2} + \lambda c + f \tag{1}$$

2 Part 1: Software Verification

2.1 Question 1a

2.1.1 Introduction

A transient Diffusion-Reaction solver was developed and used to solve the specific transient equation give by equation 2. This is the same equation as with $\lambda = 0$ and $f = 0$.

$$\frac{\partial c}{\partial t} = \frac{\partial^2 c}{\partial x^2} \tag{2}$$

The equation is subject to the following domain, boundary conditions and initial conditions.

$$\begin{aligned} x &= [0, 1] \\ t &= [0, 1] \\ c(x, 0) &= 0 \\ c(0, t) &= 0 \\ c(1, t) &= 1 \end{aligned}$$

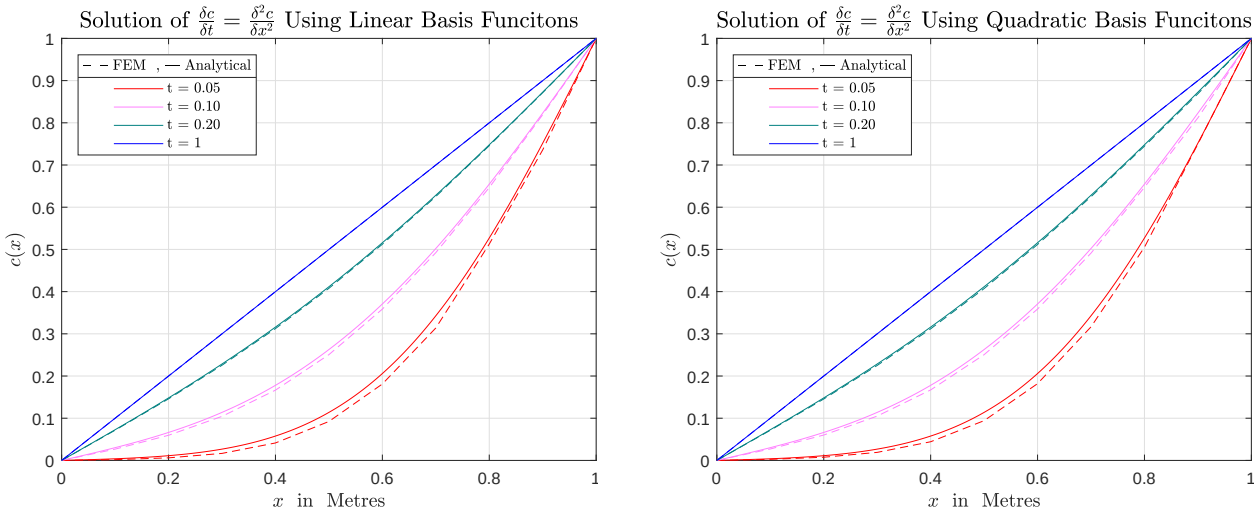
The analytical solution of equation 2 for the above conditions is given by equation 3.

$$c(x, t) = x + \frac{2}{\pi}\sum_{n=1}^{\infty}\frac{(-1)^n}{n}e^{-n^2\pi^2 t}\sin(n\pi x) \tag{3}$$

A Crank-Nicolson time stepping scheme was used with the recommended time step of $\delta t = .01$ and a 10 element mesh. The result produced using the finite element method is compared to the analytical solution for $t = 0.01,\ 0.10,\ 0.20,\ 1$ as shown by Figure 1. The FEM solver incorporates the option of quadratic basis functions and so this solution has been plotted in Figure 1b alongside the linear basis function solution shown in Figure 1a. The difference between the quadratic and linear solutions is hardly noticeable in these plots and both provide a good approximation to the analytical solution. It is clear that the FEM solutions converge on the analytical solution as time increases with the plots indistinguishable at $t = 1$ where the steady state solution has been approached.

2.2 Investigation of Time Stepping Schemes

The times stepping schemes used were Backward Euler and Crank-Nicolson. The results at $x = 0.8m$ for over the time range have been plotted both time stepping schemes at time steps of $\Delta t = 0.1s$ and $\Delta t = 0.025s$ in Figures **??** and 2 respectively. With a time step of $\Delta t = 0.1s$ the

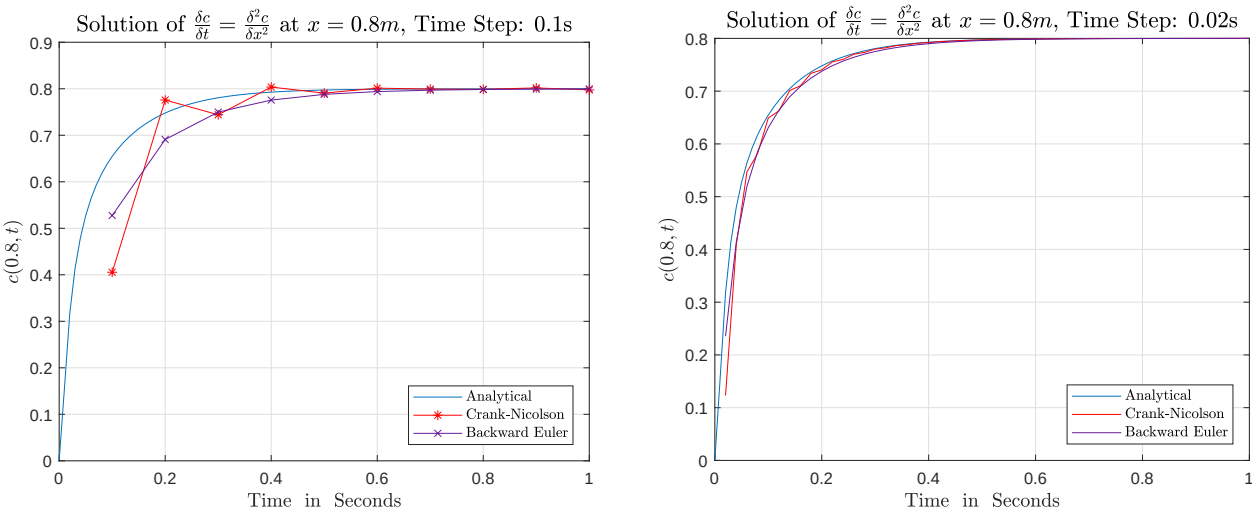


(a) Linear Basis Function Solution

(b) Quadratic Basis Function Solution

Figure 1: Comparison of FEM with Crank-Nicolson Time Stepping and Analytical Results.

Crank-Nicolson scheme is showing oscillatory solution whereas the Backward Euler scheme does not oscillate. The oscillatory response of the Crank-Nicolson reduces with time and even at the large time step is more accurate over the time domain than the Backward Euler scheme. As the time step is reduced to $\Delta t = 0.02s$ the oscillation of the Crank-Nicolson scheme is much reduced as is again clearly more accurate that the Backward Euler scheme.



(a) Time Step of 0.1s

(b) Time Step of 0.02s

Figure 2: Comparison of Crank-Nicolson and Backward Euler Time Stepping Schemes, $dx = 0.1$

2.3 Investigating the L2 Norm

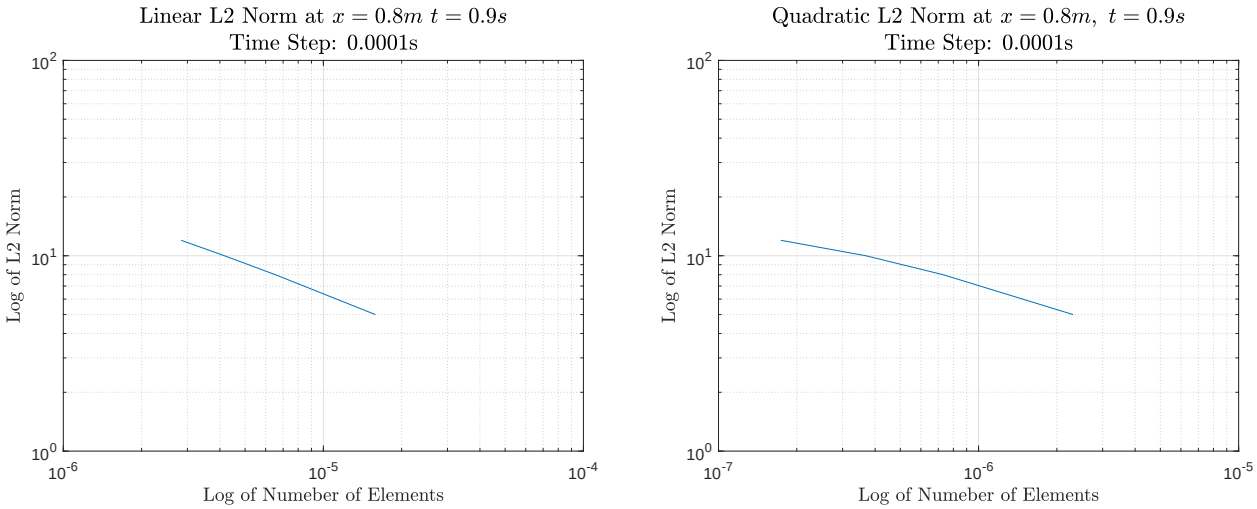
The L2 norm is the error of the FEM solution compared to the analytical solution. As the L_2 Norm is investigating the error in the spatial domain the time step had to be very small to make the time error insignificant. As a result only low resolution meshes could be tested to keep computing time reasonable. Equation 4 shows how the exact solution $CE(x)$ is approximated by the FEM solution $C(x)$ with an additional higher order terms truncated. For linear basis functions the lowest order truncated term, n, is $n = 2$, and similarly for quadratic basis functions $n = 3$.

$$CE(x) = C(x) + O(h^n) \tag{4}$$

The error $E(x)$ is $CE(x) - C(x)$ and by rewriting the higher $O(h^n)$ as Gh^n, where G is a constant, equation 4 can be written as equation 5. It can be seen that as the error converges with increasing mesh resolution the gradient of the a logarithmic plot of these should be n.

$$log(E(x)) = log(G) + n * log(h) \tag{5}$$

The L_2 norm was calculated for mesh sizes of 5 to 12 elements and plotted as shown in Figures 3a and 3b for linear and quadratic basis functions respectively. The gradients were computed to be -1.97 and -2.88 for linear and quadratic plots respectively. This shows that the L_2 norm is converging as the gradients match the expected n values of the truncated terms.



(a) Gradient = -1.97

(b) Gradient = -2.88

Figure 3: Comparison of Error Conversion for Linear and Quadratic Basis Functions.

3 Part 2: Modeling and Simulation Results

3.1 Question 1

The code developed in Part 1 was used to model heat transfer through skin. The skin is split into layers and the heat tranfer through the first three layers was investigated. The spatial domain, x, is defined by Figure 4 where the boundaries are defined as:

$$B = 0.01 \ , \quad D = 0.005 \ , \quad E = 0.00166667.$$

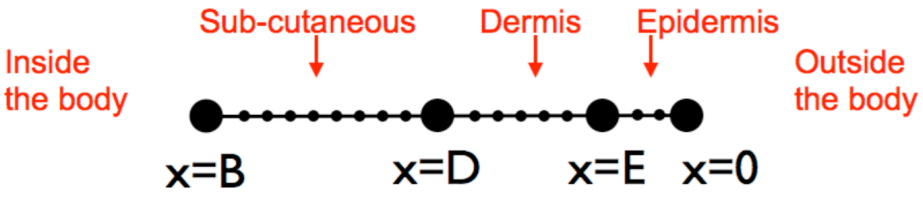


Figure 4: Layout of Skin Layers and the x Domain. XXX REF

The material properties are not consistent between layers and are defined in Table 1.

Parameter	Epidermis	Dermis	Sub-cutaneous
k	25	40	20
G	0	0.0375	0.0375
ρ	1200	1200	1200
c	3300	3300	3300
ρ_b	-	1060	1060
c_b	-	3770	3770
T_b	-	310.15	310.15

Table 1: Material Properties of Skin at Each Layer.

The governing equation of heat transfer over the x domain is given by equation 6 where parameters are not consistent between layers and are defined in Table 1.

$$\frac{\partial T}{\partial t} = \left(\frac{k}{\rho c}\right)\frac{\partial^2 T}{\partial x^2} - \left(\frac{G\rho_b c_b}{\rho c}\right)T + \left(\frac{G\rho_b c_b}{\rho c}\right)T_b \tag{6}$$

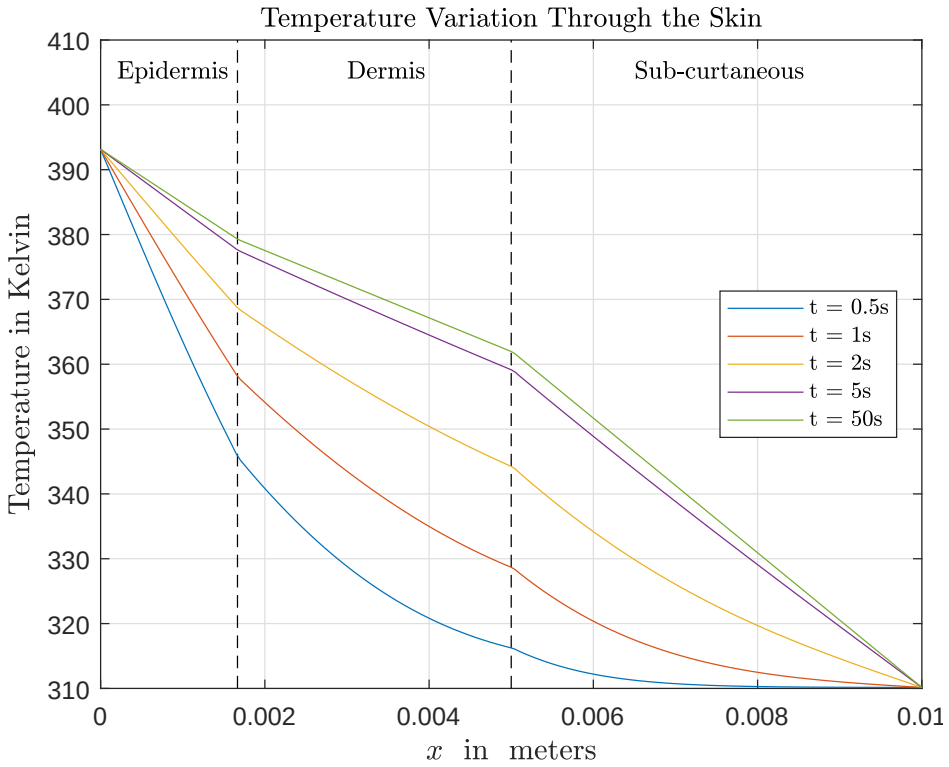


Figure 5: Temperature Profiles as Time Increases.

3.2 Question 2

The function EvalGamma.m shown in Appendix XXX, was created to return the value of Γ for a given boundary condition temperature at $x = 0$. The required value of Γ to avoid second degree burns is $\Gamma = 1$. From Part 2 Question 1 it was known that a boundary condition of $T(0,t) = 393.15K$ produced a values of Gamma many orders of magnitude greater than $\Gamma = 1$. Furthermore as the temperature which burning occurs is $T_{burn} = 317.15$ it follows that a boundary condition of $T(0,t) = 393.15K$ will produce the result $\Gamma < 1$.

Iteration	T_{high} (K)	T_{low} (K)	T_{av} (K)	$\Gamma(T_{av})$
1	394	316	355	2.05e+29
2	355	316	335.5	4.75e+09
3	335.5	316	325.75	1.13e-05
4	335.5	325.75	330.63	825
5	330.63	325.75	328.19	0.138
6	330.63	328.19	329.41	11.58
7	329.41	328.19	328.80	1.2892
8	**328.80**	328.19	**328.49**	0.4233
Result	328.8	328.5	328.6	0.74

Table 2: Result of FindBC.m XXX. at Each Iteration.

A script was made to calculate the value $T(0,t)$ to within 0.5K which would produce a result of $\Gamma = 1$ called XXX.m shown in Appendix XXX. The process followed to find the boundary condition is described by the flow chart in Figure 7. The initial high and low temperature bounds were set at $T(0,t) = 394K$ and $T(0,t) = 316K$ to respectively to give result for Γ above and below the target value of $\Gamma = 1$ as previously explained. The process took 8 iterations to find the temperature range of 0.5K which contained the boundary condition, $T(0,t) = T$, which results in $\Gamma = 1$. The range found by was $T(0,t) = 328.5K$ to $328.8K$.
It was difficult to design a more efficient algorithm such as the shooting method to determine the boundary condition due to the extreme non-linearity between Γ and $T(0,t)$. The result at each iteration of FindBC.m XXX is shown in Table 2 and gives insight into how the algorithm arrived at the solution of $T(0,t) = T$. The temperature variation with time at the Epidermis boundary has been plotted at selected iterations in Figure 6. The boundary condition for each interval in Figure 6 is given by the T_{av} for the corresponding iteration in Table 2.

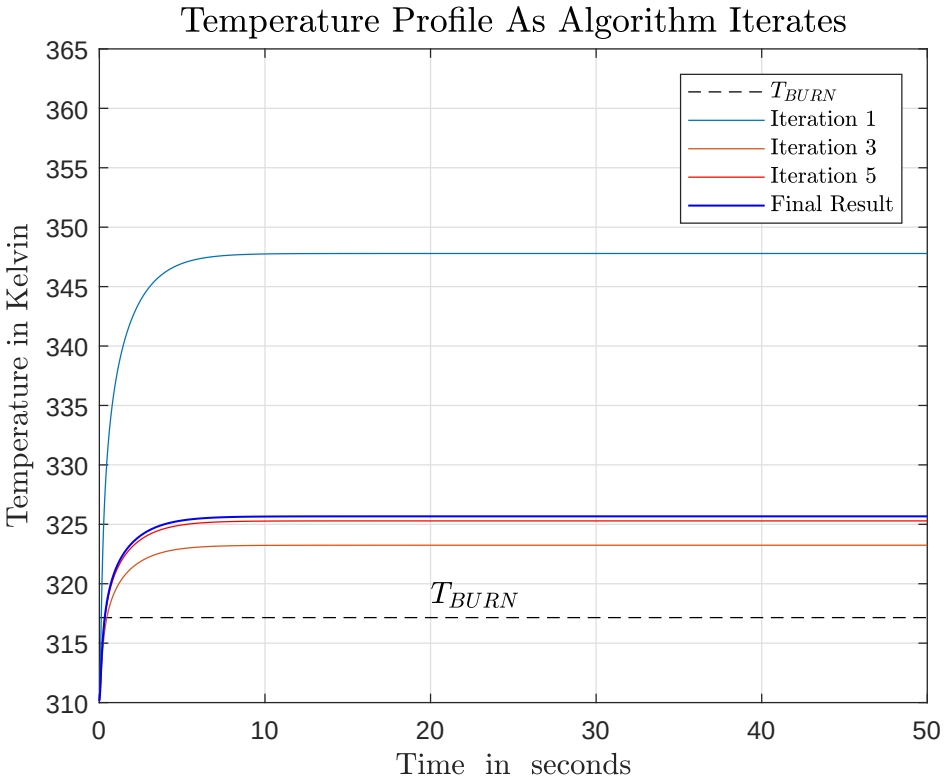


Figure 6: Temperature Profiles As Algorithm Converges on $\Gamma = 1$.

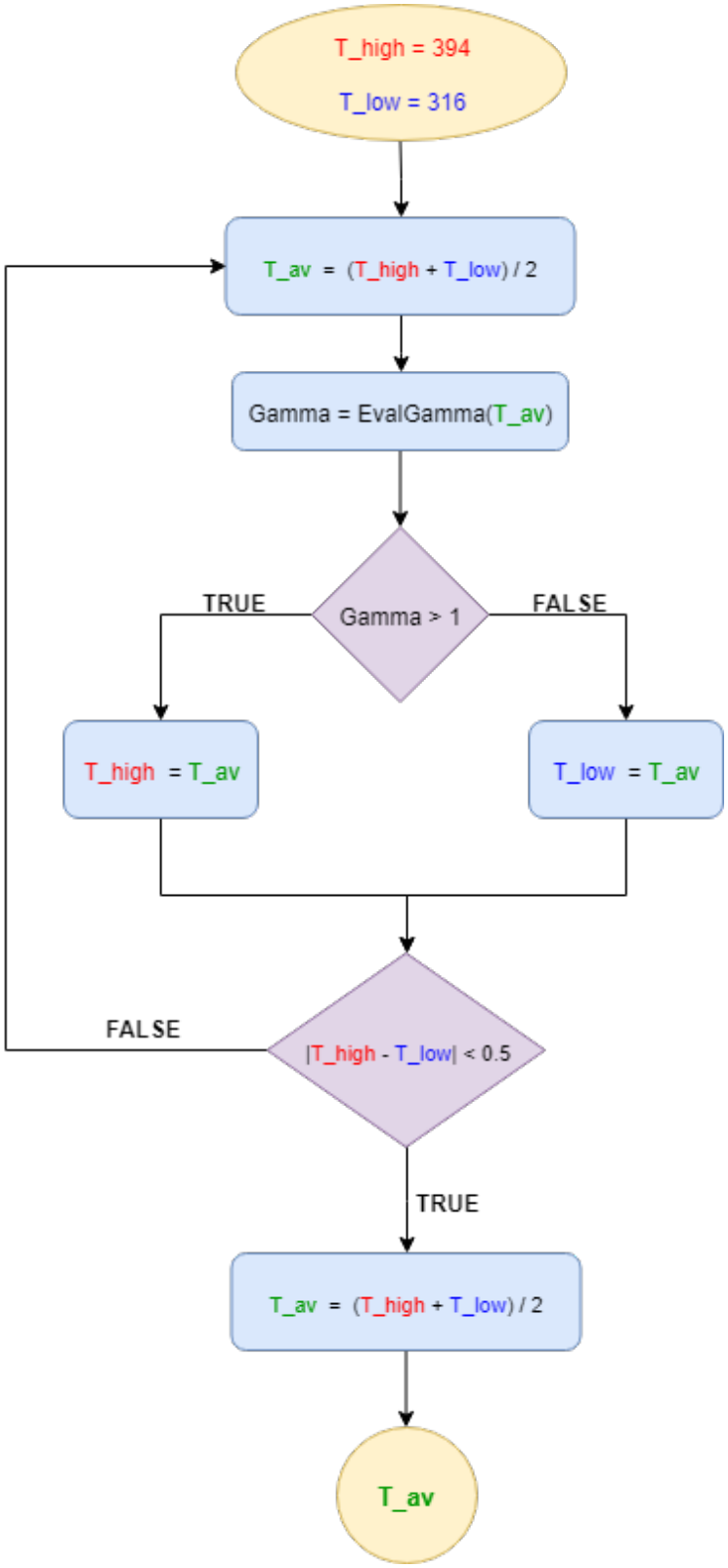


Figure 7: Flow Chart Showing The Process Followed by The XXX BCFinder.m Script

Appendices

A Flow Chart of FEM Solver

A. FLOW CHART OF FEM SOLVER

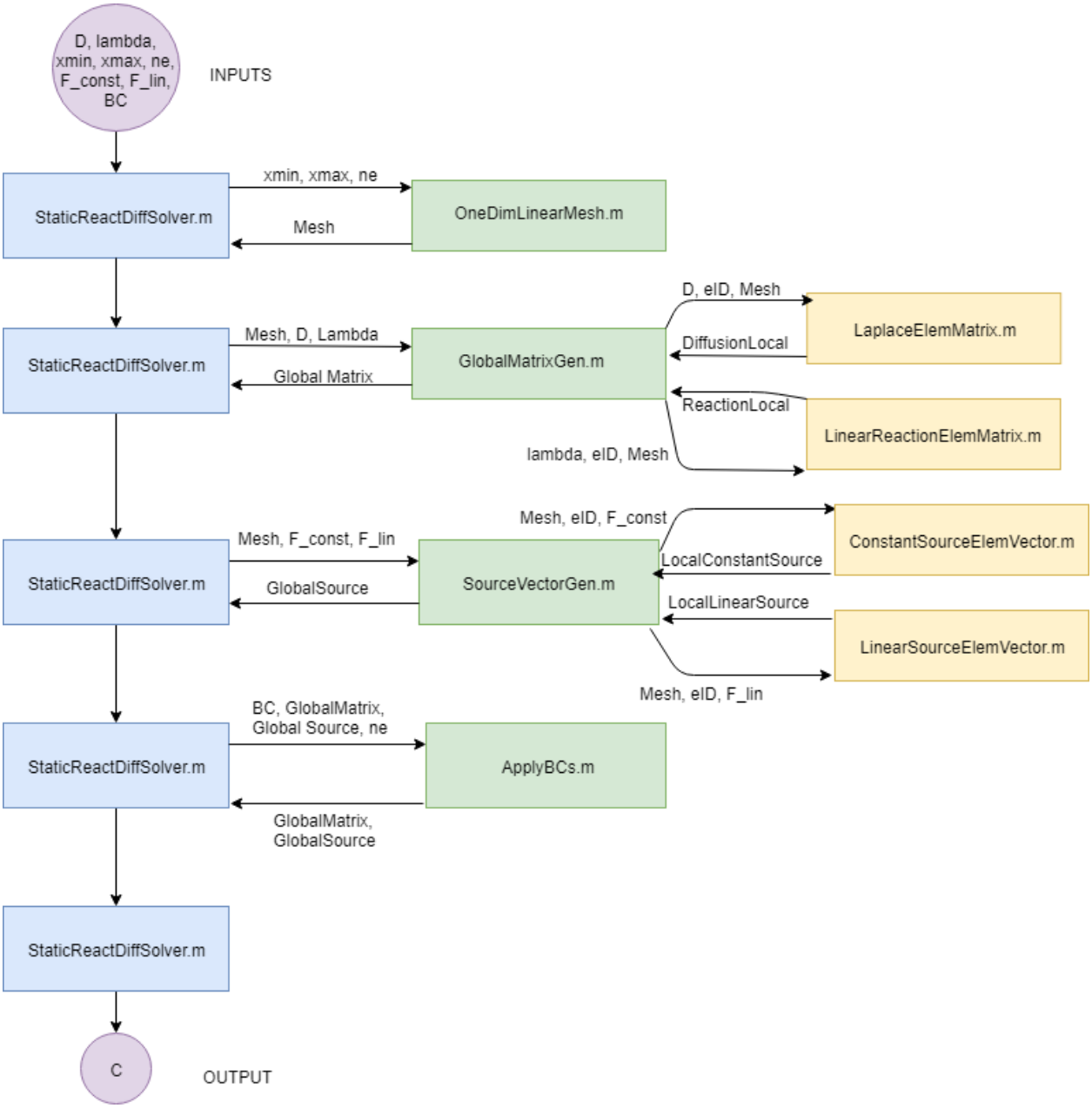


Figure 8: Flow Chart Showing The Process Followed by The FEM solver StaticReactDiffSolver.m

B Laplace Element Matrix Code

J = msh.elem(eID).J;

```
function [SqMatrix] = LaplaceElemMatrix(D, eID, msh)

SqMatrix = [0.5, -0.5; ...
            -0.5, 0.5];

%% Multiply  by (1/J) and D to get solution for the particular element
SqMatrix = (1/J) * D * SqMatrix;   %Local element matrix for element eID
end
```