Computing and Data Analysis Project :

Study of heat diffusion in the ground using Finite Difference and Data Assimilation

Brune RAYNAUD–SCHELL

November 2022 - January 2023

Project tutor : Emmanuel COSME (IGE)

Master 2 STPE (Université Grenoble-Alpes)
Parcours Système Climatique : Atmosphère Hydrosphère, Cryosphère

Contents

1 Physical problem

1.1 Presentation of the case of study

For this project I decided to model the diffusion of heat in the ground which is a result of the incident solar radiations on the surface. In order to test different numerical schemes resolving the heat equation, I created at first the simplest experiment I could. The model is unidimensional (vertical direction) and the ground is considered homogeneous, without any stratification nor changes in structure (Figure 1). I made the hypothesis of a constant solar forcing. Which means that the boundary condition on temperature is constant at the surface and bottom layer (0°C). The initial conditions on temperature will be implemented as a gaussian function with a maximum in the middle of the space domain. All of theses assumptions and hypotheses are not supposed to be realistic. They are only meant to test my numerical models first on a very simple case to practice and then make it more complex with different parameters and use data assimilation to improve the modelisation.

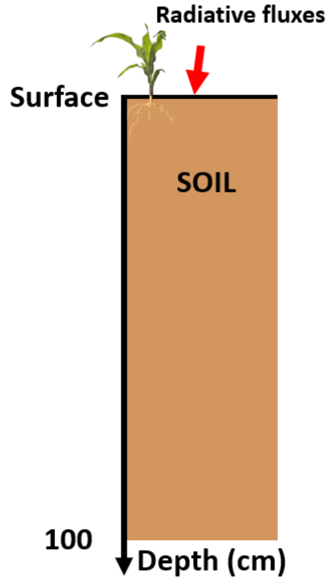


Figure 1: Schematic representation of the case of study: column of soil receiving solar radiations

1.2 Analytical solution of unsteady heat equation

The heat conduction equation shown below is a partial differential equation describing the evolution of temperature in time and space:

$$\frac{\delta T}{\delta t} - K\left(\frac{\delta^2 T}{\delta x^2}\right) = 0 \tag{1}$$

, T being the temperature ($°C$) and K the thermal conductivity (diffusivity coefficient) of the soil ($W \cdot m^{-1} \cdot ^\circ C^{-1}$). It can be noted that the equation is written without a source term.

In order to calculate the temperature at each step of time and space, the analytical solution is needed. The demonstration of how to obtain the solution was not a part of my work but according to Hamilton et al. (2007) it writes:

$$T(x,t) = \frac{1}{2\sqrt{\pi K t}} \int_0^L f(y,t_0) \exp - \left(\frac{(x-y)^2}{4Kt}\right) dy \tag{2}$$

, $f(y, t_0)$ corresponds to the initial condition of temperature at all depth (gaussian) and L is the limit of the space domain.

I represented on the Figure 2 the time evolution of the temperature field in the ground. As time passes the temperature decreases at the center of the domain and rises on the sides of the center. With conduction the heat is spreading from the center to the sides.

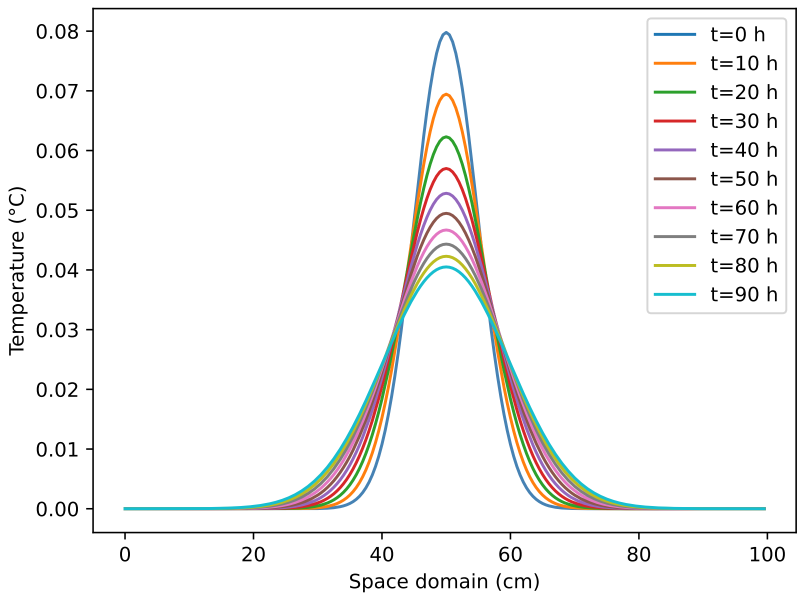


Figure 2: Temperature field in the ground at different moments calculated with the analytical solution of the heat equation (dx= 0.5 and K=0.4)

2 Numerical resolution with Finite Difference

I then implemented severall finite difference schemes to resolve the heat equation numerically. It is a first order time derivative and second order space derivative equation that can be approximated using numerical models with a range of levels of time and space. Theses kinds of methods require little computational ressources and the most basic packages on Python. The numerical solutions are tested and compared to the analytical one. Everything is detailed in the scripts on my Github page .

2.1 Different Finite difference schemes

2.1.1 Euler schemes

Euler numerical schemes are the simplest way to model. The scheme can be explicit as in equation (3), only needing the temperature field at the previous time step to give the one at the present time: forward in time and centered in space. Equation (4) is the backward implicit Euler scheme and its numerical resolution requires to solve a linear system at each time step.

$$u_j^{n+1} = u_j^n + \frac{K dt}{dx^2}\left(u_{j-1}^n + u_{j+1}^n - 2u_j^n\right) \tag{3}$$

$$u_j^{n+1} = u_j^n + \frac{K dt}{dx^2}\left(u_{j-1}^{n+1} + u_{j+1}^{n+1} - 2u_j^{n+1}\right) \tag{4}$$

2.1.2 Dufort-Frankel scheme

Dufort-Frankel scheme requires second order of time and space and it can be made into an explicit scheme quite easily as shown on the equations (5) to (7) below . When calculating the temperature field for the next time step, the two previous ones are required. I still used the same initial condition for T(t=0,x) and used the temperature field T(t=1,x) from the forward Euler scheme to initiate Dufort-Frankel.

$$
\begin{aligned}
\frac{u_j^{n+1}-u_j^{n-1}}{2\Delta t} &= \frac{\sigma}{\Delta x^2}\left(u_{j-1}^{n}-u_j^{n+1}-u_j^{n-1}+u_{j+1}^{n}\right) && (5)\\
u_j^{n+1}-u_j^{n-1} &= \frac{\sigma 2\Delta t}{\Delta x^2}\left(u_{j-1}^{n}-u_j^{n+1}-u_j^{n-1}+u_{j+1}^{n}\right) && (6)\\
u_j^{n+1} &= \frac{1-A}{1+A}u_j^{n-1}+\frac{A}{1+A}\left(u_{j-1}^{n}+u_{j+1}^{n}\right) && (7)
\end{aligned}
$$

with $A = \frac{\sigma 2\Delta t}{\Delta x^2}$

2.1.3 Crank-Nicolson scheme

The Crank-Nicolson scheme is an implicit θ-scheme with $\theta = 1/2$ and second order in time and space. The equation (8) can be rearanged separating the terms depending on T at the present time an T at the previous time (equation (9)). It could be resolved by solving a linear system like I did with the implicit Euler scheme but it was resolved with a variationnal method instead. The goal was to use a cost function and minimize it in order to find the best estimate for the temperature field of the next time step.

$$
\frac{u_j^{n+1}-u_j^{n}}{\Delta t} = \theta\frac{K}{\Delta x^2}\left(u_{j-1}^{n+1}-2u_j^{n+1}+u_{j+1}^{n+1}\right)+(1-\theta)\frac{K}{\Delta x^2}\left(u_{j-1}^{n}-2u_j^{n}+u_{j+1}^{n}\right) \tag{8}
$$

$$
-\theta A u_{j-1}^{n+1}+(1+2\theta A)u_j^{n+1}-\theta A u_{j+1}^{n+1} = (1-\theta)Au_{j-1}^{n}+(1-2A+2\theta A)u_j^{n}+(1-\theta)Au_{j+1}^{n} \tag{9}
$$

with $A = \frac{K\Delta t}{\Delta x^2}$

2.2 Precision, accuracy and stability

The first thing that can be said from the implementation of those numerical schemes is that the simulation of temperature fields is working very well and that it is not at all costly in terms of code, very much less that the analytical solution. I did not insert figures of the simulated temperature because graphically it does not differ much from the temperature fields calculated from the analytical solution (except Crank Nicolson). To have a quantitative estimation of the accuracy of the numerical solutions I calculated the Root Mean Square Error between each numerical model (p) and the analytical solution (y) in the expression below. I needed first to make sure that I resampled the analytical solution temperature fields to the size of the numerical schemes because they had different spatial resolutions (dx=0.1 cm for the first and dx=0.5 cm for the others).

$$
RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i-p_i)^2}{n}}
$$

The error evolution in time for each finite difference scheme temperature prediction compared to the analytical solution is plotted in Figure 3. The temperature values predicted tend to be further and further from the "truth" as the time of the simulation goes on. We can also see that Crank-Nicolson is less accurate than the 3 others by two orders of magnitude. But it is not a major concern since the value of the error is not very high (not exceeding 0.003 °C). The errors for Euler and Dufort-Frankel schemes are growing at the same pace approximatively and in the same range, Durfort-Frankel being the most accurate.

The next step was to test the influence of the space discretization of the numerical schemes on the accuracy of the temperature field prediction. I calculated the cumulated RMSE (at the time of simulation) for each space step dx (from 0.2 cm to 12 cm) and it is represented in the Figure 4. As in the previous figure, Crank-Nicolson has higher order error than the other schemes and also a completely different shape. The 3 others have quite the same evolution with the increasing dx but with a small dx Durfort-Frankel has oscillations. When the dx is large, above 8 cm the RMSE does not evolve at the same rate and oscillates. It must be because the discretization is not fine, errors tend to be larger

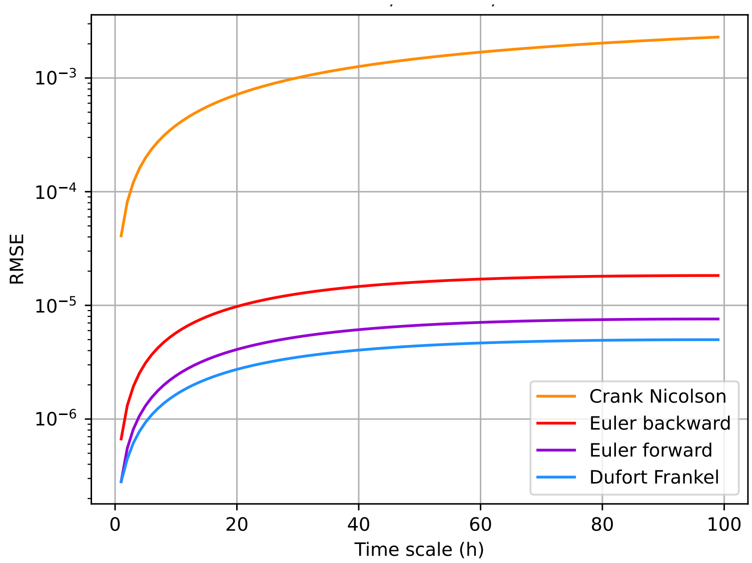


Figure 3: RMSE between finite difference solutions and analytical solution of the heat equation (dx= 0.5 and K=0.1)

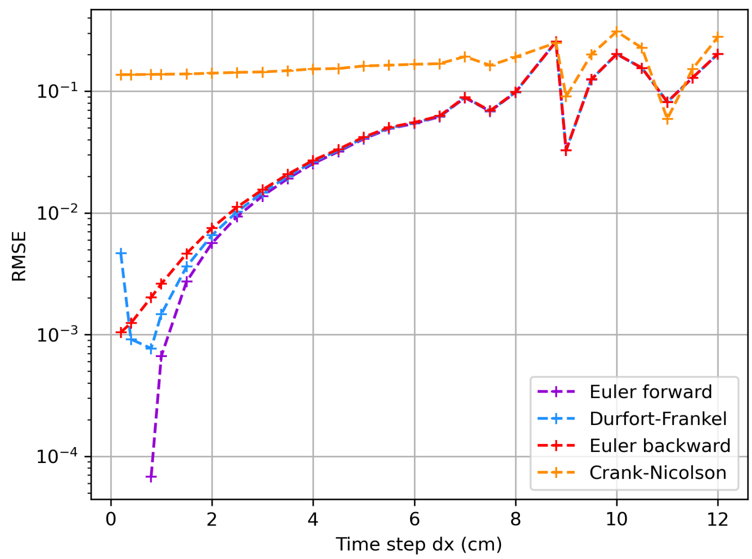


Figure 4: Cumulated RMSE between finite difference solutions and analytical solution of the heat equation with varying dx (K=0.1)

The Euler forward cumulated RMSE is the only one starting at dx = 0.8 cm instead of 0.2cm. The reason behind it is stability conditions. Dufort Frankel, Crank Nicolson and the Euler backward schemes are unconditionally stable for any values of Δx and Δt. The explicit Euler scheme is the only one having a strong constraint on the space and time sampling: $\frac{K\Delta t}{\Delta x^2}$ must be $\leq 1/2$. (Detailed in the Appendix).

3 Data Assimiliation with 3D VAR

3.1 More realistic approach

Once the efficiency of the different numerical models is tested and their accuracy compared with the simplest set-up, the model can be improved to represent reality better: different boundary conditions are implemented and a data assimilation system is used

3.1.1 Surface boundary conditions

Before, the model had homogeneous Dirichlet boundary conditions with a fixed value of 0 °C at the surface and bottom part of the soil layer. And the initial conditions were represented by a gaussian function with the maximum centred around 50 cm deep. None of it is realistic. So I chose to implement periodic boundary conditions on the surface temperature to model the diurnal cycle, and the initial condition function is now maximum at the surface since the source of heating is there.

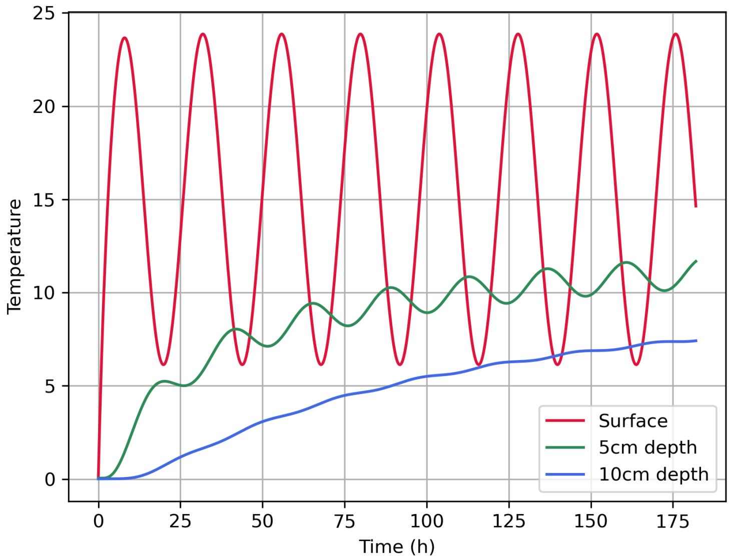
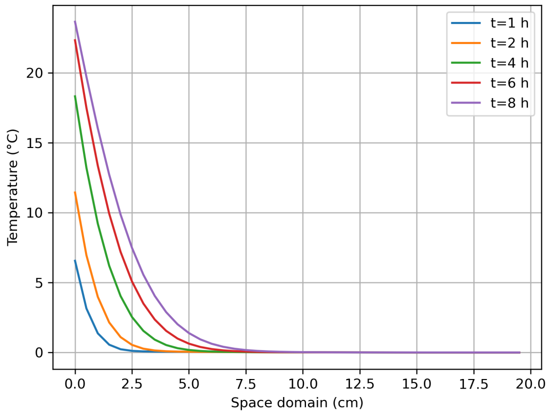
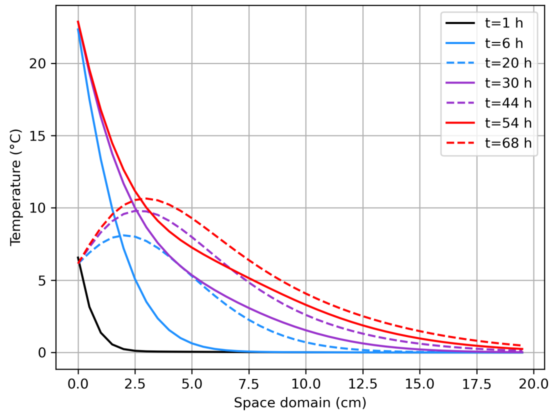


Figure 5: Surface boundary conditions describing diurnal cycle of temperature and evolution of temperature at 5 and 10 cm deep simulated by Euler scheme model of heat equation

The new boundary condition is represented in red in Figure 5 and is a perfect cycle of temperature in 24h without variability. Since the soil column was first completely cold (0°C), the temperature below tends to rise and has oscillations. The deeper it is the more the amplitude is attenuated and the shift between the daily max at the surface and lower is big, as was expected. In Figure 6 I represented the temperature fields in the ground at different stages of heat diffusion. On the left, we can see the initialization (gaussian form) for the first few hours as the temperature is rising. The temperature is diffused in depth but only up to 7.5 cm deep after 8 hours. On the right side, the modelled temperature fields are plotted for a longer time scale. In dashed lines are the fields when it is night (minimum temperature) and the full lines are when the temperature is maximum at the surface. The heat reaches deeper and deeper at each time step with the diffusion (20 cm, 68h after the beginning of the simulation). And since our boundary condition signal is periodic and not constant anymore the temperature fields have different shapes if it is the night or in the hottest hours of the day.



(a) First few hours



(b) First few days

Figure 6: Modeled temperature field in the ground (Euler forward scheme) at different times (dx=0.5 cm and K=0.3)

3.1.2 Creating Dataset

I needed in-situ measurements of temperature because I wanted to check if I gave realistic temperature cycles as boundary conditions on the surface and to help implement a Data assimilation method to improve the model. Since I did not have real soil temperature data available, I created my own dataset. My first hypothesis was to have thermometers placed at 5cm and 10 cm depth. My observations were made with a periodic function to which I added some noise and variation. I tried to shift the oscillations because of the time lag between heating on the surface and when it reaches deeper points, the results are shown in Figure 7. All of this was done arbitrarily, so it is not completely realistic, for example, every day is pretty much the same. Plus, it is quite close to the outputs of my model because I got inspired by it. I don't know whether it is "good" data or not but it is enough for what i need later on in the data assimilation method.

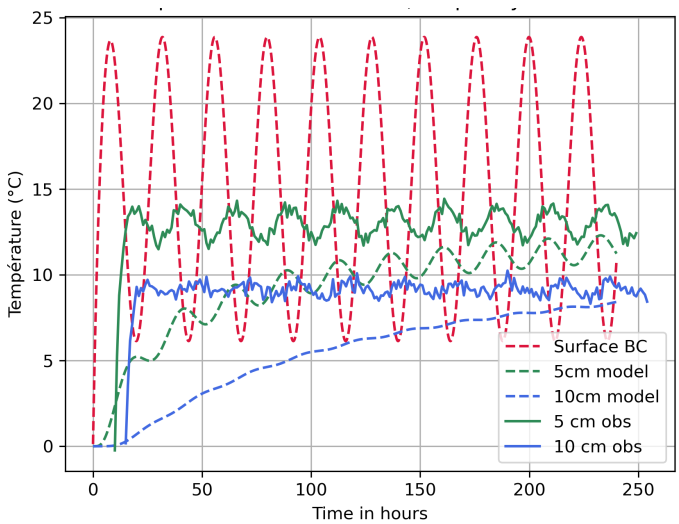


Figure 7: Evolution of temperature in the ground as predicted by the model (dashed lines) and as measured (full lines) K=0.6

3.2 3D VAR implementation

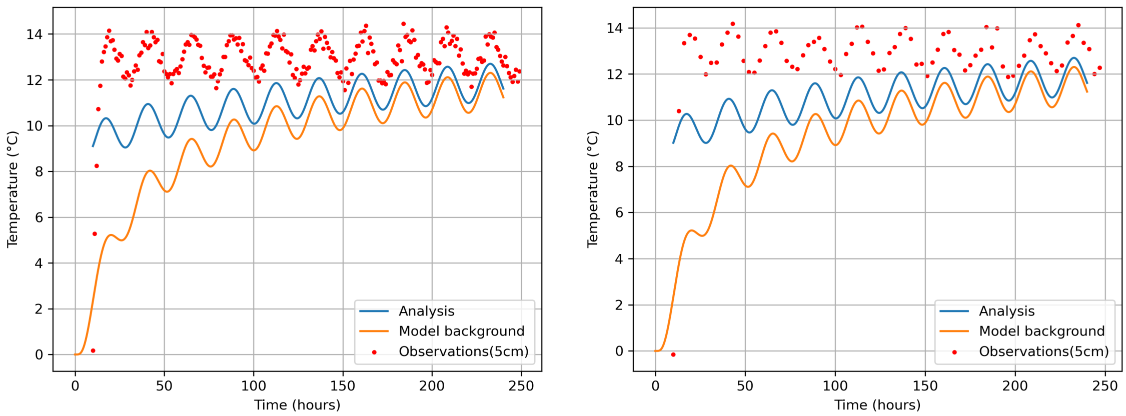
To make my model better I am implementing a data assimilation system which incorporates observations to constrain the model. The goal was to experiment with 3DVAR method because it was not covered in the courses this semester. This algorithm makes an estimate of the state of the system by variational minimization of the cost function. It follows two steps:

- Forecast: model predicts temperature field for the next time step knowing the last one (finite difference).
- Analysis: At each time when an observation is available, minimization of the Cost function (equation (10)) in order to give the optimized temperature field (x^a) at each analysis time.

$$J(x) = \frac{1}{2}(x - x^b)^T B^{-1}(x - x^b) + \frac{1}{2}\left(y - H(x)\right)^T R^{-1}\left(y - H(x)\right) \tag{10}$$

x_b is the background vector which is the estimated state, H is the observation operator, linking the state vector x to the observations y, B and R are the covariance matrices associated with the errors on the background and errors on the observations.

3.3 Impact on modelling, sensitivity



(a) Frequency 1 hour

(b) Frequency 3 hours

Figure 8: Heat diffusion in the ground. Data assimilation method (3DVAR), Finite difference model using and temperature measurements (dx=1cm and K=0.6)

On figure 8 is represented the temperature evolution at 5cm depth with the measurements, the prediction of the model and the analysis. The data assimilation algorythm is supposed to give a better result. Here it seems that the observations are not taken into account, at least not enough. I tried to change different parameters to give more confidence in the observations and tried to fix whatever problem there is but I didn't manage to find the solution in time. I tried different frequencies of measurement to observe how the analysis would change (Figure 8). But as I said there is an issue elsewhere, so it obstruates any result I could find. There is nevertheless something happening in the analysis step because it modifies the temperature fields where there are observations (Figure 11 in the Appendix). I am not yet able to have a functionning 3DVAR. I tested Data assimilation with a Kalman Filter for which I already had some practice but it wasn't any better. The issue must be deeper.

Conclusion and learnings

To sum up I would like to give some points that I would improve in my work:

- For my study, I considered a very simple case, with a homogeneous soil. But it is usually different texture and structure of the soil close to the surface and deeper, which will have an effect on the thermal conductivity. Also if there is rain the heat diffusion will be changed, because moist soil have an increased conductivity. It could be interesting to add rainfall data and use it for data assimilation as an indirect source of data. We could do the same with nebulosity measurement, because clouds obstruct the amount of received solar radiation on the ground. It will allow to have a modulation of the perfect diurnal temperature cycle.
- The data assimilation part has not been very successful, I should have tackled it earlier. I would have allowed to have time to find out what is missing with my tutor and then improve it. Even with help from all the notebooks we studied in class and a lot of time trying every idea I could think of I was not able to find the solution to my problems. Mabye doing this project by pair would make us more efficient, have more perspective on what we are doing and help us to fix some technical issues together.

While working on this project I learned some skills that will be useful later on :

- In terms of methodology, I learned or at least was reminded that taking small steps with simple cases helps to understand everything better. It allowed me to later make the subject progressively more complex without being lost.
- This project helped me to understand better the notions we studied in class by applying it on my own and made me more curious to know how it is done. It made me also practice more some mathematical concept and it is always a good news for me.
- I really made an effort to have discipline and organization with my coding, because I am still new to it and it doesn't comme naturally. I test a lot of different approachs and ideas it is easy for me to get lost in messy notebooks. So I experimented using python libraries to store all the functions I created in order to have an easier working experience. I also made an effort for my scripts to be easily understandable and accessible by commenting a lot, using Markdown and uploaded everything on Github. I think it is good practice for my future research activities and teamwork.

APPENDIX

All scripts created and used during this project can be found open access on my Git-Hub repository: https://github.com/brune-RS/M2-Projet-numerique--Finite-difference--Data-assimilation-.git

Stability analysis

For the numerical resolution of the heat equation with Euler forward scheme: The demonstration with Fourier series and Von Neumann analysis had already been done in class with Romain Brossier (Finite element 2022).

Dufort Frankel is one of the simplest scheme to be unconditionnally stable. The demonstration can be found on this page https://folk.ntnu.no/leifh/teaching/tkt4140/._main064.html

For Crank Nicolson

Example 4. *For the Crank–Nicolson scheme*

$$v_m^{n+1} = v_m^n - \frac{a\lambda}{4}(v_{m+1}^{n+1} - v_{m-1}^{n+1} + v_{m+1}^n - v_{m-1}^n)$$

we obtain

$$g(\theta) = \frac{1 - \frac{1}{2}ia\lambda\sin\theta}{1 + \frac{1}{2}ia\lambda\sin\theta} \qquad \textit{thus} \qquad |g(\theta)|^2 = \frac{1 + (\frac{1}{2}a\lambda\sin\theta)^2}{1 + (\frac{1}{2}a\lambda\sin\theta)^2} = 1$$

so this scheme is unconditionally stable.

Figure 9: Von Neumann analysis for the Crank Nicolson scheme, (from Plamen Koev course in 2012)

The implicit Euler scheme is also unconditionnally stable. It has been demonstrated by Wen Shen (2018)

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = \frac{u_{j-1}^{n+1} - 2u_j^{n+1} + u_{j+1}^{n+1}}{\Delta x^2}.$$

Writing $\gamma = \Delta t/\Delta x^2$, we can write

$$-\gamma u_{j-1}^{n+1} + (1+2\gamma)u_j^{n+1} - \gamma u_{j+1}^{n+1} = u_j^n, \qquad (1)$$

Discrete Maximum Principle:
Scheme (1) could also be written as

$$(1+2\gamma)u_j^{n+1} = u_j^n + \gamma u_{j-1}^{n+1} + \gamma u_{j+1}^{n+1}.$$

$$\begin{aligned}(1+2\gamma)\left|u_j^{n+1}\right| &\le \left|u_j^n\right| + \gamma\left|u_{j-1}^{n+1}\right| + \gamma\left|u_{j+1}^{n+1}\right| \\ &\le \max_i\left|u_i^n\right| + \gamma\max_i\left|u_i^{n+1}\right| + \gamma\max_i\left|u_i^{n+1}\right| \\ &= \max_i\left|u_i^n\right| + 2\gamma\max_i\left|u_i^{n+1}\right|\end{aligned}$$

Since this holds for all j, it also holds when the left reaches the max.We conclude

$$(1+2\gamma)\max_j\left|u_j^{n+1}\right| \le \max_j\left|u_j^n\right| + 2\gamma\max_j\left|u_j^{n+1}\right|$$

$$\rightarrow \quad \max_j\left|u_j^{n+1}\right| \le \max_j\left|u_j^n\right|.$$

Figure 10: Demonstration of unconditionnal stability for euler backard scheme

Parameters

VARIABLES	UNITY	RANGE
Temperature (T)	°C	0 - 25
Thermal conductivity (k)	$W \cdot m^{-1} \cdot^{\circ} C^{-1}$	0.1 - 0.6
Spacestep (dx)	cm	0.1 - 12
Timestep (dt)	min/hour	1 min - 30 min

Table 1: Range of the different parameters value

Data assimilation tests

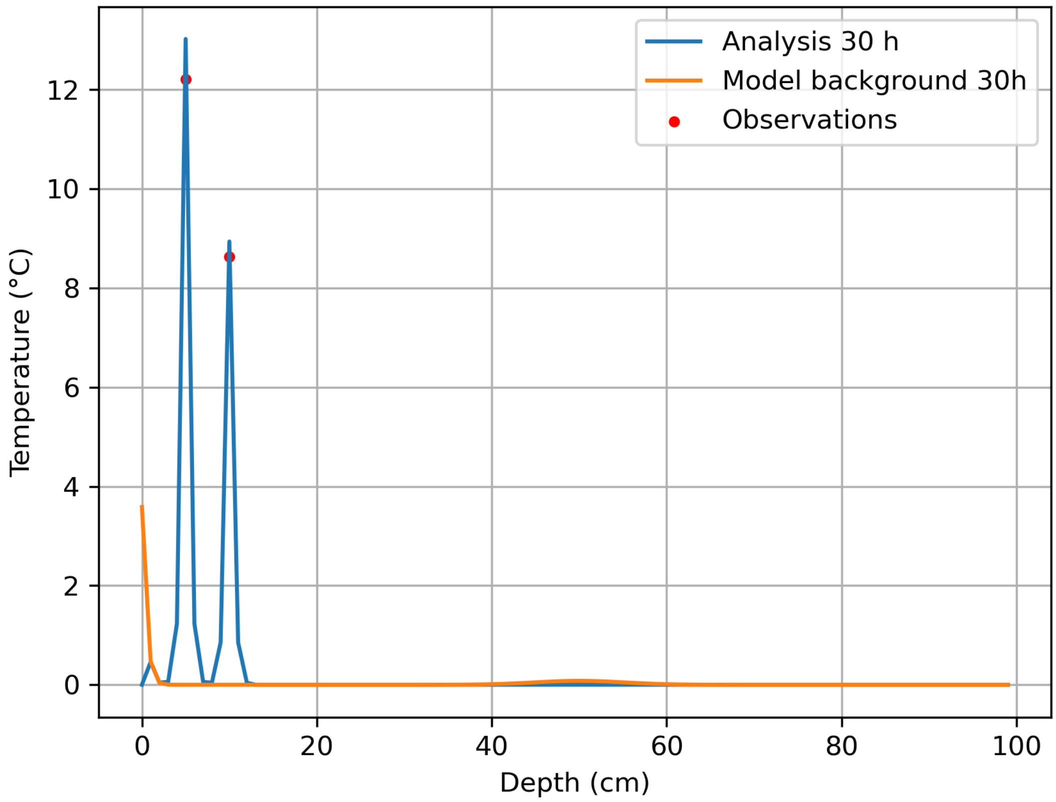


Figure 11: Temperature field at t=30h predicted by the model, and calculated usind 3Dvar

BIBLIOGRAPHY

Book:

- Hamilton, K., Ohfuchi, W. (Eds.). (2007). High resolution numerical modelling of the atmosphere and ocean. Springer Science Business Media.

Web ressources:

- Dufort Frankel scheme : http://cmth.ph.ic.ac.uk/people/a.mackinnon/Lectures/compphys/node35.html
- Calculation algorithm 3DVAR (ADAO): webpage here

Academic courses:

- Brossier, R., Numerical modelling-Finite difference, 2022.
- Cosme, E., Data Assimilation in geosciences, 2022.
- Plamen Koev, Numerical Methods for Partial Differential Equations, 2012 access