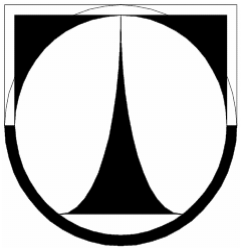
# 30. Setkání kateder mechaniky tekutin a termomechaniky


22.-24.6. 2011

Špindlerův Mlýn

***Jednotlivý příspěvek ze sborníku***



**TECHNICKÁ UNIVERZITA V LIBERCI**





INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

# Parallelization MATLAB for Solving Problems in Heat Conduction with Phase Change

Tomáš MAUDER[1]

[1] Ing, Tomáš Mauder, Brno University of Technology, Faculty Mechanical Engineering, Technická 2, Brno, Czech Republic, ymaude00@stud.fme.vutbr.cz

Abstract: *This paper deals with a suitable approach to parallelize the calculation of heat conduction with a phase change on multiple processors. The main reason of this is to speed up the solution time as much as possible. Various approaches how to solve these problems with the phase change can be used. In this paper we are dealing with the so-called enthalpy approach, which is based on the thermodynamic function of volume enthalpy. The heat conduction is numerically computed by the finite-difference method using the explicit Euler formula. The most time consuming operation through the calculation is to find the corresponding temperature from the actual enthalpy for each node at each time step. Thus, a convenient approach can be to split up the vector of enthalpy and parallelize the enthalpy-temperature search. Our origin numerical model was created in mathematical software MATLAB and the parallelization was processed on Parallel Computing Toolbox™ 5 in MATLAB.*

## 1. Introduction

In many engineering problems we can see it is necessary to encompass structural changes in the simulation for employed materials. These materials are typically characterized by their temperature-dependent thermo physical properties and latent heat, solid and liquid phase's consideration in calculations. The numerical calculation of transient heat transfer by the conduction with phase change materials requires a fine mesh and long execution times. Nevertheless, the potential of computation facility is nowadays very useful for parallel algorithms.

## 2. Phase changes materials

A phase change material (PCM) is a substance with a high heat of fusion which, melting and solidifying at a certain temperature, is capable of storing and releasing large amounts of energy. Heat is absorbed or released when the material changes from solid to liquid and vice versa; thus, PCMs are classified as latent heat storage (LHS) units. There are various approaches how to get over PCM problems, such as the apparent heat capacity method, the latent heat source approach, the enthalpy approach [1] etc. In this work the enthalpy approach was chosen. This method has a great advantage which is the accuracy of the solution, but this method suffers from fluctuations which are covered by a local refinement of the discretization near the solid-liquid interface. The major problem of this method is that in numerical terms, we are using an explicit parabolic scheme, and this is very penalizing. Nevertheless, for instance in the steel industry, where is casted a huge number of grades of steel, is very suitable to change the enthalpy-temperature (E-T) relationship for actual employed steel and to preserve the main model [2].

Heat conduction with PCM can be described by Fourier equation in the enthalpy form (eq. 1) [4]

$$\frac{\partial H}{\partial \tau} = \frac{\partial}{\partial x}\left(\lambda \frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(\lambda \frac{\partial T}{\partial y}\right) + \frac{\partial}{\partial z}\left(\lambda \frac{\partial T}{\partial z}\right), \quad (1)$$

where $\lambda$ is thermal conductivity [W/m·K], $T$ is temperature [K], $H$ is volume enthalpy [J/m$^3$], $\tau$ is time [s] and $x$, $y$, $z$ are spatial coordinates. In order to have a well-posed problem, initial and boundary conditions must be provided.

We were created our original model, where the equation (1) is discretized by the finite

difference method using an explicit formula for the time derivative [3], [4]. The temperature is calculated from the enthalpy by the searching algorithm which is described in Section 3.

The relationship between the enthalpy and the temperature can be seen for three chosen steel materials in Figure 1.

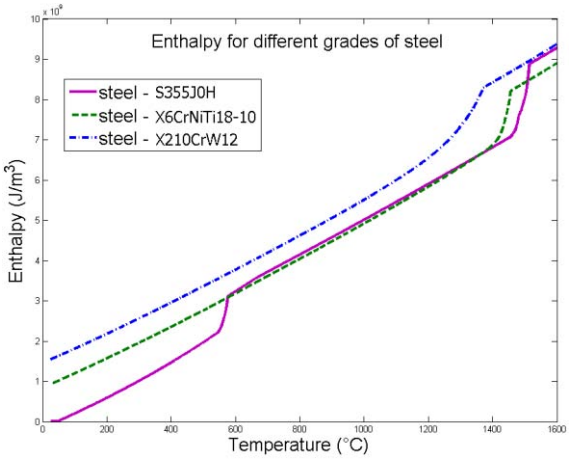



Figure 1. Relationship between temperature and enthalpy for three grades of steel

## 3. Enthalpy-temperature search

In this basic enthalpy scheme, the enthalpy is used as the primary variable and the temperature is calculated from a defined E-T relation. One method how to do this calculation can be the fitting of the E-T relationship by a curve, thus the temperature is described as a function of enthalpy. Despite the fact that this method is convenient for time execution, the poor numerical accuracy, mainly close to neighbor solidus and liquidus temperatures, makes this method inappropriate. Another way how to calculate the temperature from the actual enthalpy is to use a search technique [5]. The numerical accuracy increases but the solution time grows dramatically. In one-dimensional heat transfer the execution time is not a problem, but in heat transfer in three-dimensional space the time required by the E-T calculation is significant comparison to the rest of calculation. For instance in a mesh with 20x40x1200 nodes, we have to search for almost one million values in each time step. This is a good reason for the parallel calculation.

In our model the binary search algorithm is used. We have a table of temperatures ($T_1$,…,$T_n$) and corresponding enthalpies ($H_1$,…,$H_n$). Thus, we are looking for the position of known enthalpy which directly represents the corresponding temperature. The algorithm is shown in Figure 2. The symbol $H$ represents the actual enthalpy corresponding to the temperature we looking for. When searching is finished the desired temperature $T = i$.

1. [Initialize.] Set $l \leftarrow 1$, $u \leftarrow n$.

2. [Get midpoit.] if u<l algorithm terminates unsuccessfully. Otherwise, set $i \leftarrow [(l+u)/2]$.

3. [Compare.] if $H < H_i$, go to 4; if $H > H_i$ go to 5; and if $H = H_i$, the algorithm terminates successfully.

4. [Adjust u.] Set $u \leftarrow i-1$ and return to 2.

5. [Adjust l.] Set $l \leftarrow i+1$ and return to 2.

Figure 2. Search algorithm 1

One can say that this algorithm can be replaced by some more intelligent searching method. But for example in the temperature interval 1-1600 °C the temperature is found roughly in eight iterations. Moreover, in order to speed this algorithm, the user can use the knowledge from the previous time iteration and set the upper ($u$) and the lower ($l$) limits closer to the desired value. Although it can speed up the algorithm, the user has to check whether the desired value still in search range. Thus the total time savings are not so significant in comparison to the number of searching through the simulation.

## 4. Parallelization in MATLAB

Our numerical model was created in mathematical environment MATLAB, which is the most widely used tool in scientific and technical computing [7]. MATLAB gained popularity because of its user-friendliness,

wide internet support and plenty of mathematical toolboxes [6], [7].

MATLAB Parallel Computing Toolbox™ 5 is potentially useful for parallel programming but in the case MATLAB license does not contain this toolbox, there are many parallelization approaches supported by MATLAB like NetSolve, MultiMATLAB, pMATLAB, MATLAB*P [6], [7] etc. MATLAB Parallel Computing Toolbox™ 5 [8] allows to compute the problem in local PC on eight processors at the same time. The computation might not get entirely the eight-times improvement in speed because of a communications overhead, but the speed up should be significant.

There are more arguments why we should use parallel programming in MATLAB. One can be that some scientific problems simply do not fit into the memory of a single machine and has to be split among more CPUs. The multiple calculations are also proper on segments of code which are repetitive. For-loops to solve these cases are often used.

This is also the case of E-T search when the loop goes over the nodes from 1 to *N*, where *N* is equal to the number of nodes. Moreover, this loop is repeated in each time iteration. A number of these iterations can be huge in case of explicit formula due to the stability criteria.

The processors are separate as one MATLAB client and MATLAB workers (Figure 3).

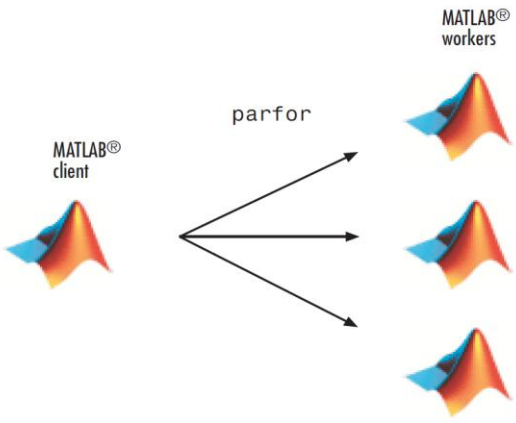



Figure 3. MATLAB parallelization schema

There are different ways how to provide parallel computing. In further text two approaches will be presented. In both cases we have to open a MATLAB pool by command:

```
matlabpool local open (n)
```

where *n* is number of processors. No index means to use all available processors.
In the first case, we modify a loop where the temperatures are searched. In MATLAB the common for-loop command is replaced by the parfor-loop command *parfor*. A part of the parfor body is executed on the MATLAB client and another part is executed in parallel on MATLAB workers. The parfor-loop is useful in situations where many loop iterations are needed.

```
parfor i = 1:N
    ....
      algorithm 1
    ....
end
```

The body of parfor-loop can contain other for-loops but the code in the MATLAB has to be adapted for parallel computing. More information can be found in [8].

The second case is based on dividing the vector containing enthalpy values in all nodes by the number of processors. Every sub-vector will be send to a specific processor and calculated here. The so-called *spmd* statement allows you to define a block of code to run it simultaneously on multiple workers [8].

```
for i = 1:n
    spmd(i)
        for i = 1:N/n
        ....
            algorithm 1
        ....
        end
    end
end
```

In both cases when we have finished with a MATLAB pool, close it with the command:

```
matlabpool close
```

which releases your pool of workers.

## 5. Results and discussion

We ran the simulations on Intel(R) Core(TM)2 CPU 6600 @ 2.40GHz with 4 GB RAM. The first simulation used one processor due the second used two processors. The simulations contained 3-dimensional heat transfer by conduction with PCM on mesh size 10x20x180 with 4000 time steps. In the one processor case the simulation time took 33.756 minutes and in the case with two processors 10.034 minutes. The realization of parallelization gave us 70.27% time saves in comparison to the non-parallel approach. With more processors the result could be even better. To compare our parallelization approaches, in the first case with the *parfor* command the calculation time was 5 minutes shorter than the case with the *spmd* command.

The parallelization in Parallel Computing Toolbox™ allows the user to easy parallelize MATLAB applications without CUDA or MPI programming on multicore processors, GPUs, and computer clusters. Another strong advantage is that existing MATLAB codes can be easily modified to a parallel structure. These reasons together with a professional support and the user's guide constitute the very convenient and strong tool for solving complex scientific problems. For users who prefer Octave, a freely available MATLAB-like scientific computing software, is available e.g. MultiOctave, ParallelOctave, [7] etc.

## 6. Conclusions

This paper was focused on parallel computations in complex engineering problems, such as the numerical simulation of heat conduction with the phase change. We tuned our original MATLAB code, for 3D heat transfer by the conduction with the search E-T algorithm, for parallel computing in order to decrease the simulation time. Two approaches were presented. The paper shows the executive time improvement and advantages of MATLAB approach; the easy implementation, the large users base, the internet support, MATLAB-friendliness, etc. As a drawback can be comprehend the MATLAB license price with Parallel Computing Toolbox™ 5, but one can easily find parallel modes for Octave.

## Acknowledgement

The author gratefully acknowledge a financial support from the project GA106/08/0606, GA106/09/0940, GAČR P107/11/1566 founded by the Czech Science, Foundation, project ED0002/01/01 - NETME Centre, Junior research project on BTU BD13102003 and by the project OC10051 of Czech Ministry of Education.

## 7. References

[1] MUHIEDDINE M., CANOT É., MARCH R.: VARIOUS APPROACHES FOR SOLVING PROBLEMS IN HEAT CONDUCTION WITH PHASE CHANGE, International Journal on Finite Volumes, 09. Volume 6 N°1 (2009), ISSN 1634-0655

[2] KAVIČKA F., ŠTĚTINA J.: THE INFLUENCE OF THERMOPHYSICAL PROPERTIES OF STEELS ON THE NUMERICAL SIMULATION OF CONCASTING PROCESS, Cambridge UK, WIT Press Boston. p. 243 - 252. (2000), ISBN 1-85312-824-4.

[3] LEE H. J., SCHIESSER W. E.: ORDINARY AND PARTIAL DIFFERENTIAL EQUATION ROUTINES IN C, C++, FORTRAN, JAVA®, MAMPLE®, AND MATLAB®, Boca Raton London New York Washington, D.C., A CRC Press Company, (2004), 1-58488-423-1

[4] TAVAKOLI R., DAVAMI P.: UNCONDITIONALLY STABLE FULLY EXPLICIT FINITE DIFFERENCE SOLUTION OF SOLIDIFICATION PROBLEMS, METALLURGICAL AND MATERIALS TRANSACTIONS, The Minerals, Metals & Materials Society and ASM International, (2007)

[5] KNUTH D. E.: THE ART OF COMPUTER PROGRAMMING, SECOND EDITION, United States of America and Canada, Addison Wesley Longman, (1998), ISBN 0-201-89685-0

[6] TREFETHEN A. E., et al.: MULTIMATLAB: MATLAB ON MULTIPLE PROCESSORS, Cornell University, (2008)

[7] CHOY R., EDELMAN A.: PARALLEL MATLAB: DOING IT RIGHT, Computer Science AI Laboratory, Proceedings of the IEEE, VOL. 93, NO. 2, Massachusetts Institute of Technology, Cambridge, MA 02139, (2005), 0018-9219

[8] USER'S GUIDE: PARALLEL COMPUTING TOOLBOX™ 5, The MathWorks, Inc. (2004–2011)