Hochschule München University of Applied Sciences

Numerical model of one-dimensional, transient heat conduction in a fin

Assignment two of three

Andreas Wenger | Vinzenz Götz
44267018 | 25173223

Prof. Roberto Nunhez

WS 23/24

Contents

List of Figures

1. Code Review

As in assignment one, the model building of the problem has already been conducted in the statement of the task. Since the solution of the problem is also similar to assignment one, big parts of the code are similar.

The biggest changes in are:

- A time loop to solve the system of linear equations for each time step
- Code to consider the influence of Δt
- Inclusion of a surface plot of the Temperature over time and length
- Plot of a residual
- Consideration of Δt and density in the source terms
- Consideration of Δt and density in the system matrix

1.1 Time loop

```
1    for j = 1:timesteps
2        T = A\b;
3        T_time(j,:) = T;
4        b = sourceTerms(meshSize(i), L, P, k, h, A_t, T_ar, T_w, cp, M_p0,
   T, dt);
5        res(j) = T_last-T(end);
6        T_last = T(end);
7    end
```

Listing 1.1: Time loop implementation

As seen in Listing 1.1, a further for loop was implemented in which the system of linear equations is solved in line 2. In line 3, the Temperature over the length is written into the j-th row of T_time, the source terms are recalculated based on the new Temperature in each cell in line 4 and in line 5 the j-th entry of the residual vector is calculated. Line 6 saves the last Temperature for the next calculation of the residual.

1.2 Comparison of time step size

```
%initialize a vector to store real time and residual for each time step
%size, dimension 1 -> number of time step sizes investigated, dimension 2
%-> maximum amount of time steps, dimension 3 -> storing real time in
% 1, residual in 2
timeRes = zeros(length(dt),endTime/min(dt),2);

for t = 1:length(dt)
    timesteps = endTime/dt(t);
```

Listing 1.2: Iteration over different time step sizes

To compare the influence of the time step size on the transient simulation, a further for loop has been added around the whole code to iterate over different time step sizes as seen in Listing 1.2. Here, a three dimensional array is initialized to hold physical time and the change in Temperature per second on the last point for each time step size.

```
    % giving res to the time step comparison vector
    timeRes(t, 1:length(res), 2) = abs(res)/dt(t);
    timeRes(t, 1:length(res), 1) = dt(t):dt(t):endTime;

    figure(6)
    semilogy(timeRes(t, :, 1), timeRes(t, :, 2))
```

Listing 1.3: Comparison of the residual for different time step sizes

1.3 Surface plot of T

```
1 figure(5)
2 surf(o,p,T_time);
3 ylim([0 200]);
4 ylabel("Time [s]");
5 xlabel("x [m]");
6 zlabel("T [C]");
```

Listing 1.4: Plotting T over x and t

Listing 1.4 shows the implementation of a plot of Temperature over time and x. In line 1 a new figure with id 5 is created, then filled with a surface plot in line 2. Line 3, 4 and 5 set labels for the axes and limits on the time axis.

1.4 Plotting the residual

As already seen in Listing 1.1, a residual vector is filled every time step. This residual is then plotted over time for every mesh size.

1.5 Change to the source terms

Since the source terms are now dependent on the temperature in each cell and the time step, they are updated each time step. The updated code reflects this dependency as seen in Listing 1.5 line 9. Also sourceTerms now takes the temperature in each cell, the density and the time step as an input.

```
1  function b = sourceTerms(meshSize, L, P, k, h, A_t, T_ar, T_w, cp, M_p0, T0
       , dt)
2      % initialize b
3      b = ones(meshSize,1);
4      % define dx
5      dx = L/meshSize;
6      %define constant m
7      n = h*P/(cp*A_t);
8      % set values of the entries
9      b = b*n*T_ar*dx+(M_p0*T0)/dt;
10     %change wrong values
```

```
    b(1) = n*T_ar*dx+(M_p0*T0(1))/dt+(2*k*T_w)/(cp*dx);
    if meshSize > 1
        b(meshSize) = (M_p0*T0(meshSize))/dt+(2*k*T_ar)/(cp*(2*k+h*dx))+n*
    T_ar*dx;
    end
end
```

Listing 1.5: Updated sourceTerms function

1.6 Change to the system matrix

The system matrix is now dependent on the time step. This is an additional input argument of the function. It is implemented analogous to sourceTerms.

2. Investigation of mesh size

Mesh size is investigated for a large amount of time steps and a small time step Δt. This is so that a time-independent solution is reached. The time step chosen was $\Delta t = 0.01s$, the end time $t_{end} = 10^4 s$. This is equal to 10^6 iterations. The mesh sizes chosen were 10, 13, 16, 19, 22, and 25 mesh elements. The error over the length at $t = 10^4 s$ can be seen in Figure 2.1.

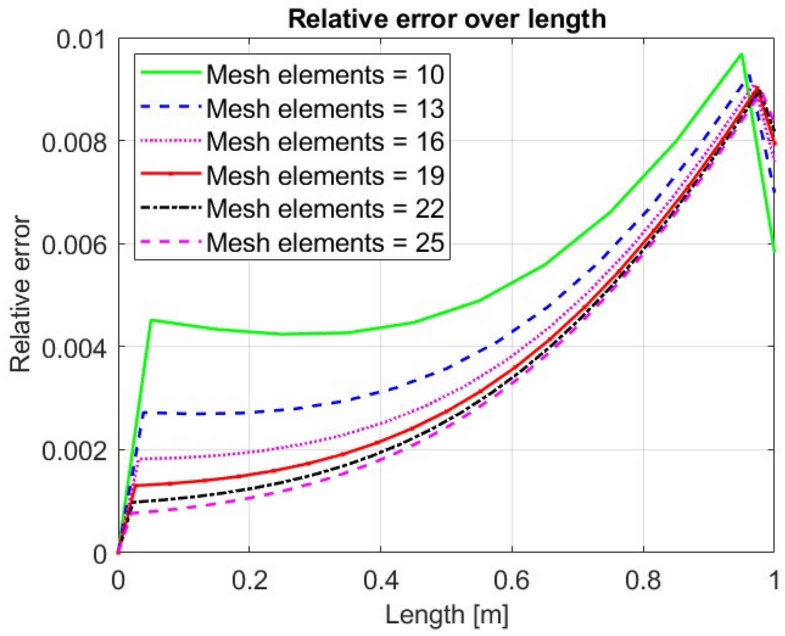

Figure 2.1: Relative error to the transient solution over the length of the fin at $t = 10^4 s$

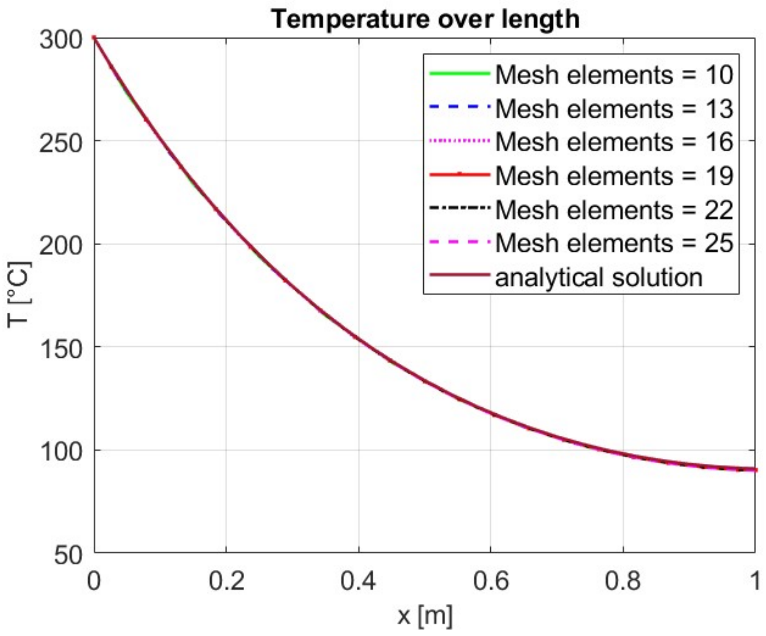

Figure 2.2: Temperature distribution of the transient solution over the length of the fin at $t = 10^4 s$ and the analytical solution

The final temperature distribution as well as the analytical, steady-state solution can be seen in Figure 2.2.

As seen in Figure 2.1 and 2.2, the distribution does not depend on mesh size at the chosen time step and end times since the error lies below 1 percent for all meshes, the distributions indistinguishable from the analytical solution. A mesh size of $n = 10$ elements is therefore chosen for further investigations.