PROGRAMMING PROJECTS PART 1 : ONE DIMENSIONAL BOUNDARY VALUE PROBLEM

Dharanidharan Arumugam

February, 2023

Chapter 1 Introduction

1.1 Problem statement

Develop a program to compute nodal unknowns and elemental fluxes in a one-dimensional boundary value problem.

1.2 Theory

The one-dimensional boundary value problem assumes the following differential form,

$$-\frac{d}{dx}\left(\alpha(x)\frac{dy(x)}{dx}\right)+\beta(x)\,y(x)=f(x), \qquad x_1<x<x_2 \tag{1.2.1}$$

with the essential or natural or mixed boundary conditions. Using Galerkin's approach, we can convert the differential equation into algebraic form and develop a system of equations for linear and quadratic elements as given in Eq. 1.2.2 and Eq. 1.2.3.

For linear elements,

$$\left(\frac{\bar{\alpha}}{6}\begin{bmatrix}1 & -1\\ -1 & 1\end{bmatrix}+\frac{\bar{\beta}L}{6}\begin{bmatrix}2 & 1\\ 1 & 2\end{bmatrix}+\begin{bmatrix}-c_1 & 0\\ 0 & c_2\end{bmatrix}\right)\begin{bmatrix}y_1\\ y_2\end{bmatrix}=\frac{\bar{f}L}{2}\begin{bmatrix}1\\ 1\end{bmatrix}+\begin{bmatrix}d_1\\ -d_2\end{bmatrix} \tag{1.2.2}$$

For quadratic elements,

$$\left(\frac{\bar{\alpha}}{3L}\begin{bmatrix}7 & -8 & 1\\ -8 & 16 & -8\\ 1 & -8 & 7\end{bmatrix}+\frac{\bar{\beta}L}{30}\begin{bmatrix}4 & 2 & -11\\ 2 & 16 & 2\\ -1 & 2 & 4\end{bmatrix}+\begin{bmatrix}-c_1 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & c_3\end{bmatrix}\right)\begin{bmatrix}y_1\\ y_2\\ y_3\end{bmatrix}=\frac{\bar{f}L}{6}\begin{bmatrix}1\\ 4\\ 1\end{bmatrix}+\begin{bmatrix}d_1\\ 0\\ -d_3\end{bmatrix} \tag{1.2.3}$$

Here, $\bar{\alpha}$, $\bar{\beta}$ and $\bar{f}$ are alpha, beta, and force respectively computed at the center of the element and assumed constant. L is the length of the element. These equations are incorporated mixed boundary conditions. In the case of essential boundary conditions, the parameters c and d will be zero and in the case of natural boundary conditions, the parameter c will be zero. The elemental fluxes for these elements are given in Eq. 1.2.4 and Eq. 1.2.5.

For linear elements,

$$\tau=-\frac{\bar{\alpha}}{L}\,(y_2-y_1) \tag{1.2.4}$$

For quadratic elements,

$$\tau=-\frac{\bar{\alpha}}{L^2}\,(\,x(4y_1-8y_2+4y_3)+x_1(4y_2-2y_3)-2x_2(y_1+y_3)-2x_3(y_1-y_2) \tag{1.2.5}$$

The above two equations show that the flux is constant in the case of a linear element and linear in the case of a quadratic element.

A general one-dimensional element problem analyzed using this formulation involves discretization, constructing elemental equations, assembling to a global system of solutions, enforcing boundary conditions, finding primary (displacements and temperatures) unknowns, and computing secondary unknowns (elemental fluxes – stresses and thermal flux) [Rajan, 2022]. A convergence check is also warranted to obtain solutions with better accuracies.

Chapter 2 Program implementation

2.1 Introduction

The program is required to read the specified input file from the user containing the information of alpha, beta, force, nodal fluxes, nodal coordinates, elemental properties, and elements connectivity, and generate an output file with the error norms, primary unknowns at the nodes and element fluxes. This implementation of this program is achieved in the following stages with the use of several functional and subroutine blocks in MATLAB R2022a:

- Reading and validating the input
- Renumbering of the nodes, elements, and element properties
- Storing the system stiffness matrix
- Applying boundary conditions
- Solving the system of equations
- Finding the error norms of the solver
- Computing nodal unknowns, element fluxes, and extreme values
- Writing the outputs in to text file

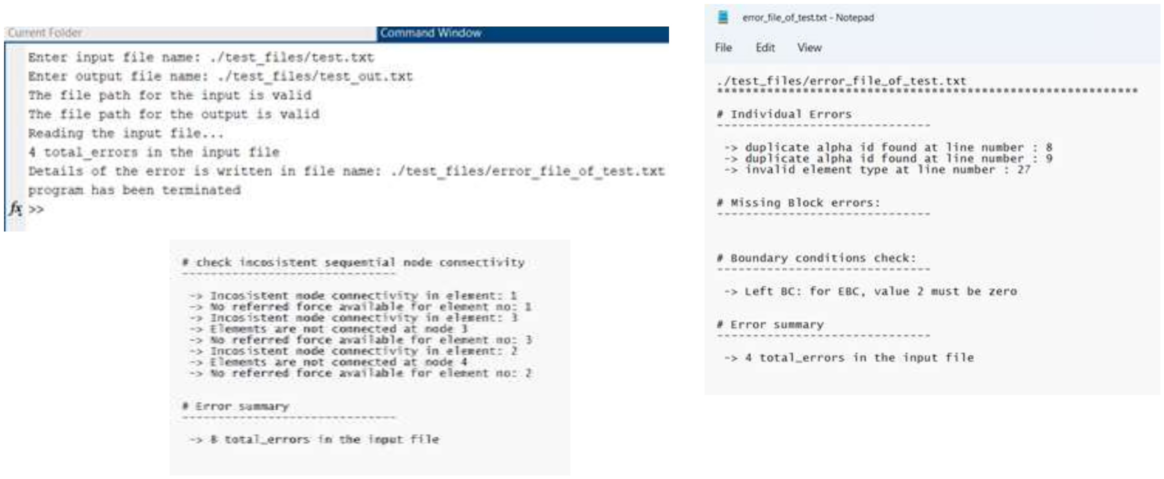
These stages are discussed in detail in the following sections:

2.2 Reading and validating the input

The inputs are read line by line and several checks are made to verify the validity of the inputs. The checks are made in two stages. In the first stage, missing data and duplicate values are checked and the user is prompted to address the raised errors. Once the errors are addressed the inputs are checked for valid boundary conditions, connectivity, and so on. The following list summarizes the checks made in the two stages,

- ➢ First stage check
    - o Missing data by checking the number of elements in the data
    - o Duplicate alpha, beta, force, and node values
    - o Improper data types
    - o Missing entire block details
    - o The use of “*end” statement
    - o Check whether the element type is valid
    - o Check whether the problem is ill-posed
- ➢ Second stage check
    - o Availability of all the referenced properties
    - o Validity of boundary conditions
    - o Availability of all the referenced nodes
    - o Check whether all the elements are connected

All these errors are recorded in the detail and written in an error file. The total number of errors is also displayed to the user. Fig. 1 shows a few generated error files for sample input cases along with the messages displayed at the command prompt.



```
Enter input file name: ./test_files/test.txt
Enter output file name: ./test_files/test_out.txt
The file path for the input is valid
The file path for the output is valid
Reading the input file...
4 total_errors in the input file
Details of the error is written in file name: ./test_files/error_file_of_test.txt
program has been terminated
fx >>
```

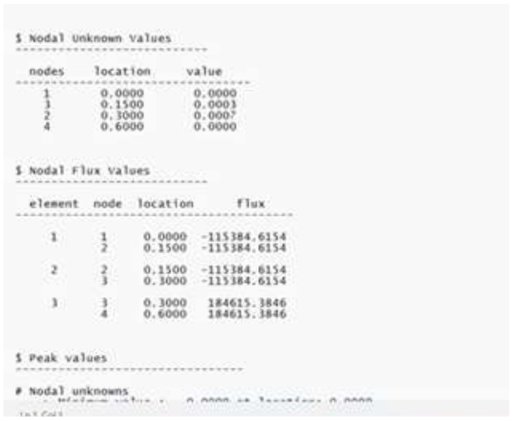
```
./test_files/error_file_of_test.txt
**********************************************************
# Individual Errors
-------------------------------

 -> duplicate alpha id found at line number : 8
 -> duplicate alpha id found at line number : 9
 -> invalid element type at line number : 27

# Missing Block errors:
-------------------------------

# Boundary conditions check:
-------------------------------

 -> Left BC: for EBC, value 2 must be zero

# Error summary
-------------------------------

 -> 4 total_errors in the input file
```

```
# check incosistent sequential node connectivity
-------------------------------

 -> Incosistent node connectivity in element: 1
 -> No referred force available for element no: 1
 -> Incosistent node connectivity in element: 3
 -> Elements are not connected at node 3
 -> No referred force available for element no: 3
 -> Incosistent node connectivity in element: 2
 -> Elements are not connected at node 4
 -> No referred force available for element no: 2

# Error summary
-------------------------------

 -> 8 total_errors in the input file
```

Fig. 1: Error messages displayed and the error files generated for sample cases

2.3 Renumbering of the nodes, elements, and element properties

Any non-sequential and non-consecutive ids of alpha, beta, force, and element or node numbers are renumbered for programming convenience and efficient storage. The final outputs are however displayed with the original input numbering. The renumbering of nodes is required to keep the half band width size 2 for the linear elements ane 3 for the quadrilateral elements. Functional blocks are written to map the original numbering to the consecutive numbering. Fig. 2 shows the output for a case where the nodal numbers are not consecutive. The program renumbers the node for efficient storage and displays the nodal outputs with the original node numbers.



```
$ Nodal Unknown Values
-------------------------

  nodes    location     value
-------------------------------
    1       0.0000      0.0000
    3       0.1500      0.0003
    2       0.3000      0.0007
    4       0.6000      0.0000

$ Nodal Flux Values
-------------------------

  element  node  location       flux
----------------------------------------

     1      1     0.0000   -115384.6154
            2     0.1500   -115384.6154

     2      2     0.1500   -115384.6154
            3     0.3000   -115384.6154

     3      3     0.3000    184615.3846
            4     0.6000    184615.3846

$ Peak values
------------------------------

# Nodal unknowns
```

Fig. 2: Output file of an input problem with non-consecutive node numbering

2.4 Storing the system stiffness matrix and force/flux vector

The system stiffness matrix is stored as half bandwidth matrix, in the case of the linear element the half bandwidth (w) is 2 and in the case of the quadratic element, the half bandwidth is 3. So the total integer storage requirement of the system with n effective degrees of freedom is $2nw$. In these particular 1DBVP cases with consecutive numbering, the half-bandwidth storage is superior to the skyline storage since the stiffness matrix is compact and banded. The following algorithm is used to build the system matrix,

1. Loop through the elements, $i = 1 \; to \; nE$
2. Extract, α, β , f, L, node numbers, and element type
3. Compute local elemental matrices in half bandwidth form using Eq. 1.2.2 and Eq. 1.2.3 without the mixed boundary constants, since it is imposed before the system solving
4. Add elemental matrices and forces with the following algorithmic format by initiating the K and F from zero,

$$K^*_{ij,new} = K^*_{ij,old} + K^{*e}_{ij}, i = n^e_i \; to \; n^e_j \; and \; j = 1 \; to \; w$$
$$F^*_{i,new} = F^*_{i,old} + F^{*e}_i, i = n^e_i \; to \; n^e_j$$

Here, $n^e_i \; and \; n^e_j$ are starting and ending node numbers of the element, w is the half bandwidth, e indicates the element, and * indicates the half bandwidth matrix.

5. Add nodal fluxes F_X to F

$$F^*_{i,new} = F^*_{i,old} + F_{X,i} \; for \; i \; where \; F_{X,i} \neq 0$$

Here, $n^e_i \; and \; n^e_j$ are starting and ending node numbers of the element, w is the half bandwidth, e indicates the element, and * indicates the half bandwidth matrix.

2.5 Applying boundary conditions

After assembling the system matrix, left and right boundary conditions are imposed. When the boundary condition type is EBC, either homogenous or non-homogenous the row corresponding to the boundary node is removed after the force values in the left hand are modified as depicted in Eq. 2.5.1.

For left BC, $$F_{i,new} = F_{i,old} + K^*_{1,i} \, D_l \, , i = 1 \; to \; w$$

For right BC, $$F_{i,new} = F_{i,old} + K^*_{i,n-i+1} D_r \, , i = n-1 \; to \; n-w+1 \qquad (2.5.1)$$

2.6 Solving the system of equations

The system of equations is solved with the Cholesky factorization and implemented for the half bandwidth matrix with the following algorithmic steps:

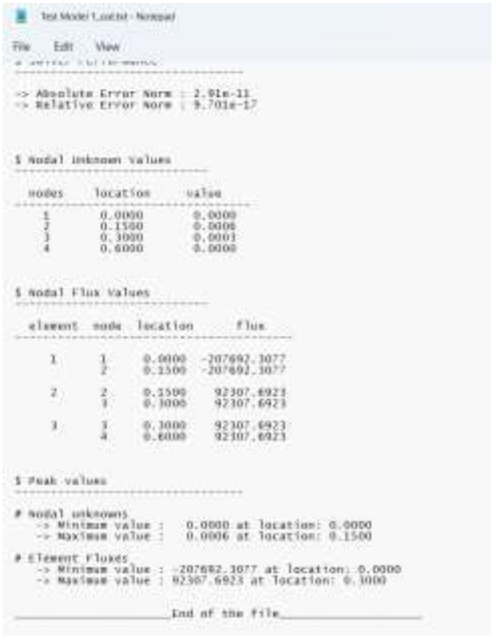
- Finding diagonal components ($\widehat{D}$) by looping through $i = 1 \; to \; n$. The diagonal components D are stored in the diagonal components of the K matrices. In our case, for the half bandwidth matrix, K^* matrix, the following algorithmic implementation is used,

  $K^*_{i,1} = K^*_{i+1} - \sum_{\max(i-w+1,1)}^{i-1} K^{*^2}_{p,i-p+1} K^*_{p+1}$, if $K^*_{i,1} < tol$, matrix ill-conditioned, exit (2.6.1)
- For lower diagonal components, (L_{ji}):
- Find Q by forward substitution

  Find D by a backward substitution

2.6 Computing outputs and writing into the text file

After the system is solved, the error norms are computed from the effective half-bandwidth matrix by letting $R = K_r D_r - F_r$. The subscript r indicates the reduced matrix after the application of the boundary conditions. The relative norms obtained for the suite of test problems were in the order of 10^{-15}. Element fluxes are then calculated through the stored element matrices. Finally, all the outputs along with the summary of the inputs are written into an output file. A snapshot of an output file is shown in Fig. 3



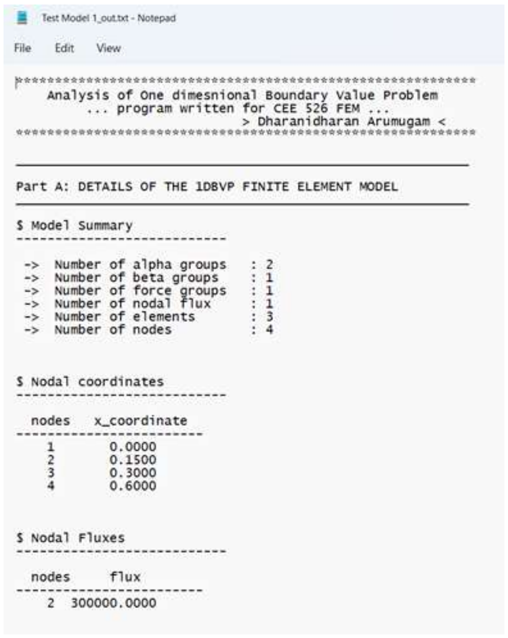
```
----------------------------------

-> Absolute Error Norm : 2.91e-11
-> Relative Error Norm : 9.701e-17


$ Nodal Unknown Values
------------------------------

  nodes    location      value
------------------------------------
    1       0.0000       0.0000
    2       0.1500       0.0006
    3       0.3000       0.0003
    4       0.6000       0.0000

$ Nodal Flux Values
------------------------------

  element  node  location       flux
------------------------------------------

     1       1    0.0000  -207692.3077
             2    0.1500  -207692.3077

     2       2    0.1500    92307.6923
             3    0.3000    92307.6923

     3       3    0.3000    92307.6923
             4    0.6000    92307.6923


$ Peak values
----------------------------------

# Nodal unknowns
   -> Minimum value :   0.0000 at location: 0.0000
   -> Maximum value :   0.0006 at location: 0.1500

# Element Fluxes
   -> Minimum value : -207692.3077 at location: 0.0000
   -> Maximum value : 92307.6923 at location: 0.3000

______________________End of the file____________________
```

Fig. 3: Snapshot of the output file generated by the program

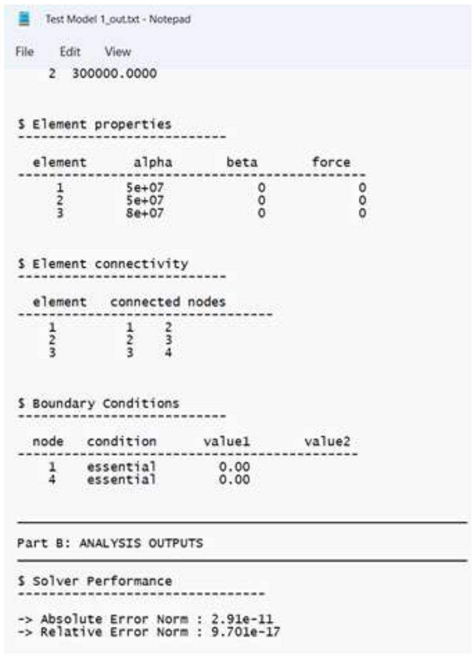
Chapter 3 Results

3.1 Summary of the results

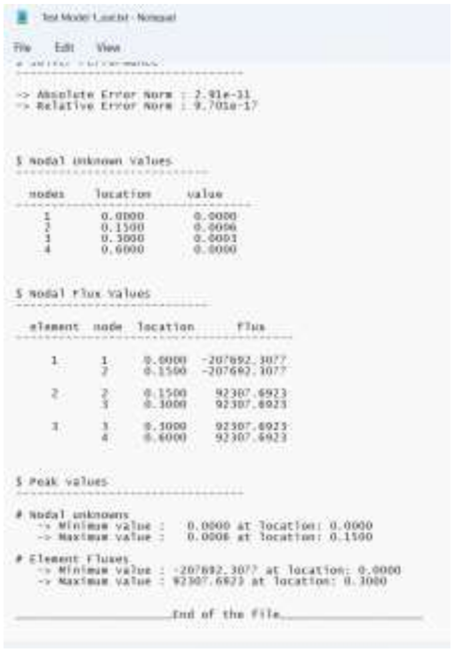
The developed 1DBVP program is tested on the three model problems given. Fig. 4 shows the output files generated for test problem 1.



```
**************************************************************
    Analysis of One dimesnional Boundary Value Problem
         ... program written for CEE 526 FEM ...
                           > Dharanidharan Arumugam <
**************************************************************

____________________________________________________________

Part A: DETAILS OF THE 1DBVP FINITE ELEMENT MODEL
____________________________________________________________

$ Model Summary
----------------------------

 ->  Number of alpha groups   : 2
 ->  Number of beta groups    : 1
 ->  Number of force groups   : 1
 ->  Number of nodal flux     : 1
 ->  Number of elements       : 3
 ->  Number of nodes          : 4


$ Nodal coordinates
----------------------------

  nodes   x_coordinate
-------------------------
    1       0.0000
    2       0.1500
    3       0.3000
    4       0.6000


$ Nodal Fluxes
----------------------------

  nodes     flux
-------------------------
    2   300000.0000
```



```
  2  300000.0000


$ Element properties
----------------------------

  element      alpha       beta       force
----------------------------------------------
     1        5e+07          0            0
     2        5e+07          0            0
     3        8e+07          0            0


$ Element connectivity
----------------------------

  element   connected nodes
----------------------------------
    1          1    2
    2          2    3
    3          3    4


$ Boundary Conditions
----------------------------

  node   condition      value1       value2
----------------------------------------------
    1    essential       0.00
    4    essential       0.00

____________________________________________________________

Part B: ANALYSIS OUTPUTS
____________________________________________________________

$ Solver Performance
---------------------------------

-> Absolute Error Norm : 2.91e-11
-> Relative Error Norm : 9.701e-17
```



```
---------------------------------

-> Absolute Error Norm : 2.91e-11
-> Relative Error Norm : 9.701e-17


$ Nodal Unknown Values
----------------------------

  nodes     location      value
----------------------------------
    1        0.0000       0.0000
    2        0.1500       0.0006
    3        0.3000       0.0003
    4        0.6000       0.0000


$ Nodal Flux Values
----------------------------

  element  node  location       flux
----------------------------------------

     1       1    0.0000  -207692.3077
             2    0.1500  -207692.3077

     2       2    0.1500    92307.6923
             3    0.3000    92307.6923

     3       3    0.3000    92307.6923
             4    0.6000    92307.6923


$ Peak values
---------------------------------

# Nodal unknowns
   -> Minimum value :   0.0000 at location: 0.0000
   -> Maximum value :   0.0006 at location: 0.1500

# Element Fluxes
   -> Minimum value : -207692.3077 at location: 0.0000
   -> Maximum value : 92307.6923 at location: 0.3000

______________________End of the File______________________
```

Fig. 4 Output file generated for test model 1 problem

3.2 Comparison of the outputs 1DBVP software

The following table shows the summary of the results obtained by my program

Table 1: Summary of the results for test model 1

	Value	location
Maximum Nodal unknown	0.0006231	0.1500
Minimum Nodal unknown	0.00	0.0000
Maximum element flux	92307.6923	0.3000
Minimum element flux	-207692.3077	0.0000

Table 2: Summary of the results for test model 2

	Value	location
Maximum Nodal unknown	9.333e-07	4.0000
Minimum Nodal unknown	0.00	0.0000
Maximum element flux	0.00	4.0000
Minimum element flux	-20.0000	0.0000

Table 3: Summary of the results for test model 3

	Value	location
Maximum Nodal unknown	304.7619	0.6000
Minimum Nodal unknown	20.0000	0.0000
Maximum element flux	12380.9524	0.0000
Minimum element flux	12380.9524	0.4500

3.2 Program best features

The following are the best features of the developed program:

- Robust checking includes checking for duplicates, missing data, invalid connections and so on
- Storing the system matrix as half bandwidth matrix
- Cholesky factorization for system solving
- Renumbering for storage efficiency

References

S. D. Rajan, "Finite Element Analysis for Engineers," *online printing,* 2022.