MultIHeaTS : an Open Source Implicit Thermal Solver for 1D Multilayered Surfaces

C. Mergny[1*] and F. Schmidt[1,2]

[1*]GEOPS, Paris-Saclay University, Orsay, France.
[2] Institut Universitaire de France, paris, France.

*Corresponding author(s). E-mail(s): cyril.mergny@universite-paris-saclay.fr;
Contributing authors: frederic.schmidt@universite-paris-saclay.fr;

Abstract

An implicit Euler scheme is proposed for solving heat conduction problems in 1D heterogeneous media, available as an open source Python code. The algorithm uses finite difference on an irregular grid and is unconditionally stable due to the implicit formulation. The boundary conditions can be imposed either in temperatures or fluxes. The thermal solver is validated against a stiff analytical solution, demonstrating its robustness in handling stiff equations. Its general applicability for heterogeneous cases is demonstrated through its use in a planetary surface scenario with non-linear boundary conditions induced by black body thermal emission. The upper boundary condition is formulated explicitly, which introduces negligible errors while maintaining the scheme's stability. Although the Crank-Nicolson scheme is generally considered more reliable for homogeneous media, our numerical study shows that the proposed implicit scheme is more accurate for larger time steps in more realistic heterogeneous media, which is particularly relevant for the planetary science field.

Keywords: Heat Diffusion Equation; Implicit Finite Difference; Heterogeneous Surfaces; Non-Linear Boundary Condition; Open-Source Python

1 Introduction

The heat equation is a fundamental partial differential equation that governs the behavior of heat transfer in various physical systems. However, when dealing with heterogeneous media, where the thermal properties of the material vary spatially, the heat equation becomes too complex to solve analytically [1, 2]. In one dimension, the heat equation for conduction transfers can be expressed as:

$$\rho(x,t)c_{\mathrm{p}}(x,t)\frac{\partial T(x,t)}{\partial t} = \frac{\partial}{\partial x}\left(k(x,t)\frac{\partial}{\partial x}T(x,t)\right) + Q(x,t) \tag{1}$$

where x denotes space and t denotes time, T is the temperature, ρ is the density, c_{p} is heat capacity, k is thermal conductivity, and Q denotes an optional source or sink term. We propose to solve numerically the heat equation where all thermal properties can vary continuously over space and time, and with non-constant time and space increments.

Several numerical methods have been developed to solve the heat equation for heterogeneous media, including finite element and finite volume methods [2–6], finite difference schemes [7, 8], and other methods [9, 10]. In this paper, we focus on finite difference schemes due to their ease of implementation, making them more accessible for coupling with multi-physics simulations in further studies [11].

The target science case is in Earth and Planetary science with surface conditions subject to solar illumination. In this scenario, the upper boundary condition (i.e., the flux leaving the surface) is determined by the energy equilibrium between the solar flux and the black body emission from the surface [7].

$$\forall t, k(0,t)\left.\frac{\partial T(x,t)}{\partial x}\right|_{x=0} = -F_{\mathrm{solar}}(t) + \epsilon \cdot \sigma_{\mathrm{SB}} \cdot T(0,t)^4 \tag{2}$$

where ϵ is the thermal emissivity and σ_{SB} the Stefan-Boltzmann constant. A simplified model for solar illumination at the equator with zero obliquity throughout a sidereal day can be expressed as a truncated sinusoidal function [7] :

$$F_{\mathrm{solar}}(t) = \begin{cases} (1-A)\dfrac{G_{\mathrm{SC}}}{d^2}\cos\left(\dfrac{2\pi t}{P}\right) & \text{if } 2\pi t/P \text{ in } \left[-\frac{\pi}{2}, \frac{\pi}{2}\right] \\ \text{or} & \\ 0 & \text{if } 2\pi t/P \text{ in } \left]\frac{\pi}{2}, \frac{3\pi}{2}\right[\end{cases} \tag{3}$$

where A is the surface albedo, G_{SC} the solar constant in W.m^{-2} and d the distance between the Sun and the planet in AU. It should be noted that while this simplified model captures the essential features of solar illumination, more realistic estimations, for instance from NASA's SPICE toolkit [12, 13] can be easily integrated.

In the planetary science context, various numerical models have been developed to study the behavior of both homogeneous surfaces [14–16], and heterogeneous media, as shown in Spencer's implementation of the explicit Euler scheme [7]. Although the explicit method is easier to numerically implement and has a faster computational time per iteration, its main drawback is its conditional stability [17]. A Crank-Nicolson scheme [18] has been proposed by Schorgofer [8]. This scheme is more complex to implement and requires a computationally expensive matrix inversion. However, it offers the advantage of being unconditionally stable and more precise than first-order implicit and explicit schemes for homogeneous media, owing to its second-order truncation error in time [19].

Although the implicit Euler scheme is also unconditionally stable, to our knowledge, there are no code available online that uses it to solve the heat equation in a heterogeneous media. Therefore, our goal is to provide the scientific community with an open-source, easy-to-use, and versatile implicit solver called MultIHeaTS (Multi-layered Implicit Heat Transfer Solver) for solving the heat equation in such conditions. The source code is available online [1]. Even though our target science is planetary science with surface conditions subject to solar illumination [7, 8, 14–16], this approach is applicable in a wide range of boundary conditions for a large set of scientific and technical cases.

In the first section of this article, we begin by presenting the mathematical derivation of the heat equation in heterogeneous media using the implicit Euler method. We then proceed to validate the MultIHeaTS algorithm by comparing its results to analytical solutions and other existing algorithms. Finally, we conduct a numerical study to evaluate the performance of the finite difference schemes under realistic planetary science conditions.

2 Methods

To discretize the heat equation, we used finite differences on an irregular spatial grid consisting of n_x points, which we iterated for a total of n_t iterations. The spatial and temporal parameters were discretized as follows:

$$\begin{cases} x \to x_n & = x_{n-1} + \Delta x_n \\ t \to t^i & = t^{i-1} + \Delta t^i. \end{cases} \tag{4}$$

Here, n is an integer such that $n \in \{0, \dots, n_x - 1\}$, representing the nth element in the spatial dimension, and i is an integer such that $i \in \{0, \dots, n_t - 1\}$, representing the ith element in the time dimension. It is worth noting that both the spatial Δx_n and temporal Δt^i increments may not be constant.

2.1 Backward Euler Finite Differences on an Irregular Grid

Following the method described in [20], the first order derivative of function f using the central difference approximation can be written as :

$$\frac{\partial f}{\partial x}(x_n) = \frac{f_{n+1} - f_n}{2\Delta x_n} + \frac{f_n - f_{n-1}}{2\Delta x_{n-1}} + \mathcal{O}(\Delta x_n^2) \tag{5}$$

and the second order derivative as :

$$\frac{\partial f^2}{\partial x^2} = \frac{2\left(f_{n+1}\Delta x_{n-1} + f_{n-1}\Delta x_n - f_n\left(\Delta x_n + \Delta x_{n-1}\right)\right)}{\Delta x_n \Delta x_{n-1}\left(\Delta x_n + \Delta x_{n-1}\right)} + \mathcal{O}(\Delta x_n^3) \tag{6}$$

[1] MultIHeaTS open-source code available at https://gitlab.dsi.universite-paris-saclay.fr/cyril.mergny/multiheats

By using Equations 5 and 6, the heat equation can be discretized with second order accuracy in space and first order in time. After factorizing each temperature term, the resulting equation is given by:

$$\begin{aligned} T_n^i + r_n^i Q_n^i =& T_{n-1}^{i+1}\left[\frac{-r_n}{\Delta x_{n-1}}\left(-\frac{1}{2}\frac{\partial k}{\partial x} + \frac{2k_n}{\Delta x_n + \Delta x_{n-1}}\right)\right] \\ &+ T_n^{i+1}\left[1 - \frac{r_n}{\Delta x_n \Delta x_{n-1}}\left(\frac{(\Delta x_n - \Delta x_{n-1})}{2}\frac{\partial k}{\partial x} - 2k_n\right)\right] \\ &+ T_{n+1}^{i+1}\left[\frac{-r_n}{\Delta x_n}\left(\frac{1}{2}\frac{\partial k}{\partial x} + \frac{2k_n}{\Delta x_n + \Delta x_{n-1}}\right)\right] \end{aligned} \tag{7}$$

with r_n a coefficient expressed as $r_n^i = \Delta t^i/(\rho_n^i c_{\mathrm{p},n}^i)$. With matrix notation this is equivalent to the system :

$$\begin{pmatrix} b_0 & c_0 & & & \cdots & & 0 \\ a_1 & b_1 & c_1 & & & & \\ & \ddots & \ddots & \ddots & & & \\ \vdots & & a_n & b_n & c_n & & \vdots \\ & & & \ddots & \ddots & \ddots & \\ & & & & a_{n_x-2} & b_{n_x-2} & c_{n_x-2} \\ 0 & & \cdots & & & a_{n_x-1} & b_{n_x-1} \end{pmatrix} \cdot \begin{pmatrix} T_0 \\ \vdots \\ T_{n-1} \\ T_n \\ T_{n+1} \\ \vdots \\ T_{n_x-1} \end{pmatrix}^{i+1} = \begin{pmatrix} s_0 \\ \vdots \\ s_{n-1} \\ s_n \\ s_{n+1} \\ \vdots \\ s_{n_x-1} \end{pmatrix}^{i} \tag{8}$$

where the coefficients a_n, b_n, c_n and s_n are given by

$$\forall n \in \{2, \ldots, n_x - 1 \begin{cases} a_n = \dfrac{-r_n}{\Delta x_{n-1}}\left(-\dfrac{1}{2}\dfrac{\partial k}{\partial x} + \dfrac{2k_n}{\Delta x_n + \Delta x_{n-1}}\right) \\ b_n = 1 - \dfrac{r_n}{\Delta x_n \Delta x_{n-1}}\left(\dfrac{(\Delta x_n - \Delta x_{n-1})}{2}\dfrac{\partial k}{\partial x} - 2k_n\right) \\ c_n = \dfrac{-r_n}{\Delta x_n}\left(\tfrac{1}{2}\dfrac{\partial k}{\partial x} + \dfrac{2k_n}{\Delta x_n + \Delta x_{n-1}}\right) \\ s_n = T_n^i + r_n^i Q_n^i \end{cases} \tag{9}$$

and by the boundary conditions.

2.2 General Boundary Conditions

MultIHeaTS accepts any type of boundary conditions, including imposed flux or imposed temperatures at the boundaries. A detailed derivation of the Dirichlet boundary conditions can be found in Supplementary Materials Section 1. In the case of planetary surface evolution, we are specifically interested in Neumann boundary conditions, which involve imposed flux. In this scenario, the flux is prescribed at the boundaries:

$$\forall t, k_0 \left.\frac{\partial T(x,t)}{\partial x}\right|_{x_0} = \phi_0(t). \tag{10}$$

By injecting Equation 10 into the discretized heat Equation 7, the upper boundary condition becomes:

$$T_0^{i+1} = T_0^i + r_0^i \left[\Delta x_0 \phi_0^{i+1} \left(k_1^i - 3k_0^i\right) + \Delta x_0^2 Q_0^i\right] + 2r_0^i k_0^i T_1^{i+1} - 2r_0^i k_0^i T_0^{i+1} \tag{11}$$

which gives the expression of the first coefficients of the tri-diagonal matrix :

$$\begin{cases} b_0 = 1 + 2r_0 k_0 / \Delta x_0^2 \\ c_0 = -2r_0 k_0 / \Delta x_0^2 \\ s_0 = T_0^i + r_0 \left(\phi_0 \left(k_1 - 3k_0\right) / \Delta x_0 + Q_0\right) \end{cases} \tag{12}$$

The same reasoning can be applied to the bottom boundary condition :

$$\forall t, \left.\frac{\partial T(x,t)}{\partial x}\right|_{x_{n_x-1}} = \phi_{n_x-1}(t) \tag{13}$$

which gives the expression of the last tri-diagonal coefficients :

$$\begin{cases} a_{n_x-1}^i = -2k_{n_x-1}^i r_{n_x-1}^i \\ b_{n_x-1}^i = 1 + 2k_{n_x-1}^i r_{n_x-1}^i \\ s_{n_x-1}^i = T_{n_x-1}^i + r_{n_x-1}^i \left[\Delta x_{n_x-1} \phi_{n_x-1}^{i+1} \left(3k_{n_x-1}^i - k_{n_x-2}^i\right) + \Delta x_{n_x-1}^2 Q_{n_x-1}^i\right] \end{cases} \tag{14}$$

2.3 Non-Linear Boundary Conditions

When dealing with planetary-like surfaces, a non-linearity problem arises from the energy equilibrium described in Equation 2. This non-linearity comes from the Stefan-Boltzmann law, which states that the surface flux ϕ is proportional to T_0^4, making it a non-linear function of temperature:

$$\phi(t, T_0) \propto T_0^4 \tag{15}$$

For the explicit scheme, the non-linearity in the surface flux does not pose any issue as the surface flux can be pre-calculated using the previous surface temperature T_0^{i-1}. However, for Crank-Nicolson and Backward Euler schemes, the non-linearity of the Stefan-Boltzmann law makes it impossible to solve the implicit formulation of the upper boundary conditions as given in Equation 12. To circumvent this issue, Schorghofer proposed a workaround [8] by linearizing the black body emission around a reference temperature T_r:

$$T_0 = \left(T_\mathrm{r} + \delta T\right)^4 \approx T_\mathrm{r}^4 + T_\mathrm{r}^3 \delta T \tag{16}$$

However it should be noted that T_r is chosen to be equal to the previous surface temperature T_0, making this formulation explicit. This approach led to the development of a new boundary condition formulation which is sensible to abrupt changes of energy input, causing the surface temperature to vary rapidly. If the reference temperature is far from the actual surface temperature this causes a significant error in the evaluation of the emitted energy. To address this issue, Schorghofer [8] uses a predictor-corrector step that iteratively adjusts the reference temperature.

We choose to formulate the boundary surface flux explicitly for the MultIHeaTS solver, following the same approach as for the explicit Euler scheme. Our validation tests, discussed in Section 3, demonstrate that this approach does not introduce large errors and the algorithm remains stable. One advantage of the implicit Euler scheme is that an explicit formulation of the surface flux can be used, while maintaining stability and adding little to no errors [21]. We propose that this robustness may be due to the strong damping of oscillations discussed in Section 2.4. Similar to Schorghofer's solver, the implicit scheme may encounter issues when there are abrupt changes in surface temperatures. Nonetheless, the scheme is capable of accurately estimating the upper boundary condition using an explicit method and maintaining stability.

2.4 Stability and Accuracy of the Finite Difference Schemes in Homogeneous Media

Before analyzing the behavior of finite difference schemes in heterogeneous media, it is instructive to perform a stability analysis in simpler homogeneous media. We follow the Fourier transform approach as described in [11, 22–25]: by injecting wave components of different frequencies $\phi \in [-\pi, \pi]$ into the heat Equation 1, we can look at the response and behavior of the different schemes.

The propagation of the finite difference solution from one time step to the next is governed by the growth factor which for the explicit method is:

$$g(\phi) = 1 - 4F \sin^2 \frac{\phi}{2} \tag{17}$$

where values of ϕ close to 0 indicate low frequency components and values close to π indicate high oscillatory wave components and F is the Fourier number defined as:

$$F = \frac{\alpha \Delta t}{\Delta x^2} \tag{18}$$

The growth of numerical wave components occurs when the absolute value of $g(\phi)$ exceeds 1, which for explicit scheme happens if $F > \frac{1}{2}$. This leads to the magnification of the corresponding wave component, indicating instability. Consequently, the explicit scheme is numerically unstable, and its domain of stability is given by:

$$F < 1/2 \tag{19}$$

The same analysis for the Backward Euler scheme leads to a growth factor of:

$$g(\phi) = \left(1 + 4F \sin^2 \frac{\phi}{2}\right)^{-1} \tag{20}$$

In this case, the growth factor is constrained by $|g(\phi)| \leq 1$ for all F and ϕ, meaning that the implicit scheme is unconditionally stable.

Finally the growth factor of the Crank-Nicolson method is given by:

$$g(\phi) = \frac{1 - 2F\sin^2\frac{\phi}{2}}{1 + 2F\sin^2\frac{\phi}{2}}. \tag{21}$$

It is clear that the Crank-Nicolson method is also unconditionally stable as $|g(\phi)| \leq 1$ for all values of F and ϕ. However we note that for high frequency components $\phi \to \pm\pi$ then $g(\phi) \to -1$, especially for large F, meaning that these components are propagated as weakly damped oscillations.

Notably, the growth factor of the implicit scheme is relatively small for high Fourier Numbers, which means that the components that exhibit the most problematic behavior in the Crank-Nicolson method are the same components that are most effectively damped by the implicit method [25]. Hence, while the Crank-Nicolson is second-order accurate it is *generally* stable for large timesteps, which is said to be L-Stable [17]. Conversely, the Backward Euler is stable for arbitrarily large time-step, said to be strongly A-stable [21], although it is only first-order accurate. This means that the MultIHeaTS solver is particularly valuable in the field of planetary sciences, where large timescales can restrict the precision of time steps.

3 Results

3.1 Validation with an Analytical Solution

Validation of our model using known analytical solutions of the heat equation can be achieved for homogeneous profiles only since no analytical solutions exist in the general case. If we consider a stiff initial condition given by a step function defined on $[0, L]$ by the expression :

$$\forall x \in [0, L],\, T(x, 0) = \begin{cases} 0, \text{ if } x < L/2 \\ 1, \text{ if } x \geq L/2. \end{cases} \tag{22}$$

then the analytical solution with zero-flux boundary conditions can be obtained (detailed derivation in Supplementary Materials Section 2) through Fourier series decomposition:

$$T(x,t) = \frac{1}{2} - \sum_{j=1}^{+\infty} \frac{2}{\pi j} \sin\left(\frac{\pi j}{2}\right) \cos\left(\frac{\pi j x}{L}\right) e^{-\alpha\left(\frac{\pi j}{L}\right)^2 t}. \tag{23}$$

To validate our model, we need to compare the computed temperatures with the analytical solutions for the same set of thermal parameters. This is done by computing the error, defined as the absolute value of difference between the temperature produced by the numerical model T and the reference temperature θ. The maximum error e_+

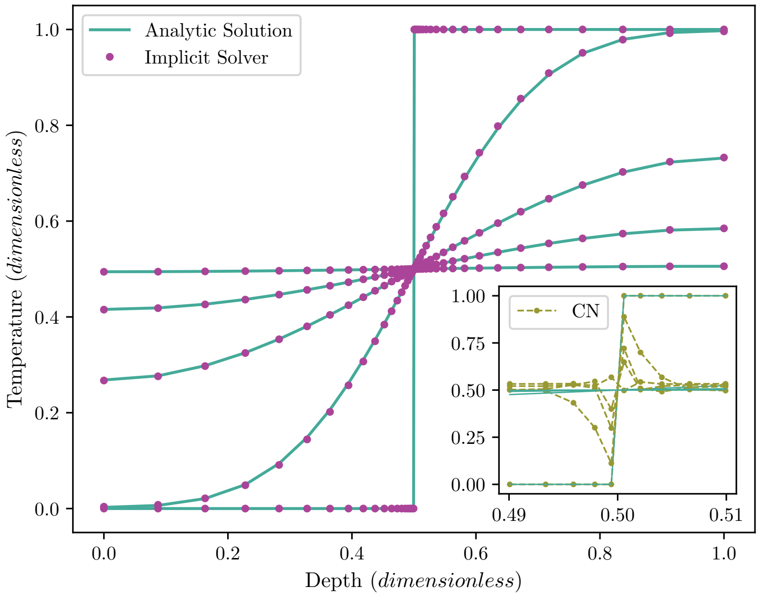

Fig. 1 Analytical validation for a stiff initial condition and homogeneous thermal properties on an irregular grid. Despite the discontinuity at $x = 0.5$, the implicit solver MultIHeaTS can compute the evolution of temperature with a very close match (maximum error $e_+ < 0.5\%$) to the analytic solution. (*Bottom Right*) Zoom on the spurious oscillations of the Crank-Nicolson solver at the location of the initial discontinuity.

is expressed as :

$$e_+ = \max_{n,i} \left| T_n^i - \theta_n^i \right| \tag{24}$$

and the average error $\overline{e}$ as :

$$\overline{e} = \frac{1}{n_x n_t} \sum_{n,i} \left| T_n^i - \theta_n^i \right| \tag{25}$$

with θ_n^i the reference temperature taken at the exact same location x_n and time t^i. For proper validation, both the numerical solver and the reference solution need to be calculated until the reach of a stationary regime, with exactly the same thermal properties and conditions (see Figure 1). For a set of $n_x = 40$ grid points defining the box of length $L = 1\,\mathrm{m}$ of constant diffusivity $\alpha \approx 0.55\,\mathrm{m^2\,s^{-1}}$ and $n_t = 700$ iterations of timestep $\Delta t = 2.3\,\mathrm{ms}$, the maximum error e_+ between the analytical solution and the implicit solver is less than 0.5% and the mean error $\overline{e}$ is under 0.02%. Results show a very close fit between the numerical and analytic solutions which proves that the MultIHeaTS model works well for homogeneous media.

The same test was performed using the Crank-Nicolson solver [8] to evaluate its ability to handle stiff initial conditions. Using the same parameters as previously, the Crank-Nicolson solver produced spurious oscillations that were located around the initial discontinuity position (Figure 1, *Bottom Right*). The maximum error e_+ between the analytical solution and the Crank-Nicolson solver reaches up to 38% and the mean error $\overline{e}$ averages 3%. Although these oscillations eventually slowly disappear with time [25], it demonstrates that despite being the most accurate finite difference scheme, the Crank-Nicolson solver is less reliable in handling stiff initial conditions [11, 17, 25], such as when two materials with different temperatures come into contact.

3.2 Validation by Comparison with Explicit Scheme

In more complex cases, in particular when the thermal conductivity vary with depth, the heat equation becomes too complex to solve analytically [1]. In these cases, validation of the model is typically performed through comparison with experimental data and/or other well-established numerical algorithms. Since our target application is the study of realistic planetary surface conditions, we decided to compare our method with Spencer's explicit scheme [7], which is a commonly used algorithm in planetary science. Spencer's algorithm was implemented in IDL and can be obtained from his personal website [2].

Parameters	Value
Distance to Sun d	9.51 AU
Emissivity ϵ	1
Albedo A	0.015
Grid points n_x	100
Max Depth	2 m
Initial Temperature T	90 K
Solar Period P	79.3 days
Number of Periods	5
Number of iterations n_t	50000
Time Step Δt	685 s

Table 1 Physics and numerical parameters used for comparing Spencer algorithm with MultIHeaTS.

Thermal Properties	Unit	Top value	Bottom value	Interface
Density ρ	$\mathrm{kg.m^{-3}}$	800	2000	25 cm
Heat Capacity c_p	$\mathrm{J.kg^{-1}.K^{-1}}$	600	1800	50 cm
Inertia I_th	$\mathrm{J.m^{-2}.K^{-1}.s^{-1/2}}$	200	20	100 cm

Table 2 Thermal properties of a bi-layered surfaced used for comparison of Spencer's explicit algorithm with MultIHeaTS. Although some of these values are close to what could be found on realistic icy surfaces, they were varied smoothly over large scales for validation purposes.

[2] https://www.boulder.swri.edu/~spencer/thermprojrs/

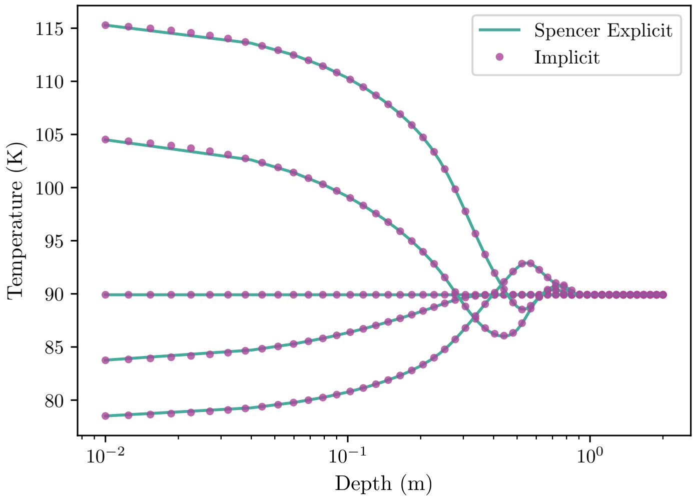

Fig. 2 Validation of MultIHeaTS against Spencer's explicit algorithm. Temperature profiles for different times are plotted simultaneously i = 0, 20, 159, 300, 699. MultIHeaTS is as accurate as Spencer's explicit solver with the advantage of being unconditionally stable. Note the uneven grid spacing of MultiHeaTS, denser at the surface, shown by the logarithmic horizontal axis.

Both algorithms were run with meticulous attention with the exact same thermal properties, boundary conditions, and numerical parameters that can be found in Table 1. For the sake of validation, values of density, heat capacity and conductivity are varied through the depth by the largest scales allowed by Spencer's explicit scheme stability criteria (see Table 2). The MultIHeaTS solver exhibited a computational time of less than 1 minute on an Intel CPU i7-10750.

Overall, the results, presented in Figure 2, show a very good similarity between the temperature profiles produced by the explicit and implicit schemes, with negligible differences between the two methods. The maximum error between MultIHeaTS and the Spencer's solver is $e_+ = 0.68\%$ and the mean error is $\bar{e} = 0.052\%$. This comparison demonstrates that the MultIHeaTS solver can accurately compute temperature profiles for planetary surfaces, without the limitations of conditional stability that can be encountered with explicit methods.

3.3 Comparison with Crank-Nicholson for Heterogeneous Media

Another finite-differences implementation of the same heat transfer problem using the Crank-Nicolson scheme for heterogeneous planetary surfaces has been developed by Schorghofer [8] and is accessible online on his Github repository[3]. Unlike the explicit scheme, the Crank-Nicolson scheme is unconditionally stable and second-order accurate [19], which is an advantage in terms of accuracy and stability. However for more

[3] https://github.com/nschorgh/Planetary-Code-Collection

complex and realistic cases, such as heterogeneous surfaces subject to surface energy equilibrium, it is unclear whether Crank-Nicolson remains the most reliable. In this section, we propose to evaluate the performance of the Crank-Nicolson scheme and our MultIHeaTS implicit scheme in comparison with the reference explicit solver outside the stability zone of the explicit algorithms. This will allow us to assess the reliability and accuracy of each scheme in simulating heat transfer in more challenging and realistic scenarios.

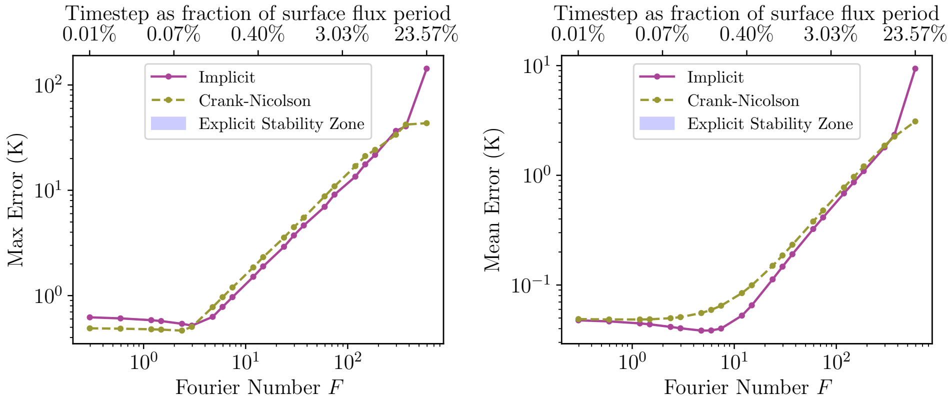

Fig. 3 Errors produced by the Crank-Nicholson and our MultIHeaTS implicit scheme in comparison to Spencer's explicit model as functions of the timesteps Δt used (*log-scales*). Average temperature is around 90 K. Timestep Δt is represented as a fraction of the surface flux' period at the top and by the Fourier number $F = \alpha \Delta t / \Delta x^2$ at the bottom. The stability zone of the explicit scheme $F < 1/2$ is represented in light blue. (Left) Maximum error e_+ produced by both solvers. (Right) Mean error $\overline{e}$ produced by both solvers.

To evaluate the effect of increasing timestep on the accuracy of our MultIHeaTS implicit scheme and the Crank-Nicolson scheme, we conducted a numerical experiment where we varied the timestep Δt while keeping the grid and thermal properties fixed. We used Spencer's explicit algorithm with a small timestep $\Delta t = 2 \times 10^{-5} P$ as a reference for the accurate computed temperatures. The properties and parameters used in this experiment are the same as for the previous experiments and can be found respectively on Table 1 and 2. The increasing timestep causes the algorithms to lose information, leading to differences in the computed temperatures compared to Spencer's reference algorithm.

The results of our numerical experiment are presented in Figure 3. As expected, both the Crank-Nicholson and our MultIHeaTS implicit scheme are able to compute the temperature profile outside the stability zone ($F > 1/2$) but with increasing error as F increases. We consider, arbitrarily, for maximum error larger than $e_+ = 10\%$ that algorithms are irrelevant. We observe that the unconditional stability of both algorithms allows us to extend the domain of computation to about two orders of magnitude beyond the explicit stability limit, while keeping the error reasonably low.

In all cases, we observed that the Crank Nicolson method is on average 30% computationally faster per iteration than our implicit algorithm. Near the explicit stability zone $F < 1/2$, both Schorghofer's Crank-Nicolson ($e_+ = 0.53\%$, $\overline{e} = 0.054\%$) and our MultiHeaTS algorithm ($e_+ = 0.68\%$, $\overline{e} = 0.052\%$) produced near equivalent results with a slightly lower maximum error for Crank-Nicolson. However, for larger timesteps at fixed grid and thermal properties, the difference in errors becomes more noticeable. Specifically, for F values ranging from 1 to 100, the MultIHeaTS algorithm is more accurate on average and for the maximum error. For example, at $F = 60$, the maximum and mean errors for MultIHeaTS are $e_+ = 7.6\%$ and $\overline{e} = 0.36\%$, respectively, while for Crank-Nicolson, the corresponding values are $e_+ = 9.6\%$ and $\overline{e} = 0.42\%$. Beyond the $F = 100$ threshold, the errors produced by both algorithms become too high for the output temperatures to be relevant.

The results of our study indicate that when working with timesteps that exceed the explicit stability zone, our implicit solver MultiHeaTS is equally reliable and may even be more accurate than the Crank-Nicolson solver. This finding is particularly relevant for planetary studies, where it is often necessary to analyze temperature variations over several years ($t^{n_t-1} \gg P$). In such scenarios, it is often computationally expensive to obtain a precise timestep for the diurnal evolution, emphasizing the importance of a more accurate solver capable of handling larger timesteps.

4 Conclusion

We have developed an open-source implicit algorithm called MultIHeaTS, which employs finite differences to solve the heat equation on 1D heterogeneous media with an irregular grid. While our primary focus is planetary science, our algorithm is adaptable and can accommodate different types of boundary conditions and surfaces.

For homogeneous cases, the algorithm was validated using a known analytical solution. This validation which used a discontinuous initial equation showed the robustness MultIHeaTS to stiff equations, in contrary to the Crank-Nicolson method. For heterogeneous cases, MultIHeaTS was validated by comparison to a well established explicit algorithm in planetary science [7]. Despite our explicit formulation of the surface energy equilibrium condition, our implicit scheme remained accurate and stable, producing results that closely matched Spencer's model. Thanks to its unconditional stability, this implicit Euler scheme allow us to extend the stability domain by two orders of magnitude, beyond which the truncation error becomes too significant ($e_+ > 10\%$).

In a numerical study, we compared the errors produced by our algorithm with an existing implementation of the Crank-Nicolson scheme by Schorghofer [8]. Our results show that both models are equivalent for low Fourier numbers, and for higher Fourier numbers, our MultIHeaTS solver becomes more accurate than Crank-Nicolson. For instance, at $F = 60$, the errors in MultIHeaTS are $e_+ = 7.6\%$ and $\overline{e} = 0.36\%$, while for Crank-Nicolson, the corresponding errors are $e_+ = 9.6\%$ and $\overline{e} = 0.42\%$.

Future research could investigate how the accuracy and stability of the MultIHeaTS algorithm are affected by the explicit formulation of the nonlinear upper boundary condition induced by black body emission. This would provide valuable insight into the behavior of the algorithm under more complex thermal conditions.

The MultIHeaTS algorithm has a wide range of potential applications, and its open-source and flexible nature make it adaptable to various domains. In planetary science, it is expected to be useful for retrieving thermal properties of multi-layered planetary surfaces using brightness temperature data. Moreover, due to its finite-difference formulation, it can be easily coupled with other physical processes to study, for example, the evolution of planetary icy surfaces.

Statements and Declarations

Funding and Competing Interests

We acknowledge support from the "Institut National des Sciences de l'Univers" (INSU), the "Centre National de la Recherche Scientifique" (CNRS) and "Centre National d'Etudes Spatiales" (CNES) through the "Programme National de Planétologie".

The authors have no competing interests to declare that are relevant to the content of this article.

Data Availability

The authors declare that the data supporting the findings of this study are available within the paper and on the online public repository https://gitlab.dsi.universite-paris-saclay.fr/cyril.mergny/multiheats

References

[1] Jaeger JC. Conduction of heat in composite slabs. Quarterly of Applied Mathematics. 1950;8(2):187–198. https://doi.org/10.1090/qam/36417.

[2] Loeb A, Earls C. Analysis of heterogeneous computing approaches to simulating heat transfer in heterogeneous material. Journal of Parallel and Distributed Computing. 2019 nov;133:1–17. https://doi.org/10.1016/j.jpdc.2019.06.004.

[3] Roubíček T. Numerical solution of the nonlinear heat equationin heterogeneous media. Numerical Functional Analysis and Optimization. 1990 jan;11(7-8):793–810. https://doi.org/10.1080/01630569008816402.

[4] Lage PLC. Application of the optimized modified strong implicit procedure to nonlinear problems. Numerical Heat Transfer, Part B: Fundamentals. 1996 dec;30(4):423–435. https://doi.org/10.1080/10407799608915091.

[5] Nissen A, Keilegavlen E, Sandve TH, Berre I, Nordbotten JM. Heterogeneity preserving upscaling for heat transport in fractured geothermal reservoirs. Computational Geosciences. 2017 dec;22(2):451–467. https://doi.org/10.1007/s10596-017-9704-6.

[6] Masson R, Jeannin L, Louvet F, Vuddamalay A. Domain decomposition methods to model heat exchanges between a well and a rock mass. Computational Geosciences. 2020 mar;24(3):1377–1392. https://doi.org/10.1007/s10596-020-09957-2.

[7] Spencer JR, Lebofsky LA, Sykes MV. Systematic biases in radiometric diameter determinations. Icarus. 1989 apr;78(2):337–354. https://doi.org/10.1016/0019-1035(89)90182-6.

[8] Schorghofer N. Fast numerical method for growth and retreat of subsurface ice on Mars. Icarus. 2010 aug;208(2):598–607. https://doi.org/10.1016/j.icarus.2010.03.022.

[9] Fudym O, Ladevie B, Batsale JC. A Seminumerical Approach for Heat Diffusion in Heterogeneous Media: One Extension of the Analytical Quadrupole Method. Numerical Heat Transfer, Part B: Fundamentals. 2002 oct;42(4):325–348. https://doi.org/10.1080/10407790190053978.

[10] Fang J, Zhao GF, Zhao J, Parriaux A. On the Truly Meshless Solution of Heat Conduction Problems in Heterogeneous Media. Numerical Heat Transfer, Part B: Fundamentals. 2009 jan;55(1):1–13. https://doi.org/10.1080/10407790802605067.

[11] Langtangen HP, Linge S. Finite Difference Computing with PDEs. Springer International Publishing; 2017.

[12] Acton CH. Ancillary data services of NASA's Navigation and Ancillary Information Facility. Planetary and Space Science. 1996 jan;44(1):65–70. https://doi.org/10.1016/0032-0633(95)00107-7.

[13] Acton C, Bachman N, Semenov B, Wright E. A look towards the future in the handling of space science mission geometry. Planetary and Space Science. 2018 jan;150:9–12. https://doi.org/10.1016/j.pss.2017.02.013.

[14] Wesselink AJ. Heat conductivity and nature of the lunar surface material. Bulletin of the Astronomical Institutes of the Netherlands. 1948 Apr;10:351–363.

[15] Rozitis B, Green SF. Directional characteristics of thermal-infrared beaming from atmosphereless planetary surfaces - a new thermophysical model. Monthly Notices of the Royal Astronomical Society. 2011 jul;415(3):2042–2062. https://doi.org/10.1111/j.1365-2966.2011.18718.x.

[16] Kieffer HH. Thermal model for analysis of Mars infrared mapping. Journal of Geophysical Research: Planets. 2013 mar;118(3):451–470. https://doi.org/10.1029/2012je004164.

[17] Press WH, Teukolsky SA, Vetterling WT, Flannery BP. Numerical Recipes in C (2nd Ed.): The Art of Scientific Computing. USA: Cambridge University Press; 1992.

[18] Crank J, Nicolson P. A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type. Mathematical Proceedings of the Cambridge Philosophical Society. 1947 jan;43(1):50–67. https://doi.org/10.1017/s0305004100023197.

[19] Mazumder S. Treatment of the Time Derivative (Parabolic and Hyperbolic PDEs). In: Numerical Methods for Partial Differential Equations. Elsevier; 2016. p. 219–275.

[20] Sundqvist H, Veronis G. A simple finite-difference grid with non-constant intervals. Tellus. 1970 feb;22(1):26–31. https://doi.org/10.1111/j.2153-3490.1970.tb01933.x.

[21] Beam RM, Warming RF, Yee HC. Stability analysis of numerical boundary conditions and implicit difference approximations for hyperbolic equations. Journal of Computational Physics. 1982 nov;48(2):200–222. https://doi.org/10.1016/0021-9991(82)90047-x.

[22] Pearson CE. Impulsive end condition for diffusion equation. Mathematics of Computation. 1965;19(92):570–576. https://doi.org/10.1090/s0025-5718-1965-0193765-5.

[23] Giles M. Stability Analysis of Numerical Interface Boundary Conditions for Parabolic Equations. University of Oxford Research Archive. 1995;.

[24] Thomas JW. Stability. In: Numerical Partial Differential Equations: Finite Difference Methods. In: Texts in Applied Mathematics. Springer New York; 1995. p. 97–145.

[25] Østerby O. Five Ways of Reducing the Crank–Nicolson Oscillations. BIT Numerical Mathematics. 2003 dec;43(4):811–822. https://doi.org/10.1023/b:bitn.0000009942.00540.94.