Coupled Problems
Computer practice

Martí Burcet Rodríguez

June 1, 2016

1 Introduction

A MATLAB code is provided that solves a 1D heat transfer equations using finite elements. The first task done was to adapt the code that solves the problem in one domain to a problem in two subdomains solved first in a Monolithic way and in an iterative way then. The new adapted codes are called *main_monolithic.m* and *main_iterative.m* respectively.

2 Testing the codes

In this section some cases have been tested to see the differences in the performance of the different methods and the influence of some parameters in the final solution.

1. **Task 1**

   For the first example tested the single heat transfer problem is used in the domain $[0, 1]$, and with Dirichlet boundary conditions at both ends $u(x = 0) = u(x = 1) = 0$.

   To see the influence of the thermal diffusion coefficient , a case with three different values ($_1 = 1, \kappa_2 = 5, \kappa_3 = 10$) has been tested. As seen in Figure 1 the higher the thermal diffusion coefficient the less the temperature in the interior of the bar because more and more heat is dissipated in the bar.

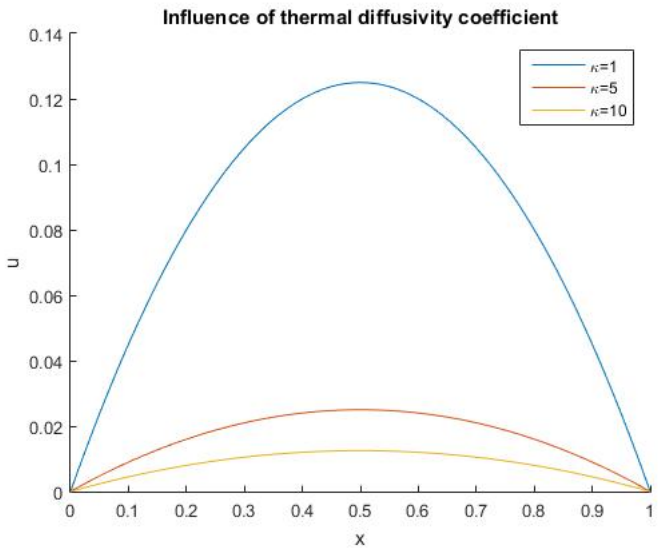

Figure 1: Influence of the thermal diffusion coefficient

Now the effect of the source term has been checked for the single heat transfer problem. Again three different values of the source term ($s_1 = 1, s_2 = 5.s_3 = 10$) have been plot in the same figure (2) to compare between solutions. Thermal diffusion coefficient has been set to $\kappa = 1$ for the three cases to don't disturb the results. It is seen as it was expected that the higher the source term in the bar, the higher the temperatures inside the bar will be.

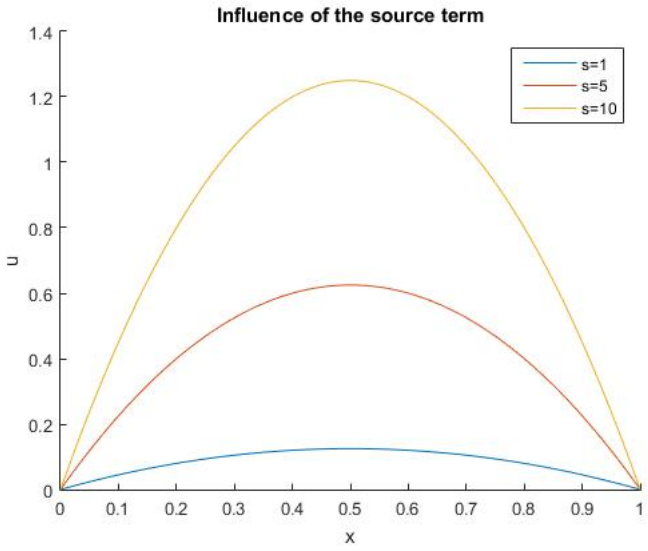

Figure 2: Influence of the thermal diffusion coefficient

Now a convergence analysis is performed to see how the solution converges to the right solution as the number of elements is increased. The maximum temperature has been taken as reference value for this analysis. To do so the analytical solution of the heat transfer equation has been calculated, which easily states that the maximum temperature is given at the mid point with a value in our case of $u(x = 0.5) = \frac{1}{8} = 0.125$.

The convergence analysis show an oscillating pattern (Figure 3) for the error in the maximum value of the temperature: for even number of elements the solution is computed very accurately because there is an element in the mid point whereas for odd meshes there is none and the error is significantly higher.

Therefore the solution of this problem with the single domain FEM approach is mesh dependent, and even for this small and easy problem the errors are very small also for odd meshes, this is not an optimal scheme and can cause problems in some situations.

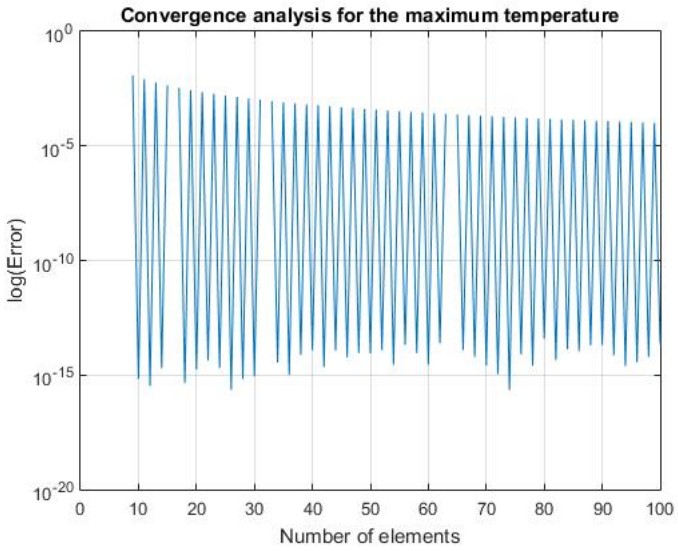

Figure 3: Convergence analysis with the maximum temperature.

2. **Task 2**

   Now two independent heat transfer problems are solved with $\kappa = 1, s = 1$ in the domains [0,0.25] and [0.25,1]. The space has been discretized with 100 elements.

   As seen in Figure 4 the solution of two independent problems with fixed Dirichlet boundary conditions in the left and right boundaries respectively and with no prescription on neither fluxes nor temperatures in the interface is discontinuous. Therefore it is clear that it is not possible to solve independent problems without any kind of communication between them through transmission conditions and get a smooth and continuous solution between the subdomains. So if we want to solve a problem in separate domains we will need to use other methods such as monolithic, partitioned or others.

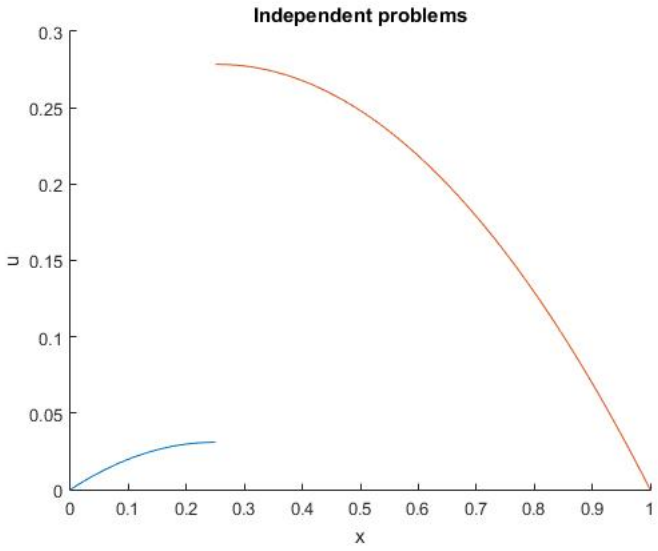

Figure 4: Solution of the independent problems

3. **Task 3**

Now the same problem of Task 2 has been solved also with two domains but on a monolithic way instead of separately problems. The boundary conditions imposed have been $u(x = 0) = u(x = 1) = 0$ and no prescriptions on neither fluxes nor temperatures on the interface between subdomains have been imposed.

What *HP_SolveMonolithic.m* does is to create the system matrices for both subdomains and assemble in a global "blocky" matrix that contains equations and unknowns for both subdomains. The interface node in this case is the only value that has shared information for both subdomains, so conitnuity of the unknown is imposed directly in this way.

The assemly of the matrices and unknowns and the solution of the global system in a monolithic way is done as follows in the code *HP_SolveMonolithic.m*:

```
npoin1 = HeatProblem.Data.nelem+1;
npoin2 = HeatProblem2.Data.nelem+1;

%Monolithic system
KMono = zeros(npoin1+npoin2-1);
FMono = zeros(npoin1+npoin2-1,1);

KMono(1:npoin1,1:npoin1) = HeatProblem.Matrices.K;
```

```
FMono(1:npoin1) = HeatProblem.Matrices.F;

KMono(npoin1:npoin1+npoin2-1,npoin1:npoin1+npoin2-1)
    = KMono(npoin1:npoin1+npoin2-1,npoin1:npoin1+
    npoin2-1) + HeatProblem2.Matrices.K;
FMono(npoin1:npoin1+npoin2-1) = FMono(npoin1:npoin1+
    npoin2-1) + HeatProblem2.Matrices.F;

%Solve Monolithic
UMono = KMono\FMono;
```

The resultant graph (Figure 5)for the monolithic approach is a continuous distribution of temperatures with the same shape as the single domain problem with the same parameters solved in Task 1 (Figure **??**). Although the solution is good with this method a large system is needed to be solved what for larger problems can be a limitation. Therefore to do further another a method that could be solved by subdomains with few communication between them is needed.

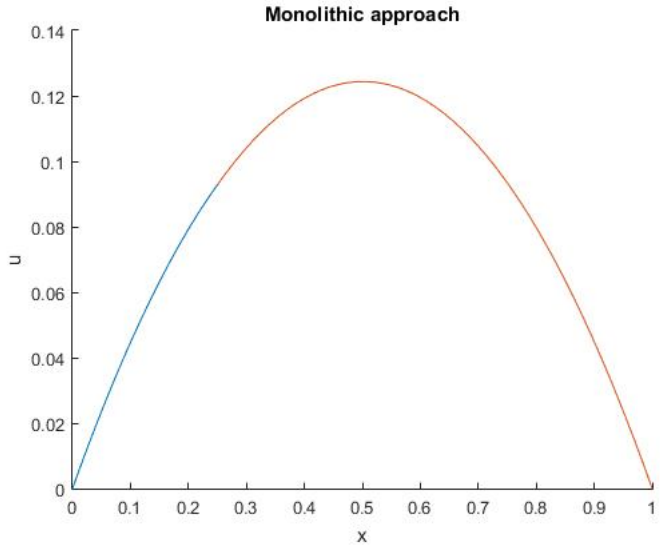

Figure 5: Solution with monolithic approach

If now instead of solving in both subdomains with the same thermal diffusion coefficient (κ) we give different pairs of values, the solution also changes. As seen in Figure 6 even though the value of the temperature is the same in the interface node (because as we said the solution is done in one single matrix that assembles equations for both subdomains), the function on the interface is not differentiable because there is a sharp

change in the distribution.

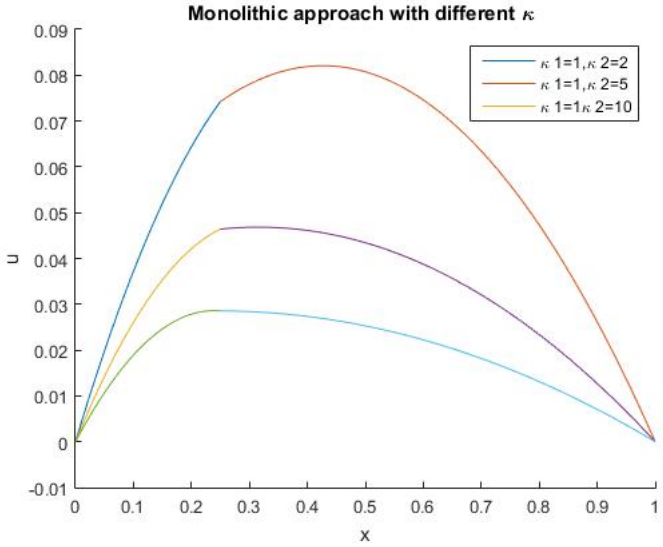

Figure 6: Solution with monolithic approach with different κ in both subdomains.

This is a consequence of only imposing the value of the unknown to be the same on the interface but not the fluxes. Therefore the first derivative with respect position of the temperature (the gradient) hasn't to be the same on the interface and actually it is shown that for different values of κ the gradients are not the same. We know from the analytical solution and from experience that not only the temperature but also the gradient has to be the continuous across an interface, so the monolithic approach is not useful to solve problems without imposing the fluxes on the interface.

So a method that can result with continuous and continuous derivatives by imposing also the fluxes is needed, which is the iterative approach explained in the next task.

4. **Task 4**

The iterative scheme has been implemented transferring the fluxes (Neumann) in the left domain and the temperatures (Dirichlet) in the right one. For the first example analysed the diffusion coefficient has been left to $\kappa = 1$.

A convergence analysis has been conducted with a tolerance in the temperatures in the interface of 1×10^{-8}. As seen in Figure 7 the algorithm converges to the right solution in 16 iterations with quadratic convergence.

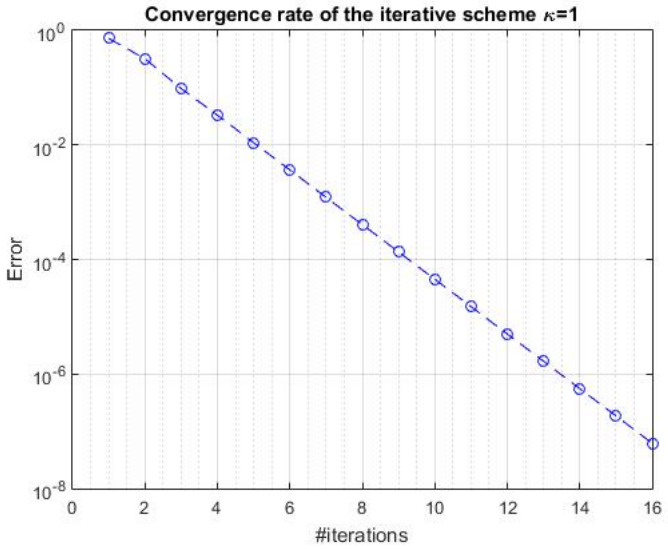

Figure 7: Convergence analysis of the iterative scheme.$\kappa = 1$

If now the thermal diffusion parameter is changed to $\kappa = 100$ in the first domain the convergence is obtained faster (Figure 8), in only 4 iterations. This is because the profile of temperatures changes faster for higher diffusion coefficients so it also converges faster.

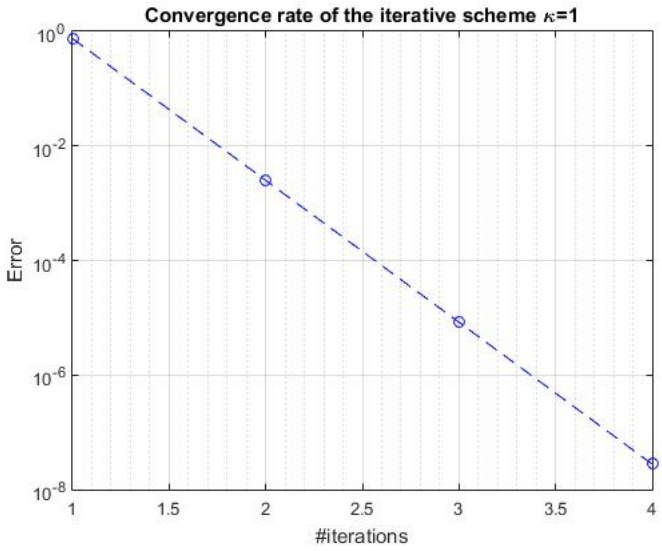

Figure 8: Convergence analysis of the iterative scheme.$\kappa = 100$

On the other hand if $\kappa = 0.01$ in the first domain the algorithm doesn't even converge as seen in Figure 9

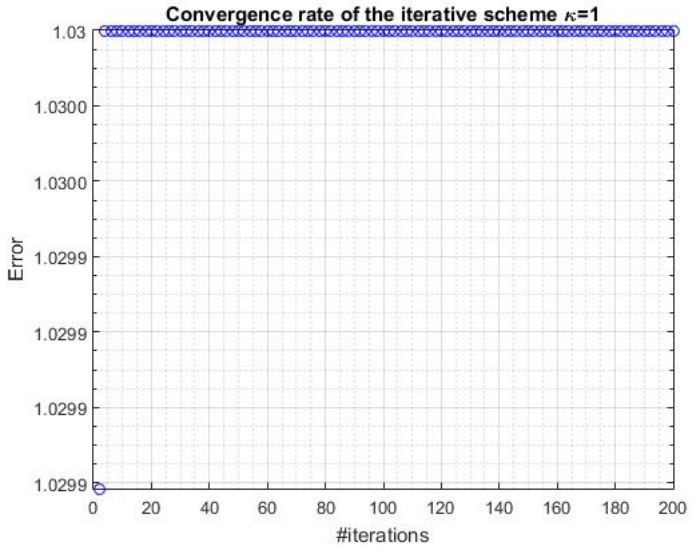

Figure 9: Convergence analysis of the iterative scheme.$\kappa = 0.01$

The fact that the algorithm doesn't converge with $\kappa = 0.01$ is due to the

fact that the Poincaré inequality that controls the stability of the method when imposing boundary is not satisfied. When we decrease κ the stability region is more restrictive til the point that this is broken and the method don't longer converges.

5. **Task 5**

In order to improve the not convergence for small values of the diffusion (κ) a relaxed scheme with constant relaxation parameter (w) is proposed. The idea of this algorithm is to relax the restriction of Poincaré inequality by means of decreasing the actualisation of system matrices at each time step.

To do so a parameter w is introduced in the solver of the iterative scheme in the following way:

```
%Left
    if HeatProblem.Data.FixFluxesLeft == 0
        HeatProblem.Matrices.K(1,1) = HeatProblem.
            Matrices.K(1,1) + w*HeatProblem.Data.
            kappa*1/h;
        HeatProblem.Matrices.K(1,2) = HeatProblem.
            Matrices.K(1,2) - w*HeatProblem.Data.
            kappa*1/h;
    else
        HeatProblem.Matrices.F(1) = HeatProblem.
            Matrices.F(1) + w*HeatProblem.Data.
            LeftFluxes;
    end

    %Right
    if HeatProblem.Data.FixFluxesRight == 0
        HeatProblem.Matrices.K(npoin,npoin) =
            HeatProblem.Matrices.K(npoin,npoin) + w*
            HeatProblem.Data.kappa*1/h;
        HeatProblem.Matrices.K(npoin,npoin-1) =
            HeatProblem.Matrices.K(npoin,npoin-1) - w
            *HeatProblem.Data.kappa*1/h;
    else
        HeatProblem.Matrices.F(end) = HeatProblem.
            Matrices.F(end) + w* HeatProblem.Data.
            RightFluxes;
    end
```

Choosing a small enough relaxation parameter the solution of the problem with $\kappa = 0.01$ in the left domain converges in few iterations (11) as seen in Figure 10.

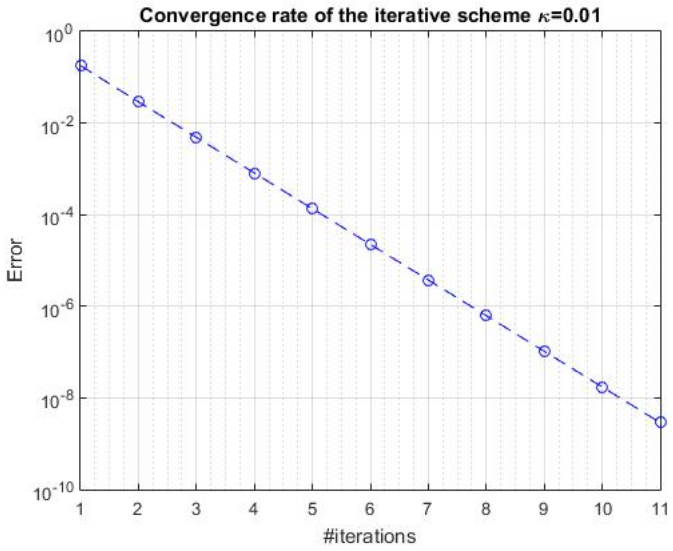

Figure 10: Convergence analysis of the iterative scheme with constant relaxation parameter $w = 0.005$,$\kappa = 0.01$

The temperatures distribution in this case is of the form of Figure 11. As seen the solution converges in the interface and the distribution of temperatures is characterised by the very small diffusivity on the left domain that induced the fast change in temperatures.

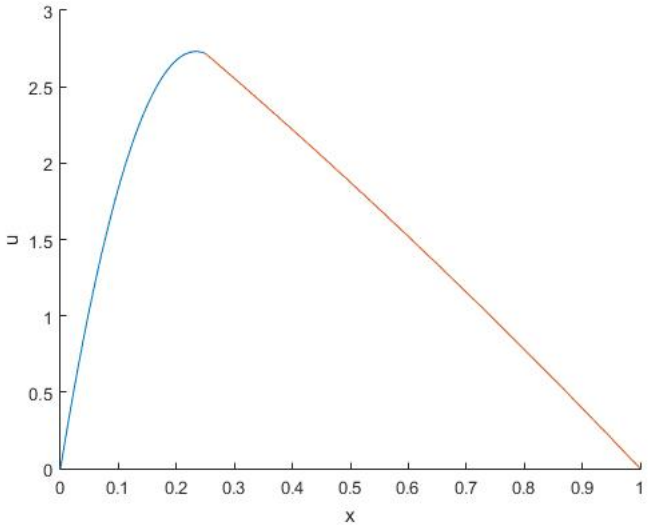

Figure 11: Temperature distribution of the solution with relaxed iterative approach: $w = 0.005$,$\kappa = 0.01$

Another possible solution to