Journal of Universal Computer Science, vol. 26, no. 9 (2020), 1177-1188
submitted: 22/2/20, accepted: 6/7/20, appeared: 28/9/20

Numerical Treatment of a Data Completion Problem in Heat Conduction Modelling

Augusto C. de Castro Barbosa
(UERJ - Rio de Janeiro State University, Brazil
accb@ime.uerj.br)

Carlos A. de Moura
(UERJ - Rio de Janeiro State University, Brazil
demoura@ime.uerj.br)

Jhoab P. de Negreiros
(UNIGRANRIO - Great Rio University, Brazil
jhoabnegreiros@unigranrio.edu.br)

J. Mesquita de Souza Aguiar
(UERJ - Rio de Janeiro State University, Brazil
jaguiar@ime.uerj.br)

Abstract: This work deals with a question in the mathematical modelling for the temperature evolution in a bar, for a long time linked as an inverse problem. The one-dimensional model is the parabolic partial differential equation $u_t = \alpha u_{xx}$, known as the heat diffusion equation. The classic direct problem (DP) involves this equation coupled to a set of constraints: initial and boundary conditions, in such a way as to guarantee existence of a unique solution. The data completion (DC) problem hereby considered may be described as follows: the temperature at one of the bar extreme points is unknown but there is a fixed interior point where it may be measured, for all time. Finite difference algorithms (FDA) were tested to approximate the solution for such a problem.

The important point to be emphasized is that FDA may show up distinct performances when applied to either DP or DC, which is due to the way the discrete variables follow up the mesh steps – advancing in time, for the first case, on the space direction, for the other.

Key Words: Inverse Problem, Data Completion, Diffusion Equation, Heat Conduction, Finite Differences
Category: G.1.8, J.2, J.6

1 Introduction

Consider a homogeneous heat insulated one-dimensional bar. Representing by $u(x,t)$ the temperature at time t of its particle with coordinate x, the bar temperature evolution can be modelled, cf. [Cannon 1984], by the equation

$$u_t = \alpha u_{xx}\ , \quad \alpha \ \text{ a constant.}$$

The so-called **direct problem** associated to this model corresponds to determining a function – which describes the bar temperature state – that satisfies the above equation as well as fixed constraints, namely: its behaviour at the initial time and at the bar extreme points are assigned – it should be said, previously known. Take 0 and 1 as the bar extreme points coordinates. It is a basic fact from differential equations that, as long as the initial condition $u(x, 0) = \phi(x)$ and, for $i = 0$ and 1, the boundary conditions, *i.e.* either

$$(\mathbf{a.})\ u(i,t) = \alpha_i(t)$$

or

$$(\mathbf{b.})\ u_x(i,t) = \beta_i(t)$$

fulfill convenient hypotheses, one can guarantee the existence of a unique solution to this problem[1] [Friedman 1964].

Many problems may be considered for this model equation that fall into the class of **inverse problems**, cf. [Kirsch 1996], [Moura-Neto 2014], [Alifanov 1994] and [Wikströml 2006], like the one to be studied in the sequel, namely: knowledge of the state at the right end of the bar will be lacking, but the temperature value at an interior point x_0 will somehow be known, cf. [Shidfar 2007]. The solution will be proven to be unique and we show that it can be retrieved from the knowledge of this intermediate point temperature value, besides the temperature values at the left of x_0, and for all time instants under consideration. This study was previously known as an inverse problem but nowadays it is coined as a **data completion** – DC – problem.

In order to numerically calculate the temperature value at the right of x_0, we will apply two finite difference algorithms: Crank-Nicolson and *leap-frog*, both modified in their implementations, following the results in [Negreiros 2010]. The *leap-frog* scheme performance for this problem brings a surprise, when compared with the way it behaves for the DP. This somehow confirms Peter Lax saying in [Lax 2013]: "[We] claim that the theory of difference schemes is much more sophisticated than the theory of differential equations".

It is worth quoting some additional approaches to deal with other inverse problems for this model, as [Cannon 1998], [Frankel 1996] and [Pasquetti 1995].

This paper is organized as follows. Next section is devoted to the formulation of the data completion problem, plus the proof of its solution uniqueness; in the third section, the way the finite difference procedures implementation were carried out is explained; the following section mentions a conditionally stable numerical method for the DC-problem, and exhibits a simpler proof for its claimed behaviour; numerical tests are described in the fifth section; to close the paper, a discussion is presented and final conclusions are outlined.

[1] We could also require (a.) at right and (b.) at left or vice-versa.

2 Mathematical model

This section considers the problem of heat diffusion in a bar of length L:

$$u_t = \alpha u_{xx}, \quad 0 \leq x \leq L, \quad t \geq 0 \,, \tag{1a}$$

$$u(x,0) := f(x), \quad 0 \leq x \leq L \,, \tag{1b}$$

$$u(0,t) := l(t), \quad t \geq 0 \,, \tag{1c}$$

$$u(x_0,t) := i(t), \quad t \geq 0 \,, \tag{1d}$$

where α is a constant related to the material characteristics and $0 < x_0 < L$. The functions $f(x)$, $l(t)$ and $i(t)$ are known and $u(x,t)$ in (1a) is the function to be determined.

The approach to the DC-problem is carried out in two steps. In the first one, the bar, being considered to occupy the interval $[0, x_0]$, generates an initial and boundary value formulation – a direct problem – in the corresponding slab $[0, x_0] \times [0, T]$, with time up to $t = T$.

The second step considers the bar full length, *i.e.*, $[0, L]$, which leads to the non-classical problem of requiring the determination of a function u which satisfies (1a-1c), plus a corresponding constraint (1d), now for L replaced by x_0. We emphasize, that essentially it amounts to determine the solution values at the right-end border $x = L$. As mentioned above, for a long time this was refereed as an inverse problem. Here, we often will keep this terminology.

Theorem 1. The solution $u(x,t)$ of the DC-problem described by conditions (1a)-(1d) is unique in the domain $\Omega_L := \{(x,t) |\, 0 \leq x \leq L,\ t \geq 0\}$, no matter the value for x_0, a previously fixed point inside the size L bar.

Proof. Existence and uniqueness holds, as regards to equations (1a)-(1d), when considering the slab

$$\Omega_{x_0} := \{(x,t) \mid 0 \leq x \leq x_0,\ t \geq 0\} \,.$$

As mild assumptions on data guarantee analiticity, on the interior of the slab, for the solution of such mixed value problem, the principle of analytic extension implies uniqueness for the DC-problem.

3 Numerical Procedures

The finite differences method consists of first replacing the domain of the unknown function by a discrete one [Burden 2011]. Thus consider the heat diffusion equation (1a) in the domain $\Omega_{LT} := \{(x,t) |\ 0 < x < L, 0 < t < T\}$, L and T positive constants. The computational mesh is given by

$$x_i := i\Delta x = ih, \quad i = 0, 1, 2, \ldots, M, \quad L = M\Delta x \,,$$

$$t_j := j\Delta t = jk, \quad j = 0, 1, 2, \ldots, N, \quad T = N\Delta t\ ,$$

where M an N are integer numbers. In the already discretized domain, equation (1a) is then written, throughout the generated nodes, according chosen finite difference approximations to the relation

$$\frac{\partial u(x_i, t_j)}{\partial t} = \alpha \frac{\partial^2 u(x_i, t_j)}{\partial x^2}. \tag{2}$$

3.1 Regressive differences

Regressive differences is the first of two algorithms we have applied to DC. It turns out to be an unconditionally stable implicit method with local truncation error of order $\mathrm{O}(h^2+k)$. For DP, *i.e.*, $0 \le x_i \le x_0$ and $0 \le t^j \le M$, the scheme description follows. Denote by U_i^j the approximation to be found for $u(x_i, t_j)$, by means of the algebraic relation that simulates the differential equation, coupled to the values assigned to U_i^j in correspondence to initial and boundary values:

$$\frac{U_i^{j+1} - U_i^j}{k} = \alpha \frac{U_{i+1}^{j+1} - 2U_i^{j+1} + U_{i-1}^{j+1}}{h^2}\ , \tag{3a}$$

$$u(x, 0) := f(x),\ 0 \le x \le x_0\ , \tag{3b}$$

$$u(0, t) := l(t),\ 0 \le t \le T\ , \tag{3c}$$

$$u(x_0, t) := i(t),\ 0 \le t \le T. \tag{3d}$$

On the inverse problem the same algorithm of the direct problem is applied in $x_0 \le x_i \le M$ and $0 \le t^j \le N$,

$$\frac{U_i^{j+1} - U_i^j}{k} = \alpha \frac{U_{i+1}^{j+1} - 2U_i^{j+1} + U_{i-1}^{j+1}}{h^2}\ , \tag{4a}$$

$$u(x, 0) := f(x),\ x_0 \le x \le M\ , \tag{4b}$$

$$u(x_0, t) := i(t),\ 0 \le t \le T\ . \tag{4c}$$

However, due to the lack of information on the right border an adaptation must be carried out.

The diffusion equation is simulated by (4a) on x_0, in $j = 1$ level,

$$\frac{U_{x_0}^1 - U_{x_0}^0}{k} = \alpha \frac{U_{x_0+1}^1 - 2U_{x_0}^1 + U_{x_0-1}^1}{h^2}\ . \tag{5}$$

The nodes $U_{x_0}^1$ and $U_{x_0}^0$ are given by $i(t)$ and $U_{x_0-1}^1$ has been calculated in the direct problem. Consequently, we obtain an approximation for the $U_{x_0+1}^1$ node, which is an inverse problem node. Repeating this procedure until reaching the node U_{M-1}^1, we obtain the vector

$$U^1 = (U_{x_0+1}^1, U_{x_0+2}^1, \ldots, U_{M-1}^1, U_M^1).$$

Advancing up to level $j = N$ and repeating the same procedure above, we obtain approximations for all nodes in the inverse problem.

Therefore, when the finite difference algorithm is applied to the inverse problem, it is no longer an implicit scheme as it has occurred on the direct problem, but it is now an explicit scheme. Being $r := h^2/(\alpha k)$ in (3a), this follows from

$$U_{i+1}^{j+1} = r(U_i^{j+1} - U_i^j) - U_{i-1}^{j+1} + 2U_i^{j+1} . \qquad (6)$$

Theorem 2. *The method of regressive differences is* **unconditionally unstable** *when applied to the current inverse problem.*

Proof. Expression (6) is equivalent to

$$U_{i+1}^{j+1} = (r+2)\, U_i^{j+1} - rU_i^j - U_{i-1}^{j+1},$$

which can be written, in matrix form, as

$$\begin{pmatrix} U_{i+1}^1 \\ U_{i+1}^2 \\ U_{i+1}^3 \\ \vdots \\ U_{i+1}^T \end{pmatrix} = \begin{pmatrix} r+2 & 0 & 0 & 0 & \cdots & 0 & 0 \\ -r & r+2 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & -r & r+2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & -r & r+2 \end{pmatrix}$$

$$\times \begin{pmatrix} U_i^1 \\ U_i^2 \\ U_i^3 \\ \vdots \\ U_i^T \end{pmatrix} - r \begin{pmatrix} U_i^0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} - \begin{pmatrix} U_{i-1}^1 \\ U_{i-1}^2 \\ U_{i-1}^3 \\ \vdots \\ U_{i-1}^T \end{pmatrix}$$

where $U_{i+1} = AU_i + V$, with $A = (r+2)I + N$, being N a nilpotent matrix:

$$A = \begin{pmatrix} r+2 & 0 & 0 & \cdots & 0 & 0 \\ -r & r+2 & 0 & \cdots & 0 & 0 \\ 0 & -r & r+2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -r & r+2 \end{pmatrix}$$

$$= \begin{pmatrix} r+2 & 0 & 0 & 0 \cdots & 0 \\ 0 & r+2 & 0 & 0 \cdots & 0 \\ 0 & 0 & r+2 & 0 \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \; \vdots & \vdots \\ 0 & 0 & 0 & 0 \cdots & r+2 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 \cdots & 0 \\ -r & 0 & 0 & 0 \cdots & 0 \\ 0 & -r & 0 & 0 \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \; \vdots & \vdots \\ 0 & 0 & 0 & 0 \; -r & 0 \end{pmatrix} .$$

Denoting by v any eigenvector of A, we have

$$Nv = \lambda v - (r+2)v = (\lambda - (r+2))\, v.$$

As N is nilpotent, there is a natural k such that $N^k = 0$. Consequently

$$N^k v = (\lambda - (r+2))^k\, v = 0 \;\Rightarrow\; (\lambda - (r+2))^k = 0 \;\Rightarrow\; \lambda = r+2.$$

Thus, the only eigenvalue of A is $r+2$, with multiplicity n. As r is positive, all eigenvalues of A have absolute value greater than 1, so the method is unstable.

3.2 Crank-Nicolson and *leap-frog*

We have taken the choice to employ the Crank-Nicolson algorithm for the direct problem, the *leap-frog* scheme to the inverse problem, cf. [Forsythe 1960]. The relevant distinction between these two methods as regards to the regressive differences is the local truncation error: both present order $O(k^2 + h^2)$, thus exhibiting an efficiency greater than the regressive differences method, which is limited to $O(k + h^2)$. It should be mentioned the option to reach for this equation a fully quadratic truncation error, even with a two-level scheme, see [de Moura 1993].

The Crank-Nicolson algorithm coupled to progressive differences is performed by replacing (3a) by the following scheme:

$$\frac{U_i^{j+1} - U_i^j}{k} = \frac{\alpha}{2}\left(\frac{U_{i+1}^j - 2U_i^j + U_{i-1}^j}{h^2} + \frac{U_{i+1}^{j+1} - 2U_i^{j+1} + U_{i-1}^{j+1}}{h^2}\right) . \tag{7}$$

The algorithm with local truncation error of order $O(k^2+h^2)$ which generated the best results when applied to the inverse problem was the *leap-frog*. However, it may be remarked that its application differs from the way the other ones are treated, due to the advance of the variables now in space rather than in time. The structure of the inverse problem approach is done by replacing (4a) by the scheme

$$\frac{U_i^{j+1} - U_i^{j-1}}{2k} = \alpha \frac{U_{i+1}^j - 2U_i^j + U_{i-1}^j}{h^2} . \tag{8}$$

The *leap-frog* algorithm, which proved to be very precise and consistent in several computational tests, is applied to the inverse problem as follows:

$$U_{i+1}^j = s(U_i^{j+1} - U_i^{j-1}) + 2U_i^j - U_{i-1}^j , \tag{9}$$

with now $s := h^2/(2\alpha k)$.

The equation (9) shows that the $(i+1)$-th step in space needs the values of the i-th and $(i-1)$-th steps in space, as illustrated by Figure 1.

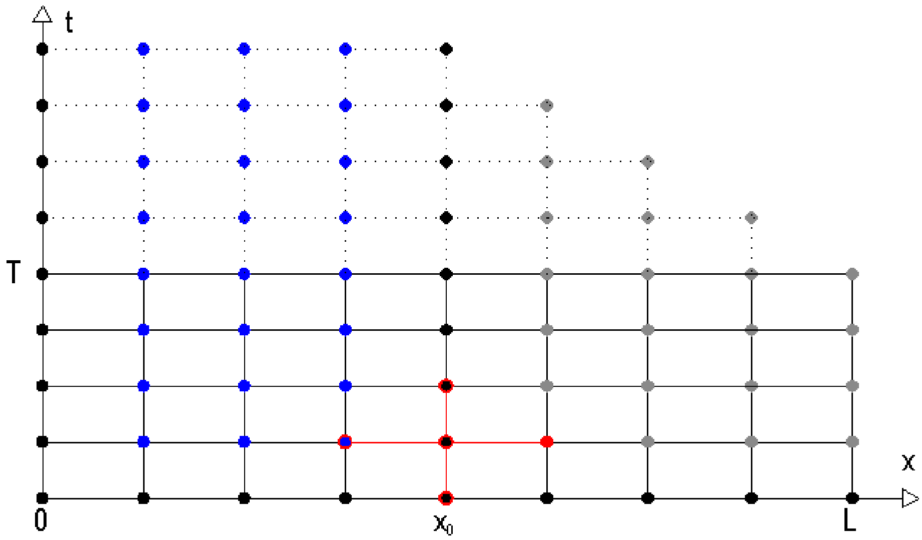

Figure 1: Mesh employed for Crank-Nicolson and *leap-frog* algorithms

The *Leap Frog* Method As described in (8), the template for the method amounts to

$$U_{i+1}^{j} = s(U_i^{j+1} - U_i^{j-1}) + 2U_i^j - U_{i-1}^j \ , \quad s = \frac{h^2}{2\alpha k}.$$

In matrix form we have

$$\begin{pmatrix} U_{i+1}^1 \\ U_{i+1}^2 \\ U_{i+1}^3 \\ \vdots \\ U_{i+1}^T \end{pmatrix} = \begin{pmatrix} 2 & s & 0 \cdots & 0 \\ -s & 2 & s \cdots & 0 \\ 0 & -s & 2 \cdots & 0 \\ \vdots & \vdots & \vdots \cdots & 0 \\ 0 & 0 & 0 \cdots & 2 \end{pmatrix} \begin{pmatrix} U_i^1 \\ U_i^2 \\ U_i^3 \\ \vdots \\ U_i^T \end{pmatrix} + \begin{pmatrix} -sU_i^0 \\ 0 \\ 0 \\ \vdots \\ sU_i^T \end{pmatrix} - \begin{pmatrix} U_{i-1}^1 \\ U_{i-1}^2 \\ U_{i-1}^3 \\ \vdots \\ U_{i-1}^T \end{pmatrix},$$

but

$$\begin{pmatrix} 2 & s & 0 \cdots & 0 \\ -s & 2 & s \cdots & 0 \\ 0 & -s & 2 \cdots & 0 \\ \vdots & \vdots & \vdots \cdots & 0 \\ 0 & 0 & 0 \cdots & 2 \end{pmatrix} = \begin{pmatrix} 2 & 0 & 0 \cdots & 0 \\ 0 & 2 & 0 \cdots & 0 \\ 0 & 0 & 2 \cdots & 0 \\ \vdots & \vdots & \vdots \cdots & 0 \\ 0 & 0 & 0 \cdots & 2 \end{pmatrix} + \begin{pmatrix} 0 & s & 0 \cdots & 0 \\ -s & 0 & s \cdots & 0 \\ 0 & -s & 0 \cdots & 0 \\ \vdots & \vdots & \vdots \cdots & s \\ 0 & 0 & 0 \ -s & 0 \end{pmatrix}.$$

That is, $M = 2I + A$, where I is the identity and A is an anti-symmetric matrix. Since any vector of $\mathbb{C}^n$ is an eigenvector of $2I$, with eigenvalue 2, if $v \in \mathbb{C}^n$ is an eigenvector for A with eigenvalue $\lambda \in \mathbb{C}^n$, so v is an eigenvector for M whose eigenvalue is $2+\lambda$. Recall that the eigenvalues of a real antisymmetric matrix are all pure imaginary, and thus $|2 + \lambda| \geq 2$. Having M an eigenvalue with absolute value bigger than 1, the associated method is unstable.

Our *leap-frog* algorithm implementation has followed the steps described in the sequel.

- Step 1 - the additional interval $[T, T + \eta]$ where data will be needed in order to perform the inverse problem part is estimated;
- Step 2 - approximations up to the limit $T + \eta$ are obtained in the direct problem;
- Step 3 - solution approximations for the inverse problem are reached by calculating the vectors

$$\mathbf{U_{x_0+1}} = (U^1, U^2, U^3, \ldots, U^{T+\eta-1}) \ ,$$
$$\mathbf{U_{x_0+2}} = (U^1, U^2, U^3, \ldots, U^{T+\eta-2}) \ ,$$

up to

$$\mathbf{U_N} = (U^1, U^2, U^3, \ldots, U^T).$$

4 Numerical results

This section exhibits the numerical results generated with our chosen algorithm, namely (CN + LF). All examples (5-1)–(5-4) below consider the heat equation (1a) with $\alpha = 1, \ \ 0 \leq x \leq 1$ and $t \geq 0$, plus initial condition $u(x, 0) = \sin(x)$, boundary condition $u(0, t) = 0$ and at the interior point $u(x_0, t) = \sin(x_0)e^{-t}$. Knowledge of this problem solution, namely $u(x, t) = \sin(x)e^{-t}$, has been employed towards the error data computing.

Example 1.

These test data allow to verify the relation between the point x_0 location and the computing errors at the boundary point $x = 1$.

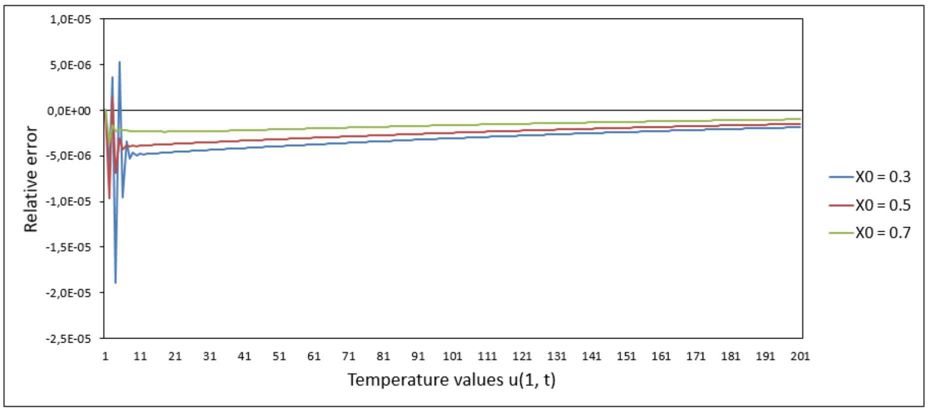

Figure 2: Temperature values $u(1,t)$ calculated with (CN + LF)

The values obtained for the solution at $x = 1$ correspond to $\Delta x = 0.05$ and $\Delta t = 0.02$, with x_0 at 0.3, 0.5 and 0.7.

It is clearly shown at the picture in Fig. 2 that the closer x_0 gets to the right boundary, the better the approximation values become.

Example 2.

The present test is intended to verify the relationship between the step length Δt and the resulting error at $u(x,1)$.

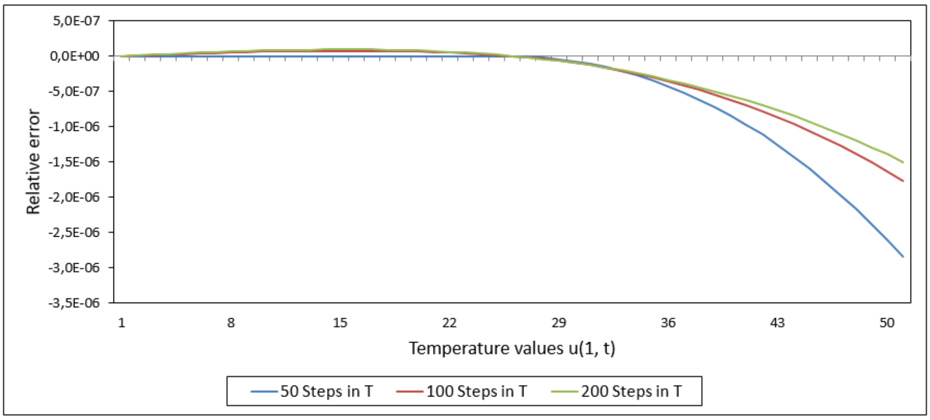

Figure 3: Temperature values $u(1,t)$ calculated by (CN + LF)

Approximations for solution at $x = 1$ were reached by taking $\Delta x = 0.02$ and $\Delta t = 0.02$ - 0.01 - 0.005, with $x_0 = 0.5$.

The graph at figure 3 confirms the method consistency, since as the step length Δt gets thinner, the corresponding error also gets smaller.

Example 3.

The aim for this test was to verify the relation between the location of the point x_0 with the error at the boundary point $x = 1$.

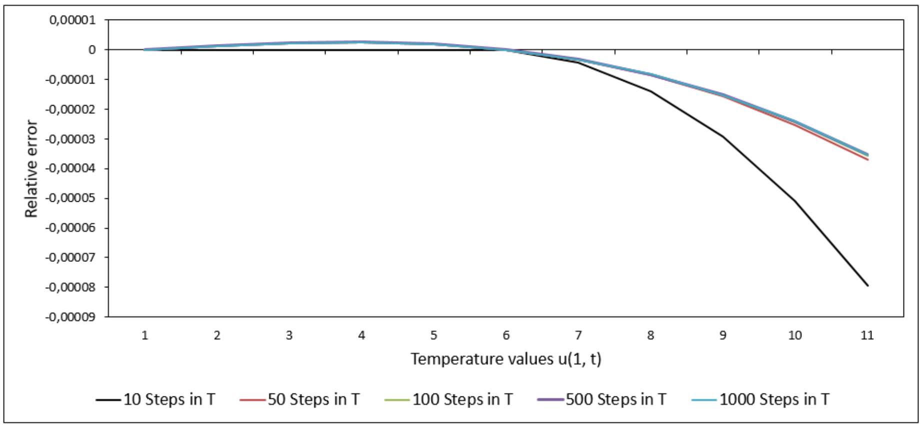

Figure 4: Values for the temperature $u(1, t)$ calculated with (CN + LF)

The solution was obtained at $x = 1$ with 10 steps for the variable x and the value $x_0 = 0.5$.

Example 4.

The present test purpose was to measure the relationship between the step length Δx with the approximations at $x = 1$.

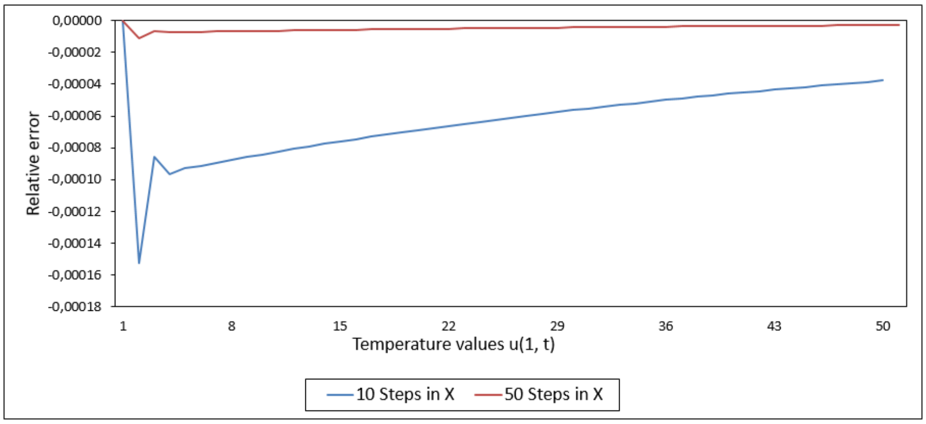

Figure 5: Values for the temperature $u(1, t)$ calculated with (CN + LF)

Approximations for the solution at $x = 1$ have been obtained with $\Delta t = 0.02$ and $x_0 = 0.5$, with $\Delta x = 0.1$ and 0.02.

5 Conclusions

In the present work we have applied a finite difference algorithm for the so called data completion problem. We have been restricted to a quite simple framework, namely, the homogeneous one-dimensional heat equation with a constant diffusion coeficient. The employed algorithm (CN + LF) has always shown its consistency.

As should be expected, in the direct problem Crank-Nicolson classical method has performed quite well. The special surprise was *leap-frog* good performance: it is universally known that for the diffusion equation direct problem this is an unconditionally unstable method. The good behavior we have met is certainly due to the way the discrete variables pave their advance; here the space variable holds the control, not the time variable, as in the direct problem. This surely is a seminal point, despite not yet fully absorbed: numerical simulation of a partial differential equation may turn out strongly dependent of the direction the discretized variables advance.

Besides, from the performed tests we observe the obvious importance of the placing of the point x_0: the closer to the right boundary it lies, the better results are generated. Of course this is a constraint not easily fulfilled in practical situations.

The ideas hereby described and tested certainly demand their employment in more demanding models. Among them it is worth mentioning two-dimensional bodies, external heat sources as well as less restrictive equations, like the ones considered in [Shidfar 2007].

Acknowledgement

One of the authors (CAdeM) was partially supported by a grant from FAPESP – the Carlos Chagas Filho Rio de Janeiro State Foundation for Science Research Support –, namely ref. E-26/202.500/2019.

References

[Alifanov 1994] O. M. Alifanov, Inverse Heat Conduction Problem. Springer-Verlag, New York, 1994.

[Burden 2011] R. L. Burden, J. D. Fairies, Numerical Analysis, 9^{th} ed. Cengage Learning, Boston, 2011.

[Cannon 1984] J. R. Cannon, The One-Dimensional Heat Equation. Addison-Wesley, Boston, 1984.

[Cannon 1998] J.R. Cannon, P. Duchateau, Structural identification of a term in a heat equation, Inverse Problem 14, 535-551 (1998).

[de Moura 1993] C. A. de Moura, An explicit, really quadratic 2-level scheme for the diffusion equation, Journal of Computing and Information – Toronto, Canada –, 3 (1) 99-115 (1993).

[Forsythe 1960] G. E. Forsythe, W. R. Wasow, Finite Difference Methods for Partial Differential Equations. J. Wiley, New York, 1960.

[Frankel 1996] J. I. Frankel, Residual-minimization least-squares method for inverse heat conduction, Comp. Math. Appl. 332 (4), 117-130 (1996).

[Friedman 1964] A. Friedman, Partial Differential Equations of Parabolic Type. Prentice-Hall Inc., Upper Saddle River, New Jersey, 1964.

[Kirsch 1996] A. Kirsch, An Introduction to the Mathematical Theory of Inverse Problems, Applied Mathematical Sciences. Springer-Verlag, New York, 1996.

[Kouashi 2006] S. Kouashi, Eigenvalues and Eigenvectors of Tridiagonal Matrices. Electronic Journal of Linear Algebra, 15, 115-133 (2006).

[Kouashi 2008] S. Kouashi, Eigenvalues and Eigenvectors of some tridiagonal matrices with non-constant diagonal entries. Applicationes Mathematicae, 35 (1), 107-120 (2008).

[Lax 2013] P. D. Lax, Stability of Difference Schemes, pp.5-14, *in* C. A. de Moura, C. S. Kubrusly (Eds.), The Courant-Friedrichs-Lewy (CFL) Condition. Birkhäuser, New York, 2013.

[Moura-Neto 2014] F. D. Moura Neto, A. J. Silva Neto, An Introduction to Inverse Problems with Applications. Springer-Verlag, Berlin-Heidelberg, 2014.

[Negreiros 2010] J. P. Negreiros, An Inverse Problem for Heat Difusion Modelling. *In Portuguese*: Um Problema Inverso na Modelagem da Difusão do Calor. Dissertation for M.Sc. in Computational Sciences, Rio de Janeiro State University – UERJ, Rio de Janeiro, 2010.

[Pasquetti 1995] R. Pasquetti, D. Petit, Inverse Diffusion by Boundary Elements, Engineering Analysis with Boundary Elements 15, 197-205 (1995).

[Shidfar 2007] A. Shidfar, J. Damirchi, P. Reihani, An stable numerical algorithm for identifying the solution of an inverse problem, Appl. Math. Comput., 190, 231-236 (2007).

[Wikströml 2006] P. Wikström, A study of surface temperature and heat flux estimations by solving an Inverse Heat Conduction Problem. Royal Institute of Technology, School of Industrial Engineering and Management, Stockhölm, 2006.