Project 3: The heat equation

Cruz Perez Gustavo Hazael

Superior School of Physics and Mathematics, Mexico City, Mexico.

Email: haxaleg@gmail.com

1. Equations solved.

Problem statement.

Consider a cylindrical aluminum bar of length and thickness placed along the shaft. This bar is thermally insulated along its length, but not at its ends. Initially the bar is at a uniform temperature and the ends are in contact with an ice bar at . Heat flows only through the non-insulated ends. Determine how the temperature varies along the bar as a function of time. $L = 1m\omega x T_0 = 100^\circ C 0^\circ C$

The model that describes this phenomenon is the heat equation which is apartial differentiating equation, linear, parabolic is a special case of the diffusion equation, where the diffusion coefficient depends on thermal properties of the material by which the heat is diffused, for the case of a dimension it is written as

$$\frac{\partial T(x,t)}{\partial t} = \mathrm{K}\frac{\partial^2 T(x,t)}{\partial^2 x} \qquad \dots (1)$$

This equation has analytic solutions that can be obtained using differentmethods such as Laplace and Fourier transforms, Green functions, Lie group method, separation of variables, the latter being the most used.

Using the method of separation of variables we propose a solution to equation (1) as the multiple of two functions as follows.

$$T(x,t) = X(x)\text{Ʈ}(\mathrm{t})$$

When replacing you have

$$\frac{\mathrm{d}\text{Ʈ}(\mathrm{t})}{\mathrm{dt}}\frac{1}{\text{Ʈ}(\mathrm{t})} = \mathrm{K}\frac{d^2X(x)}{dt^2}\frac{1}{X(x)} = -\lambda$$

Two ordinary differential equations must be solved.

$$\frac{\mathrm{d}\text{Ʈ}(\mathrm{t})}{\mathrm{dt}} + \text{Ʈ}(\mathrm{t})\frac{\lambda}{\mathrm{K}} = 0 \qquad \ldots (2)$$

$$\frac{d^2X(x)}{dt^2} + X(x)\lambda = 0 \qquad \ldots (3)$$

With the following initial and border conditions .

$$T(x,0\,) = 100°C\,;\;\; T(0\,,t) = T(1\,,t) = 0$$

Making an analysis in the sign of for equation (2) it is verified that for , we have the trivial solution and for leads us to contradictions, therefore gives us the solutions we are looking for, making the change of variable , we obtain the following solution for $\lambda\lambda = 0\lambda < 0\lambda > 0\lambda = \beta^2X(x)$.

$$X(t) = A'e^{-i\beta x} + B'e^{i\beta x} = \mathrm{A}\sin\beta x + \mathrm{B}\cos\beta x$$

Applying conditions at the border.

$$\mathrm{A}\sin\beta = 0 \rightarrow \beta = n\pi; n = 0,1,2,\ldots$$

$$\therefore X(x) = A_n sin(n\pi x)$$

The solution of (2) is as follows.

$$\text{Ʈ}(\mathrm{t}) = B_n e^{-\beta^2\mathrm{K}t} = B_n e^{-n^2\pi^2\mathrm{K}t}$$

Therefore

$$\mathrm{T(x,t)} = C_n sin(n\pi x)e^{-n^2\pi^2\mathrm{K}t}$$

By the principle of overlap, we have that the general solution is.

$$\mathrm{T(x,t)} = \sum_{n=0}^{\infty} C_n sin(n\pi x)e^{-n^2\pi^2\mathrm{K}t}$$

To determine them, we apply the initial condition.C_n

$$\mathrm{T(x,0)} = \sum_{n=0}^{\infty} C_n sin(n\pi x) = 100$$

Thus, it represents the$C_n$coefficients of a series Fourier expansion , so we can calculate them as.

$$C_n = \frac{2}{1}\int_0^1 100\sin(n\pi x)\,dx = \frac{200}{n\pi}(1-(-1)^n)$$

Therefore the solution to our problem is.

$$\mathrm{T(x,0)} = \frac{400}{\pi}\sum_{n=1,3,5,\ldots}^{\infty}\frac{sin(n\pi x)}{n}e^{-n^2\pi^2\mathrm{K}t}$$

With$\mathrm{K} = \frac{k}{C\rho}$

$$\mathrm{T(x,0)} = \sum_{n=1,3,5,\ldots}^{\infty}\frac{400}{n\pi}\,sin(n\pi x)\,e^{-\frac{n^2\pi^2kt}{C\rho}} \qquad \ldots(4)$$

2. **Numerical method or algorithm used.**

There are many numerical methods to solve partial differential equations such as finite differences, finite element, finite volume, we will use finite differences to approximate a solution to the problem.

Let's express the derivatives that appear in (1) using the forward difference approximation.

$$\frac{\partial T(x,t)}{\partial t} \approx \frac{T(x,t+\Delta t) - T(x,t)}{\Delta t}$$

$$\frac{\partial T(x,t)}{\partial x} \approx \frac{T(x+\Delta x,t) - T(x,t)}{\Delta x}$$

Applying the same approximation to obtain the second derivative of $T(x,t)$ we obtain.

$$\frac{\partial^2 T(x,t)}{\partial^2 x} = \frac{T(x+\Delta x,t) + T(x-\Delta x,t) - 2T(x,t)}{(\Delta x)^2}$$

Now we can substitute the expressions of the approximate derivatives in equation (1)

$$\frac{T(x,t+\Delta t) - T(x,t)}{\Delta t} = \mathrm{K}\frac{T(x+\Delta x,t) + T(x-\Delta x,t) - 2T(x,t)}{(\Delta x)^2}$$

As we want to know the values of our function throughout the bar but for a next time interval we clear the term that depends on $.t + \Delta t$

$$T(x,t+\Delta t) = \mathrm{T}(x,t) + \frac{\mathrm{K}\Delta t}{(\Delta x)^2}(T(x+\Delta x,t) + T(x-\Delta x,t) - 2T(x,t))$$

We change the notation to simplify the previous expression.

$$\eta = \frac{\kappa\Delta \mathrm{t}}{(\Delta x)^2}$$

Since the coordinates are now discretized, the variables can only take the following values.

$$x = i\Delta x\,;\;\; t = j\Delta t \therefore T(x,t) = T(i\Delta x, j\Delta t) = T_{ij}$$

Now substituting this notation the equation that gives us the temperature in the next time interval.

$$T_{i,j+1} = T_{ij} + \eta\left(T_{i+1,j} + T_{i-1,j} - 2T_{ij}\right)$$

If we have divided the length of the bar into segments and the time into segments, the initial and boundary conditions in the finite difference method are expressed as.$n_x n_t$

$$T_{0,j} = T_{n_x,j} = 0°C \text{ condiciones de frontera}$$

$$T_{i,0} = 100°C \text{ condicion inicial}$$

To ensure that our numerical solution converges, the Von Neumann-Courant stability condition must be met.

$$\eta = \frac{\kappa \Delta \mathrm{t}}{(\Delta x)^2} = \frac{K}{C\rho} \frac{\Delta t}{(\Delta x)^2}$$

So for a die we will have the following restriction for $\Delta x \Delta t$

$$\Delta t = \frac{\eta}{\kappa}(\Delta x)^2 = \frac{C\rho\eta}{K}(\Delta x)^2$$

Where K is the thermal conductivity, C specific heat and density of the materialρ, the values of these parameters for aluminum are.

$$K = 205\frac{\mathrm{W}}{\mathrm{m} \cdot \mathrm{K}} ;\ C = 880\frac{\mathrm{J}}{\mathrm{Kg} \cdot \mathrm{K}};\ \rho = 2698.4\frac{\mathrm{Kg}}{\mathrm{m}^3} \quad \therefore \quad \eta = 8.4222499528 \times 10^{-5}\frac{\mathrm{m}^2}{\mathrm{s}} \cdot \frac{\Delta \mathrm{t}}{(\Delta x)^2}$$

And the following for wood.

$$K = 0.274\frac{\mathrm{W}}{\mathrm{m} \cdot \mathrm{K}} ;\ C = 2268\frac{\mathrm{J}}{\mathrm{Kg} \cdot \mathrm{K}};\ \rho = 450\frac{\mathrm{Kg}}{\mathrm{m}^3} \quad \therefore \quad \eta = 2.684695277 \times 10^{-7}\frac{\mathrm{m}^2}{\mathrm{s}} \cdot \frac{\Delta \mathrm{t}}{(\Delta x)^2}$$

3. Code listing.

Table 1: Code used for heat EDP resolution

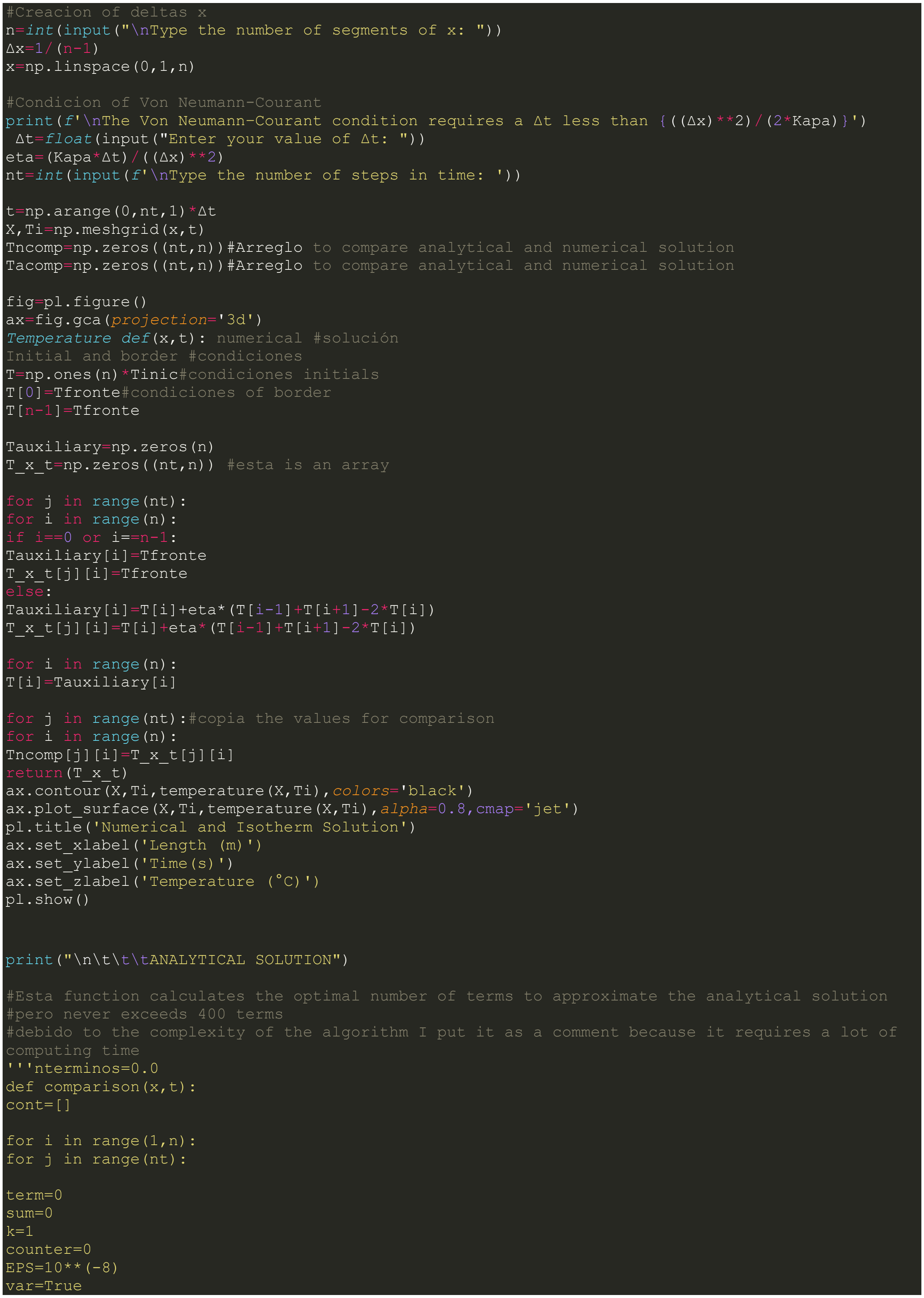
```
import matplotlib.pylab as pl
import numpy as np
from pylab import *
from mpl_toolkits.mplot3d import Axes3D
L=1
#propiedades of wood
#k=0.274
#c=2268
#ro=450
#Propiedades of the aluminum bar

k=205
c=880
RO=2698.4

Kapa=k/(c*ro)

Initial and border #condiciones
Tinic=100
Tfronte=0

print("\t\t\tNUMERICAL SOLUTION")
```



```
#Creacion of deltas x
n=int(input("\nType the number of segments of x: "))
Δx=1/(n-1)
x=np.linspace(0,1,n)

#Condicion of Von Neumann-Courant
print(f'\nThe Von Neumann-Courant condition requires a Δt less than {((Δx)**2)/(2*Kapa)}')
 Δt=float(input("Enter your value of Δt: "))
eta=(Kapa*Δt)/((Δx)**2)
nt=int(input(f'\nType the number of steps in time: '))

t=np.arange(0,nt,1)*Δt
X,Ti=np.meshgrid(x,t)
Tncomp=np.zeros((nt,n))#Arreglo to compare analytical and numerical solution
Tacomp=np.zeros((nt,n))#Arreglo to compare analytical and numerical solution

fig=pl.figure()
ax=fig.gca(projection='3d')
Temperature def(x,t): numerical #solución
Initial and border #condiciones
T=np.ones(n)*Tinic#condiciones initials
T[0]=Tfronte#condiciones of border
T[n-1]=Tfronte

Tauxiliary=np.zeros(n)
T_x_t=np.zeros((nt,n)) #esta is an array

for j in range(nt):
for i in range(n):
if i==0 or i==n-1:
Tauxiliary[i]=Tfronte
T_x_t[j][i]=Tfronte
else:
Tauxiliary[i]=T[i]+eta*(T[i-1]+T[i+1]-2*T[i])
T_x_t[j][i]=T[i]+eta*(T[i-1]+T[i+1]-2*T[i])

for i in range(n):
T[i]=Tauxiliary[i]

for j in range(nt):#copia the values for comparison
for i in range(n):
Tncomp[j][i]=T_x_t[j][i]
return(T_x_t)
ax.contour(X,Ti,temperature(X,Ti),colors='black')
ax.plot_surface(X,Ti,temperature(X,Ti),alpha=0.8,cmap='jet')
pl.title('Numerical and Isotherm Solution')
ax.set_xlabel('Length (m)')
ax.set_ylabel('Time(s)')
ax.set_zlabel('Temperature (°C)')
pl.show()


print("\n\t\t\tANALYTICAL SOLUTION")

#Esta function calculates the optimal number of terms to approximate the analytical solution
#pero never exceeds 400 terms
#debido to the complexity of the algorithm I put it as a comment because it requires a lot of
computing time
'''nterminos=0.0
def comparison(x,t):
cont=[]

for i in range(1,n):
for j in range(nt):

term=0
sum=0
k=1
counter=0
EPS=10**(-8)
var=True
```

```
while var==True:
term=((4*Tinic)/(k*pi))*sin(k*pi*x[i])*exp(-(k**2)*(pi**2)*Kapa*t[j])
sum+=term
k+=2
counter+=1
comp=abs(term/sum)
If comp<eps:
var=False

cont.append(counter)

for i in range(len(cont)):
cont[i]=cont[i]**2

sum=0
for i in range(len(cont)):
sum+=cont[i]

sum=sum/len(cont)
nterminos=int(sqrt(sum))
print(f'\nIt is recommended to take {nterminos} terms of the analytical solution')
comparison(x,t)'''




fig=pl.figure()
ax=fig.gca(projection='3d')
def Graphic-theoretical(x,t):
z=0
p=400#int(input("How many terms do you want to take"))
for i  in  range(1,p*2):#Toma 400 terms of the analytical solution calculated in the previous
function
if i%2==0:
Pass
else:
z+=((4*Tinic)/(i*np.pi))*np.sin(i*np.pi*x)*np.exp(-(i**2)*(np.pi**2)*Kapa*t)

for j in range(nt):#copia values for later comparison
for i in range(n):
Tacomp[j][i]=z[j][i]

return z
ax.plot_surface(X,Ti,Graphatheorical(X,Ti),alpha=0.8,cmap='jet')
pl.title('Analytical Solution')
ax.set_xlabel('Length (m)')
ax.set_ylabel('Time(s)')
ax.set_zlabel('Temperature (°C)')
pl.show()

def compare(Tacomp,Tncomp):#compara numerical and analytical solution at each point of the mesh
List=[]
sum=0
for j in range(nt):
for i in range(n):
if Tacomp[j][i]==0:
error=abs(Tacomp[j][i]-Tncomp[j][i])
List.append(error)
else:
error=abs(Tacomp[j][i]-Tncomp[j][i])
error=error/Tacomp[j][i]
List.append(error)

for i in range(len(List)):
sum=List[i]+sum

average=(sum/len(List))*100
print(f'Percentage error\nE={average}%')
compara(Tacomp,Tncomp)

input()
```

4. Visualization.

If we run the program for the following parameters, with the aluminum data.

$$\text{Número de diviciones en } x = 100$$

$$\therefore \quad \Delta x = \frac{1}{99}$$

$$\text{Tomando en cuenta la condición de estabilidad de Von Neumann} - \text{Courant}, \Delta t = 0.5s$$

$$\text{Número de pasos en el tiempo} = 5000$$

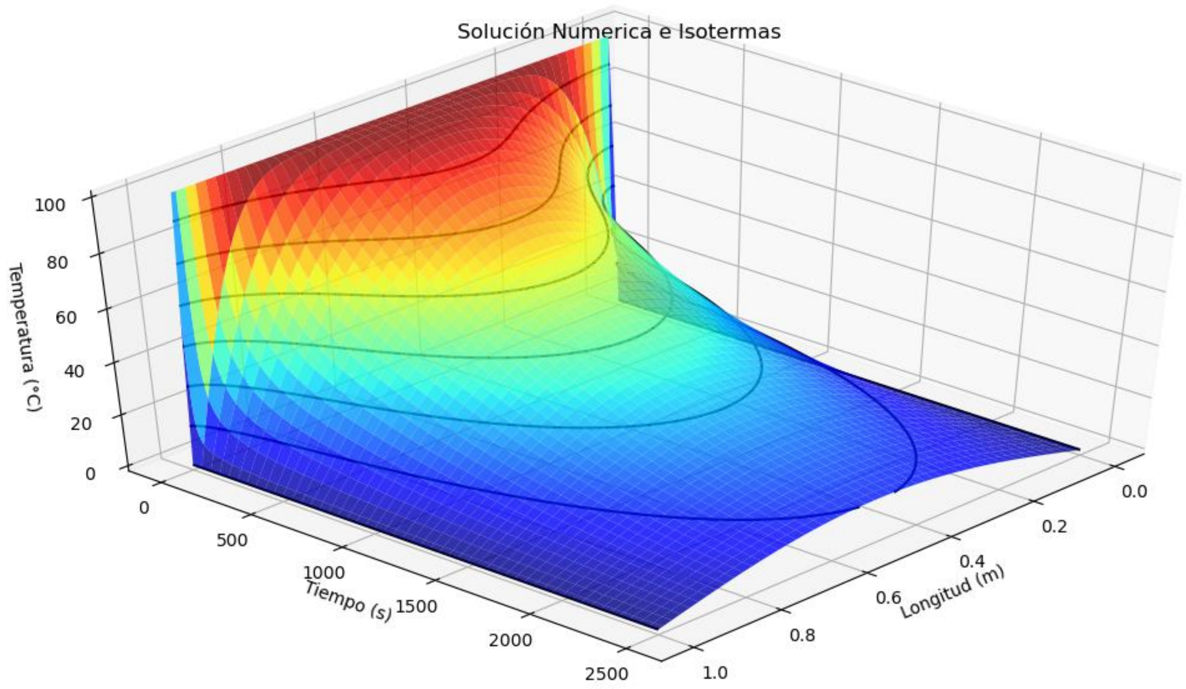


Figure 1: Snumerical solution for aluminum, smoothed for a division of 100 segment length, one and 5000 steps in time.$\Delta t = 0.5s$

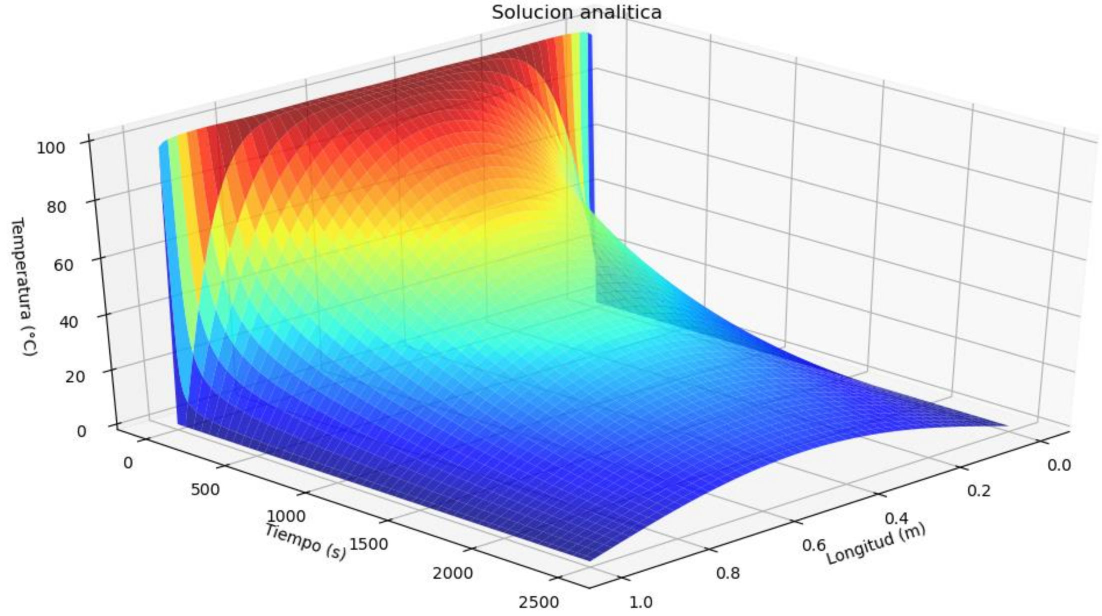


Figure 2: Analytical solution for aluminum, smoothed and cut to 400 series terms, with the same time range

The compare(Tacomp,Tncomp)function calculates the difference in absolute value of the theoretical function value and the value of the numerical function for each mesh value, and averages the value of all errors at each point, to have an overall percentage error.

For the typed conditions we have an error of.

$$E = 1.0765181411525 22\%$$

If we run the program for the following parameters, with the wood data.

$$\text{Número de diviciones en } x = 100$$

$$\therefore \Delta x = \frac{1}{99}$$

$$\text{Tomando en cuanta la condición de estabilidad de Von Neumann} - \text{Courant}, \Delta t = 100s$$

$$\text{Número de pasos en el tiempo} = 500$$

$$E = 1.076518141152522\%$$

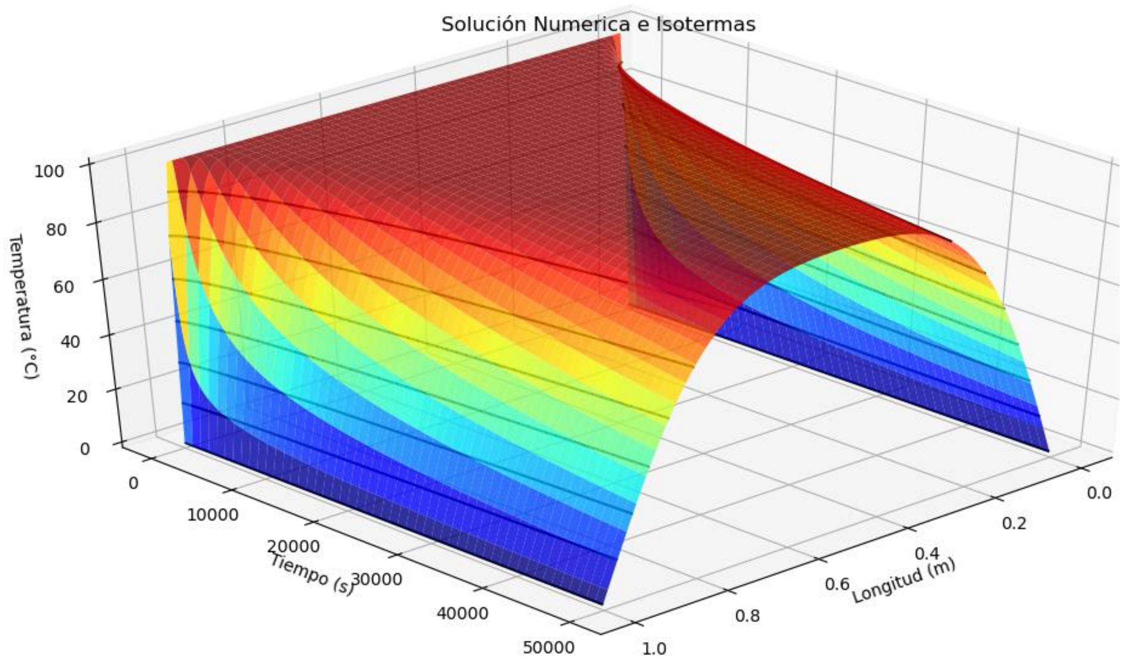


Figure 3:S numerical solution for wood, smoothed for a division of 100 segment in length, a Δt = 100s and 500 steps in time.

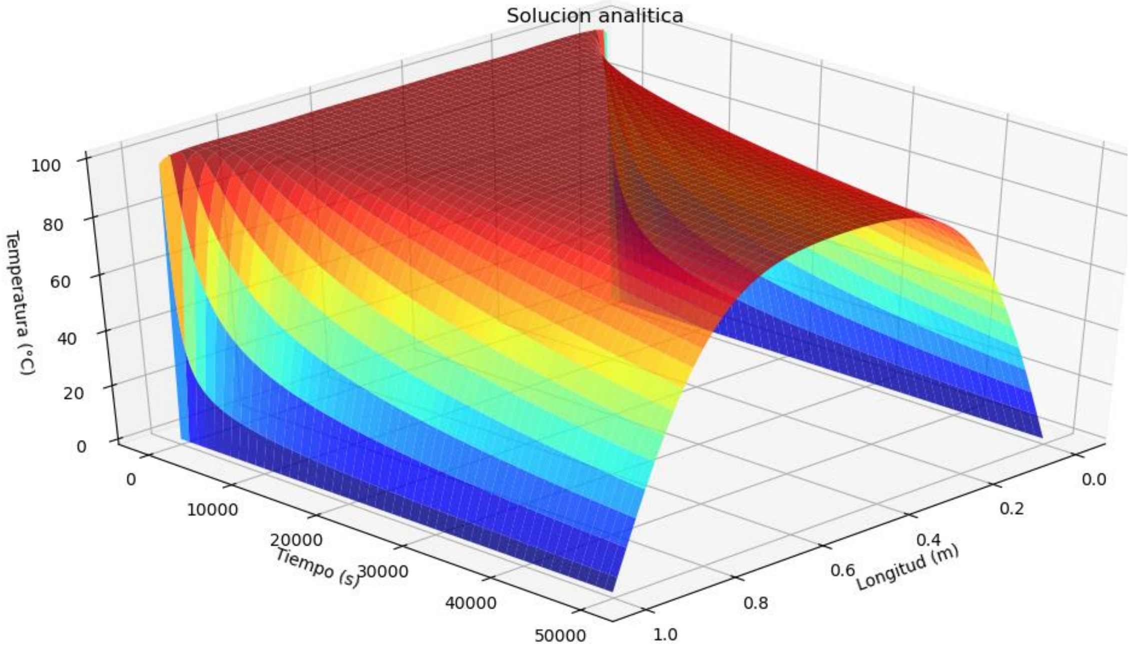


Figure 4:Analytical solution for the mother, smoothed and cut to 400 terms of the series, with the same time range

Let's see what happens when the Von Neumann-Courant stability condition is not met, using the same data shown to obtain graph 1, but the condition tells us to take and only 100 steps in time.$\Delta t < 0.5909282527168707$, $\Delta t = 1$

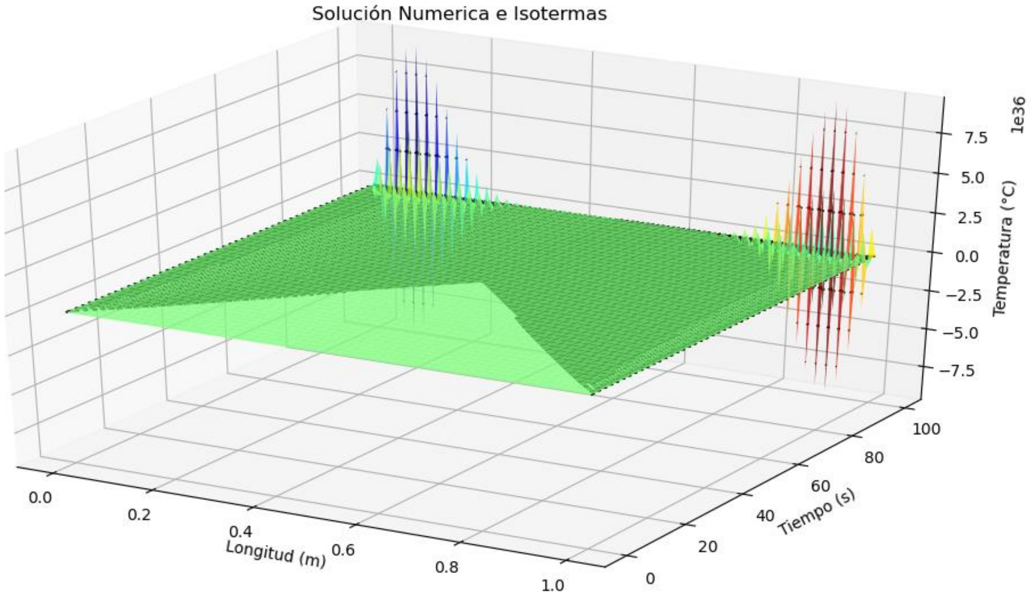


Figure 5: Numerical solution when the Von Neumann-Courant stability condition is not met

5. Discussion.

We see that our numerical solution reaches equilibrium for very large times, andthis was to be expected because the next equation that defines the values of the numerical solution for next instants of time.

$$T_{i,j+1} = T_{ij} + \eta\left(T_{i+1,j} + T_{i-1,j} - 2T_{ij}\right)$$

The factor will -2 decrease the value in each iteration for the next instant of time, so after a very large number of iterations we will find the bar at zero degrees.

To buy with the analytical solution (4) it is necessary to cut the series for a certain level of significance (), so an algorithm similar to10^{-8}l that was used in the zero project was impelled, due to the discretization of time (steps in time) and space (divisions of length) the $n_t n_x$ tolerance level was calculated for each of the $n_x n_t$ points of the mesh and keeping the number of terms necessary to obtain a tolerance level less than , it is clear that each point will need a different number of terms, 10^{-8}because the level of significance is punctual, so in the end a weighting of how many terms was used in each point is made, and that weighting is the number of suggested terms that the analytical solution must have, of course this implies $n_x n_t$ operations, a very high cost of time and memory, so the calculation was made for n_x and n_t much higher than usual ($n_x \approx 200\,, n_t \approx 10{,}000$), using that reference calculation, it was fixed as the number of terms that are tobe marked for the numerical solution; 400 terms.

Once we have defined solutions, $T(x,t)$,we calculate the value of at each point of the discrete mesh with both solutions, and for each point the percentage error is calculated, and all err ores are averaged, some of these errors are shown in the visualization section. On the error, this decreases as a greater number of divisions (n_x) of the length of the bar is taken, that is, we will have low errors for small values of Δx.

The condition of stability is important that it is always fulfilled if we want to have physically acceptable results because, although the calculations never present indeterminacies or mathematical contradictions, the results they throw do not describe reality, we can see in graph 5, that for the first instants of time the temperature begins to oscillate drastically in the between, When the assumption has already been made that it was at a constant temperature and physically it cannot increase, much less to values as large as shown in the graph this shows us that the solution diverges.

The solution obtained is also consistent with what is expected for two materials because for aluminum which is a good conductor of heat andthis reachestemperatures less than or equal to $15°C$ in only 0.69 hours, while wood in the center continues to have a temperature of $99.5°C$ even after 14 hours.

6. Criticizes

The method of finite differences, although it is one of the simplest numerical methods for EDP, is no less accurate the errors we have for small divisions of the bar are very low, that is to say that this method can be used even to solve more complicated equations such as the Navier-Stokes equations or the Einstein field equations [2], The next step to be able to improve the solution, in the sense that best approximates the physical phenomenon is to model a bar that is not isolated adiabatically along and that allows radiation to the outside, then start modeling the equation in 2 and 3 dimensions, of a solid bar, to finally model the most complex case that is the one we pose at the beginning; a cylindrical hollow bar , which gives us an idea of the complexity of PDE to model physical phenomena

7. Bibliography.

1) Rubin H. Landau, Manuel Paez, Cristian Constantin Bordeianu. (2012). Computational Physics Problem Solving with Computers. United States of America: Wiley-VCH.
2) M. Alcubierre. (2005). Introduction to numerical relativity. 25/11/2020, from Mexican Association of Physics Website: http://www.scielo.org.mx/pdf/rmf/v53s2/v53s2a3.pdf