

I.1. Newton's law of cooling

We know that the Newton's law of cooling provides a formula (differential equation) for the dynamics of the temperature $T=T(t)$ of a body immersed in an environment with temperature T_A :

$$\frac{dT}{dt} = -\alpha (T_A - T),$$

where α is called the **cooling rate**, can be solved as $T(t) = T_A + c \exp(-\alpha t)$, where c is a real constant.

- Given T_A , explain what would you measure in order to obtain α . In particular, discuss the physical role of the constant c .
- Imagine that the environment temperature is variable, given by $T_A(t) = 8 + 8 \sin\left(\frac{1}{12}\pi(t-11)\right)$, in Celsius degrees, where t units are hours. Using Maple compute and represent the temperature of a body immersed in this environment along one day ($t \in [0, 24]$) if $T(0) = 37^\circ\text{C}$. For different α values ($\alpha \in \{0.01, 0.03, 0.09\}$, h^{-1}) check and interpret the role of the cooling rate.
- Obtain an exact solution by means of Maple and determine the asymptotic behaviour.

Suggestion: Try also to interpret the data, either from the solution obtained by Maple or the graphics. That is, comment the effect of the parameters on the solutions. For instance, what happens to the rate of convergence to the asymptotic state when we increase α ,...?

We upload the necessary Maple libraries:

```
> restart;  
> with(DEtools):
```

- Given T_A , design an experiment to obtain α .

The equation for the Newton's law of cooling is:

```
> eqdif:=(alpha, TA) -> diff(T(t), t) = alpha*(TA - T(t));
```

$$(\alpha, TA) \rightarrow \frac{d}{dt} T(t) = \alpha (TA - T(t)) \quad (1)$$

We solve it,

```
> dsolve(eqdif(alpha, TA), T(t));
```

$$T(t) = TA + e^{-\alpha t} C1 \quad (2)$$

and we define the solution in terms of the parameters:

```
> SolNewton:=(t, alpha, TA, C1) -> TA + exp(-alpha*t)*C1;
```

$$(t, \alpha, TA, C1) \rightarrow TA + e^{-\alpha t} C1 \quad (3)$$

We realize that, given T_A , we have two unknowns, denoted as α and $C1$, so we need two additional informations to be able to determine them. The initial temperature is a typical datum. If we assume that we know $T(0)=T_0$, then:

```
> C1k:=solve(SolNewton(0, alpha, TA, C1) - T0, C1);
```

$$-TA + T0 \quad (4)$$

Therefore, we have determined one of the two unknowns: $C1 = T_0 - T_A$.

Yet, we have to determine α . One solution (not the unique) would be knowing $T(t[1])=T_1$ for some $t[1]>0$; that is, to register the temperature after some time period. In this case, we insert his data and obtain the value of the remaining unknown α :

> alphak:=solve(SolNewton(t[1],alpha,TA,C1k)-T1,alpha);

$$-\frac{\ln\left(\frac{T1-TA}{-TA+T0}\right)}{t_1} \quad (5)$$

To visualize a specific solution, let us assume the temperature in Celsius degrees, the time in hours, and the known data TA=20, T(0)=36, T(1)=30. We determine the two unknowns in the same Maple cell:

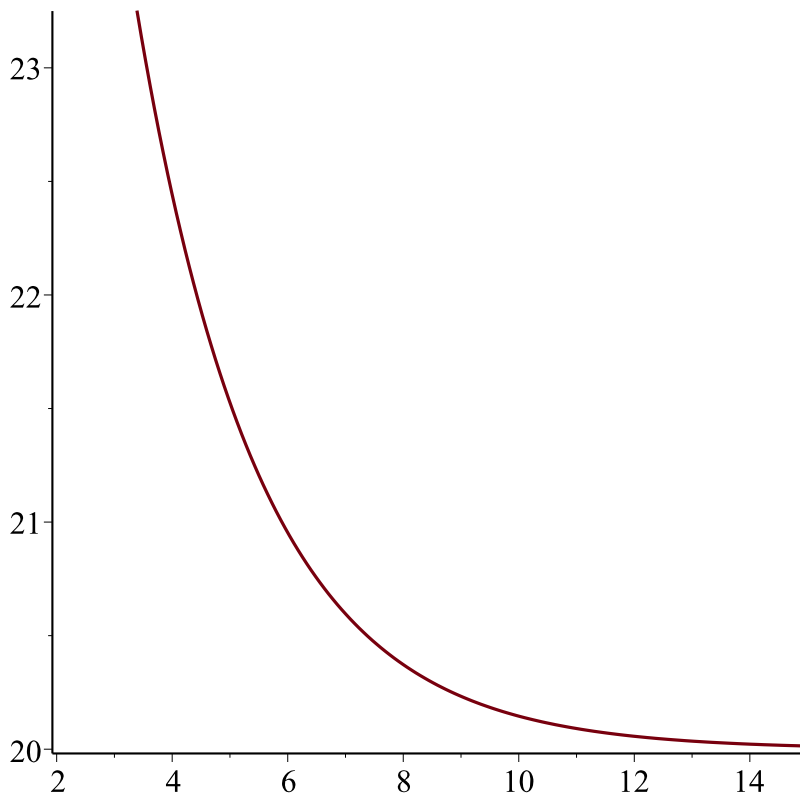
> C1k:=solve(SolNewton(0,alpha,20,C1)-36,C1);

alphak:=solve(SolNewton(1,alpha,20,C1k)-30,alpha);

$$\frac{16}{-\ln\left(\frac{5}{8}\right)} \quad (6)$$

We plot the obtained solution in this case along 15 hours (arbitrary length to see that the temperature approaches 20 Celsius degrees in less than 10 hours):

> plot(SolNewton(t,alphak,20,C1k),t=0..15);



As seen in the previous statement, where $c=C1$, this value can be obtained from the initial condition. In fact, any value of $T(t^*)$ for a specific time t^* would be enough. From the expression $C1=T(0)-TA$, we observe that it corresponds to the amplitude of the decay (if $T0>TA$, or the rise if $T0<TA$) from the initial condition to the asymptotic solution. In other words, the size of the range of temperatures we are going to experience in an experiment with TA constant.

b) Imagine that the environment temperature is variable, given by $T_A(t) = 20 + 4 \sin\left(\frac{1}{12}\pi(t-8)\right)$, in Celsius degrees, where t units are hours. Using Maple, compute and represent the temperature of a body immersed in this environment along one day ($t \in [0, 24]$) if $T(0) = 37^\circ\text{C}$. For different α values ($\alpha \in \{0.01, 0.03, 0.09\}$, h^{-1}) check and interpret the role of the cooling rate.

We define the given ambience fluctuation $TA(t)$ and solve the corresponding differential equation. First we restart and re-charge libraries to avoid overlapping of definitions.

```
> restart;
with(plots):
with(DEtools):
> TA:=t->20+4*sin(Pi*(t-8)/12);
eqdif2:=(alpha)->diff(T(t),t)=alpha*(TA(t)-T(t));
```

$$t \rightarrow 20 + 4 \sin\left(\frac{1}{12} \pi (t - 8)\right) \quad (7)$$

$$\alpha \rightarrow \frac{d}{dt} T(t) = \alpha (TA(t) - T(t))$$

```
> dsolve(eqdif2(alpha),T(t));
```

$$T(t) = e^{-\alpha t} C1 + \frac{4 \left(12 \alpha \pi \cos\left(\frac{1}{12} \pi t + \frac{1}{3} \pi\right) - 144 \alpha^2 \sin\left(\frac{1}{12} \pi t + \frac{1}{3} \pi\right) + 5 \pi^2 + 720 \alpha^2 \right)}{\pi^2 + 144 \alpha^2} \quad (8)$$

We create a function with the explicit solution:

```
> SolNewton2:=(t,alpha,C1)->exp(-alpha*t)*C1+(4*(12*Pi*cos((1/12)*
Pi*t+(1/3)*Pi)*alpha-144*alpha^2*sin((1/12)*Pi*t+(1/3)*Pi)+5*
Pi^2+720*alpha^2))/(Pi^2+144*alpha^2);
```

$$(t, \alpha, C1) \rightarrow e^{-\alpha t} C1 + \frac{4 \left(12 \pi \cos\left(\frac{1}{12} \pi t + \frac{1}{3} \pi\right) \alpha - 144 \alpha^2 \sin\left(\frac{1}{12} \pi t + \frac{1}{3} \pi\right) + 5 \pi^2 + 720 \alpha^2 \right)}{\pi^2 + 144 \alpha^2} \quad (9)$$

As we did in (a), we can impose $T(0)=37$, and determine $C1$:

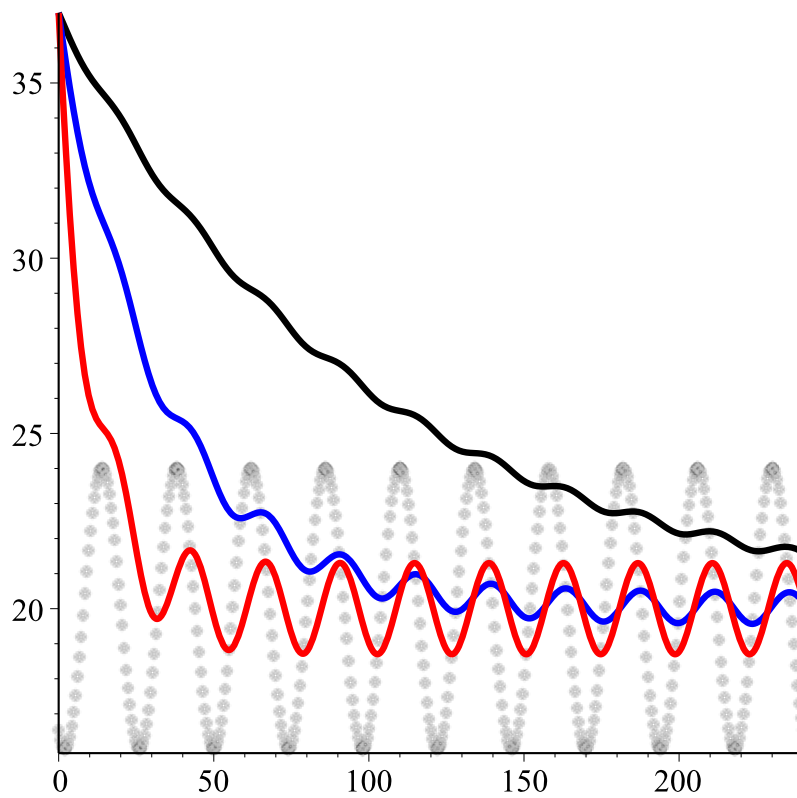
```
> C1k:=alpha->solve(SolNewton2(0,alpha,C1)-37,C1);
alpha->solve(SolNewton2(0,alpha,C1)-37,C1) \quad (10)
```

One can observe the usefulness of the computer algebra system, now. To fulfil statement (b), we define a plot of the solution depending on α :

```
> TotalTime:=240;
dibSolNewton2:=(alpha,c)->plot(SolNewton2(t,alpha,C1k(alpha)),t=
0..TotalTime,color=c,thickness=3);
```

Now, we plot the 3 required solutions, with $\alpha=0.01, 0.03, 0.09$, together with the external input, that is, the solar radiation $TA(t)$:

```
> dibTA:=plot(TA(t),t=0..TotalTime,color=grey,thickness=1,style=
point):
display(dibTA,dibSolNewton2(0.01,black),dibSolNewton2(0.03,
blue),dibSolNewton2(0.09,red));
```



We can observe the effect of α , which is the cooling rate or thermal diffusivity: the bigger α (red curve), the faster its decay towards the asymptotic behaviour and the faster the adaptation to the ambience conditions. It means that the velocity of change in the temperature is bigger and so the object we are modelling, more permeable/vulnerable to ambience conditions.

For the same reason, we also observe that the peaks in the body temperatures for the different α s occur a little bit after the peaks in the solar radiation. Indeed, the lower the α , the longer the separation between peaks.

We also observe a higher fluctuation asymptotic fluctuation for bigger α s. That is, an object with high thermal diffusivity is driven to an oscillatory regime with higher fluctuations. This fact can be quantitatively checked in the next statement.

c) Obtain an exact solution by means of Maple and determine the asymptotic behaviour.

The exact solution is the one called SolNewton2. We compute the limit taking advantage of the Maple command "limit":

> L:=alpha->limit(SolNewton2(t,alpha,C1k(alpha)),t=infinity);

$$\alpha \rightarrow \lim_{t \rightarrow \infty} \text{SolNewton2}(t, \alpha, C1k(\alpha))$$

(11)

```
> evalf(L(0.01));  
evalf(L(0.03));  
evalf(L(0.09));
```

```
19.84160626..20.15839375  
19.49573054..20.50426947  
18.34747436..21.65252564
```

(12)

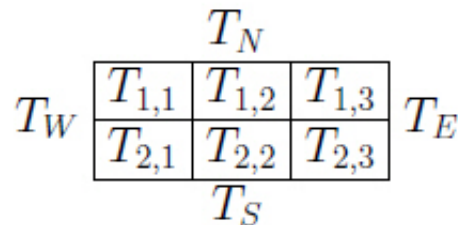
We observe three important facts: (1) the limit does not exist (we receive an interval as solution in each case); (2) the amplitude increases as we increase alpha; (3) the average temperature is always the same and it coincides with the average temperature of the external input.

Activity03-1516Q1. Heat transfer (numerical linear algebra)

We are going to study the heat distribution in a building with some boundary conditions. We assume a rectangular structure of the building, where the North wall is kept at a constant temperature $T_N=20.2$ C, the East and West wall at $T_E=15.6$ C and $T_W=15.6$ C, respectively, and the South wall at $T_S=11.4$ C. We are assuming them constant to simplify the problem, but this is a non-natural hypotheses. Later on, we are going to put more realistic constraints.

The physical principle behind our treatment of the heat distribution inside the building lays on the assumption that, as time goes by, the temperature of each room cell varies according its own temperature and that of the neighbouring cells.

To simplify the problem, and assuming uniform temperature inside each rectangle, we first consider the room divided into 6 rectangles as follows:



We would like to control the temperatures inside the building; that is, $T_{1,1}$, $T_{1,2}$, $T_{1,3}$, $T_{2,1}$, $T_{2,2}$ and $T_{2,3}$. We will solve this problem using basic linear algebra. Our initial conditions are:

$$T^0 = (T_N^0, T_S^0, T_E^0, T_W^0, T_{1,1}^0, T_{1,2}^0, T_{1,3}^0, T_{2,1}^0, T_{2,2}^0, T_{2,3}^0) = (20.2, 11.4, 15.6, 15.6, 18.3, 18.1, 18.4, 18.3, 18.5, 18.2)$$

We suppose that the temperature in each cell after $(n+1)$ hours is obtained by adding the current temperature of the cell itself divided by 2 with that of each of the neighbouring cells divided by 8. For instance:

$$T_{1,1}^{(n+1)} = T_{1,1}^{(n)} / 2 + (T_N + T_{1,2}^{(n)} + T_{2,1}^{(n)} + T_W) / 8,$$

where $T_{i,j}^{(n)}$ represents the current temperature of the cell (i,j) and $T_{i,j}^{(n+1)}$ represents this cell's temperature one hour later. Observe that, for all $n \geq 1$, $T_N^{(n)}$ satisfies the equation $T_N^{(n)} = T_N^{(n-1)}$ and so, $T_N^{(n)} = T_N$ for all n . Analogously, $T_S^{(n)} = T_S$, $T_E^{(n)} = T_E$ and $T_W^{(n)} = T_W$ for all n , since the external temperature is not influenced by that of the room.

We will divide the assignment into three blocks.

BLOCK I

The LinearNumericalAlgebra-Activity.mws contains the corresponding code. You just have to read and understand it, apply it appropriately at each statement and write down the conclusions you get from there.

We answer in this cell only those statements that should be done by hand. Statements (c)-(f) are answered along the way in the Maple computations.

a) Find the equations satisfied by the cells temperatures $T_{1,2}$, $T_{1,3}$, $T_{2,1}$, $T_{2,2}$ and $T_{2,3}$, and rewrite them as a 10×10 linear system with variables $T_N, T_S, T_E, T_W, T_{1,1}, T_{1,2}, T_{1,3}, T_{2,1}, T_{2,2}$ and $T_{2,3}$. Express it in matrix form $T^{(n+1)} = A T^{(n)}$, where $T^{(j)}$ is the vector of temperature at step j .

This statement only required to write down some equations to convince about the construction of the matrix A already provided in the Maple function "MatrizA".

b) Check that T^2 , the temperatures after two hours, can be obtained from $A^2 \cdot \text{VectorTemp}^{(0)}$. Check it with some examples.

This is a manual proof since $T^1 = A T^0$ and $T^2 = A T^1$. Then, $T^2 = A(A T^0) = A^2 T^0$.

c) Describe the heat distribution of the 6 cells during 24 hours. Plot the temperature profile for each of the 6 interior cells.

d) (optional) Consider now the same problem with a $(2r) \times (3r)$ cells' grid. As in statement (e), describe the heat distribution of the $6r^2$ cells during 24 hours, for $r=2$ and $r=3$. Take one cell and compare the temperature profile for $r=1,2,3$.

```
> restart:
```

```
with(LinearAlgebra):
```

```
with(plots):
```

Program to generate the matrix.

```
> MatrizA:=proc(nFil,nCol)
  local Ecuaciones,Incognitas,i,j,NorteDeLaCelda,SurDeLaCelda,
  EsteDeLaCelda,OesteDeLaCelda,NuevaEcuacion,A,b,fM0:
  Ecuaciones:=[TN,TS,TE,T0]:Incognitas:=[TN,TS,TE,T0]:
  # These grouped variables contain the equations and the unknowns
  fM0:=Matrix(1..nFil,1..nCol):
  for i from 1 to nFil do
    for j from 1 to nCol do
      Incognitas := [op(Incognitas),T[i,j]];
      # Next, the temperatures of the cells adjacent to cell (i,j)
      NorteDeLaCelda := `if`(i=1, TN, T[i-1,j]);
      SurDeLaCelda := `if`(i=nFil, TS, T[i+1,j]);
      OesteDeLaCelda := `if`(j=1, T0, T[i,j-1]);
      EsteDeLaCelda := `if`(j=nCol, TE, T[i,j+1]);
      NuevaEcuacion := T[i,j]/2+(NorteDeLaCelda+ SurDeLaCelda +
      OesteDeLaCelda + EsteDeLaCelda)/8;
```

```

    Ecuaciones      := [op(Ecuaciones),NuevaEcuacion];
end do;
end do;
A,b:=GenerateMatrix(Ecuaciones, Incognitas):
A:
end:

```

Inserting the initial data for each cell. Tini1 corresponds to the case of 2x3 interior cells, Tini2 corresponds to the case of 4x6 interior cells, Tini3 corresponds to the case of 6x9 interior cells.

```

> Tini1:=<20.2,11.4,15.6,15.6,18.3,18.1,18.4,18.3,18.5,18.2>;
Tini2:=<20.2,11.4,15.6,15.6,
18.3,18.3,18.1,18.1,18.4,18.4,
18.3,18.3,18.1,18.1,18.4,18.4,
18.3,18.3,18.5,18.5,18.2,18.2,
18.3,18.3,18.5,18.5,18.2,18.2
>;
Tini3:=<20.2,11.4,15.6,15.6,
18.3,18.3,18.3,18.1,18.1,18.1,18.4,18.4,18.4,
18.3,18.3,18.3,18.1,18.1,18.1,18.4,18.4,18.4,
18.3,18.3,18.3,18.1,18.1,18.1,18.4,18.4,18.4,
18.3,18.3,18.3,18.5,18.5,18.5,18.2,18.2,18.2,
18.3,18.3,18.3,18.5,18.5,18.5,18.2,18.2,18.2,
18.3,18.3,18.3,18.5,18.5,18.5,18.2,18.2,18.2
>;

```

$$Tini1 := \begin{bmatrix} 20.2 \\ 11.4 \\ 15.6 \\ 15.6 \\ 18.3 \\ 18.1 \\ 18.4 \\ 18.3 \\ 18.5 \\ 18.2 \end{bmatrix}$$

$$Tini2 := \begin{bmatrix} 1 \dots 28 \text{ Vector}_{column} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran_order} \end{bmatrix}$$

$$Tini3 := \begin{bmatrix} 1 \dots 58 \text{ Vector}_{column} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran_order} \end{bmatrix} \quad (1)$$

a) This is the matrix A for the 2x3 distribution. We also define matrices for the 4x6 and 6x9 distributions (r=2, r=3 in statement (f)).

```
> A1:=MatrizA(2,3);
A2:=MatrizA(4,6):
A3:=MatrizA(6,9):
```

$$A1 := \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{8} & 0 & 0 & \frac{1}{8} & \frac{1}{2} & \frac{1}{8} & 0 & \frac{1}{8} & 0 & 0 \\ \frac{1}{8} & 0 & 0 & 0 & \frac{1}{8} & \frac{1}{2} & \frac{1}{8} & 0 & \frac{1}{8} & 0 \\ \frac{1}{8} & 0 & \frac{1}{8} & 0 & 0 & \frac{1}{8} & \frac{1}{2} & 0 & 0 & \frac{1}{8} \\ 0 & \frac{1}{8} & 0 & \frac{1}{8} & \frac{1}{8} & 0 & 0 & \frac{1}{2} & \frac{1}{8} & 0 \\ 0 & \frac{1}{8} & 0 & 0 & 0 & \frac{1}{8} & 0 & \frac{1}{8} & \frac{1}{2} & \frac{1}{8} \\ 0 & \frac{1}{8} & \frac{1}{8} & 0 & 0 & 0 & \frac{1}{8} & 0 & \frac{1}{8} & \frac{1}{2} \end{bmatrix} \quad (2)$$

b) Check that T^2 , the temperatures after two hours, can be obtained from $A^2 \cdot \text{VectorTemp}^{\{(0)\}}$. Check it with some examples.

It is a trivial fact. We just check that $A(A(T0))=(A^2)(T0)$.

```
> A1.(A1.Tini1)-(A1.A1).Tini1;
```

(3)

$$\begin{bmatrix} 0. \\ 0. \\ 0. \\ 0. \\ 0. \\ 0. \\ 3.55271367880050 \cdot 10^{-15} \\ 0. \\ 0. \\ 3.55271367880050 \cdot 10^{-15} \end{bmatrix}$$

(3)

c) Describe the heat distribution of the 6 cells during 24 hours. Plot the temperature profile for each of the 6 interior cells.

d) (optional) Consider now the same problem with a $(2r) \times (3r)$ cells' grid. As in statement (e), describe the heat distribution of the $6r^2$ cells during 24 hours, for $r=2$ and $r=3$. Take one cell and compare the temperature profile for $r=1,2,3$.

We program altogether the cases $r=1,2,3$. We define the value Nhours to fix the number of hours of simulation.

```
> Nhours:=48:
```

Next cell generates the Nhours simulation for the 3 cases of r:

```
> T1[0]:=Tini1:
   T2[0]:=Tini2:
   T3[0]:=Tini3:
   for j from 1 to Nhours do
     T1[j]:=A1.T1[j-1]:
     T2[j]:=A2.T2[j-1]:
     T3[j]:=A3.T3[j-1]:
   od:
```

Example: the first 5 hours evolution in the 2×3 cells case.

```
> Matrix(<T1[0]|T1[1]|T1[2]|T1[3]|T1[4]>);
```

(4)

20.2	20.20000000000000	20.20000000000000	20.20000000000000	20.20000000000000
11.4	11.40000000000000	11.40000000000000	11.40000000000000	11.40000000000000
15.6	15.60000000000000	15.60000000000000	15.60000000000000	15.60000000000000
15.6	15.60000000000000	15.60000000000000	15.60000000000000	15.60000000000000
18.3	18.17500000000000	18.01250000000000	17.84316406250000	17.6833740234375
18.1	18.47500000000000	18.49843750000000	18.37421875000000	18.2004638671875
18.4	18.21250000000000	18.02656250000000	17.84843750000000	17.6853515625000
18.3	17.12500000000000	16.39687500000000	15.92011718750000	15.5923583984375
18.5	17.50000000000000	16.76093750000000	16.21523437500000	15.8087646484375
18.2	17.08750000000000	16.38281250000000	15.91484375000000	15.5903808593750

(4)

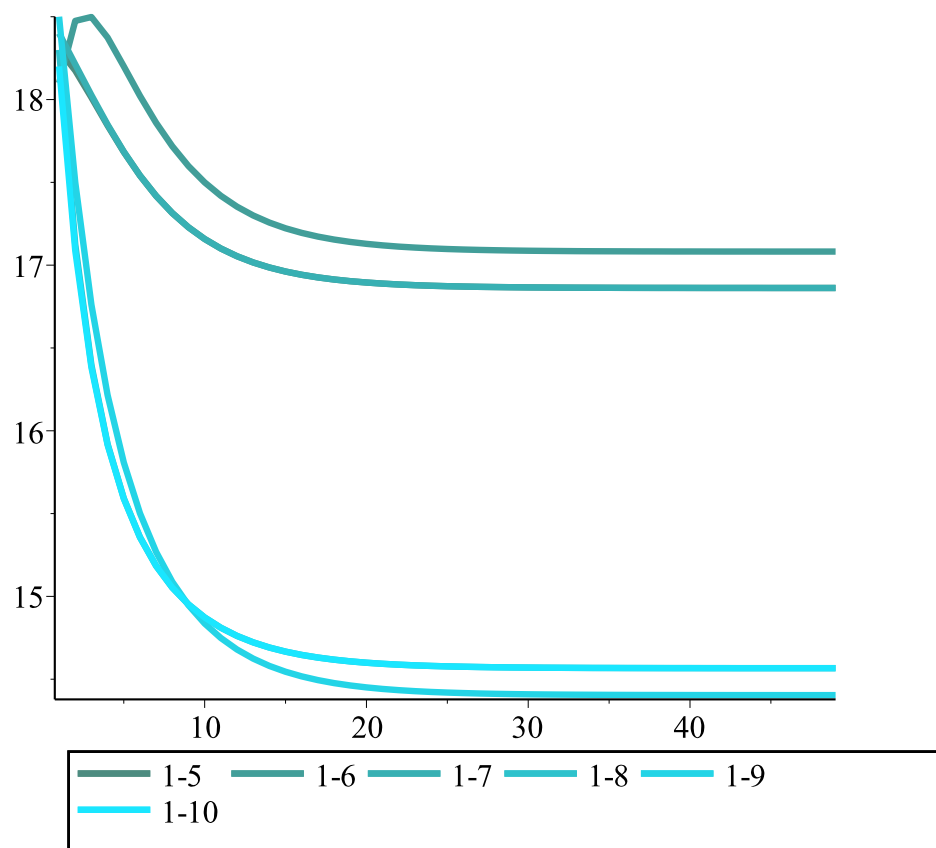
Now, we prepare separate plots of each cell along time. In general, `dibTr[k]` contains the temperatures, hour by hour, of the cell `k` in the case of $(2r) \times (3r)$ cells.

```
> for l from 1 to 10 do
  dibT1[l]:=listplot([seq(T1[k][l],k=0..48)],color=ColorTools:-
  Color("RGB", [0.5-0.4*l/10., 0.2+0.7*l/10., l/10.0]),legend=
  typeset("1-%1",l),thickness=3):
od:
for l from 1 to 28 do
  dibT2[l]:=listplot([seq(T2[k][l],k=0..48)],color=ColorTools:-
  Color("RGB", [0.5+0.4*l/28., 0.2+0.7*l/28., l/28.0]),legend=
  typeset("2-%1",l),thickness=3):
od:
for l from 1 to 58 do
  dibT3[l]:=listplot([seq(T3[k][l],k=0..48)],color=ColorTools:-
  Color("RGB", [0.5+0.4*l/58., 0.2+0.7*l/58., l/58.0]),legend=
  typeset("3-%1",l),thickness=3):
od:
```

Next, we perform some computations related to the demands of statements (e) and (f).

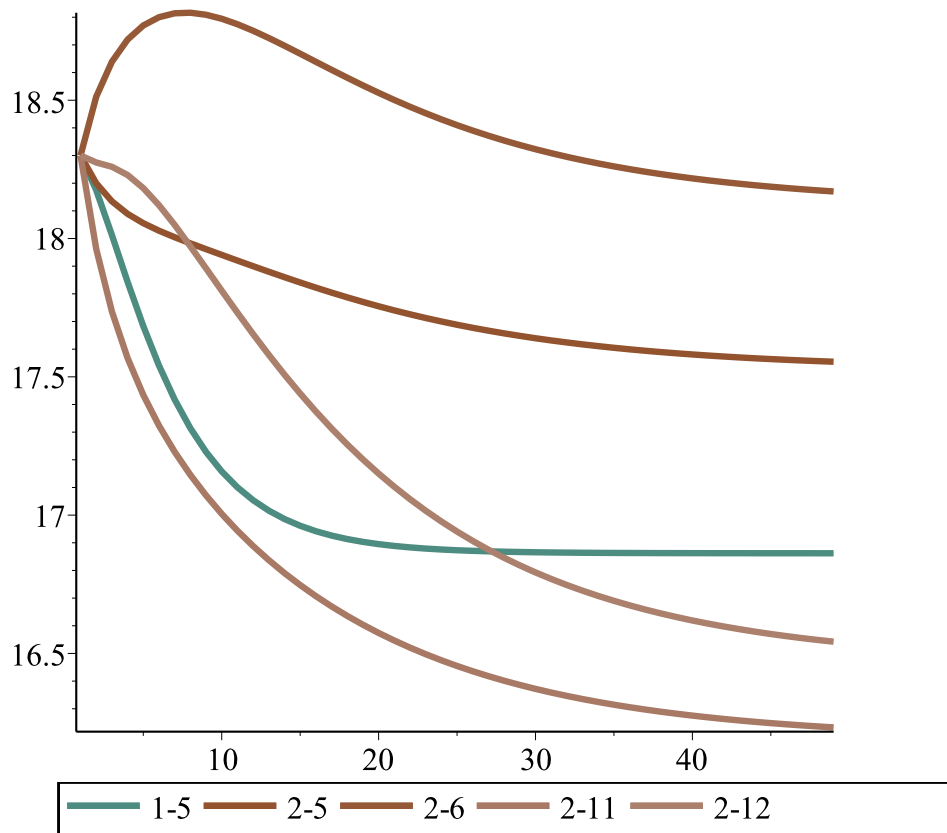
First, the plot of the 6 interior cells in the case $r=1$ (statement (e)). Observe that the south cells (brightest colours) reach a lower temperature due to the low initial condition of T_S .

```
> display(dibT1[5],dibT1[6],dibT1[7],dibT1[8],dibT1[9],dibT1[10]);
```



Now, we compare cell 5 for $r=1$, for the subcells of it considered for $r=2$ (that is, cells 5,6,11,12 for $r=2$). Observe the heterogeneity of the subcells (brownish colours) compared to the "big" cell (blueish colour).

```
> display(dibT1[5],dibT2[5],dibT2[6],dibT2[11],dibT2[12]);
```

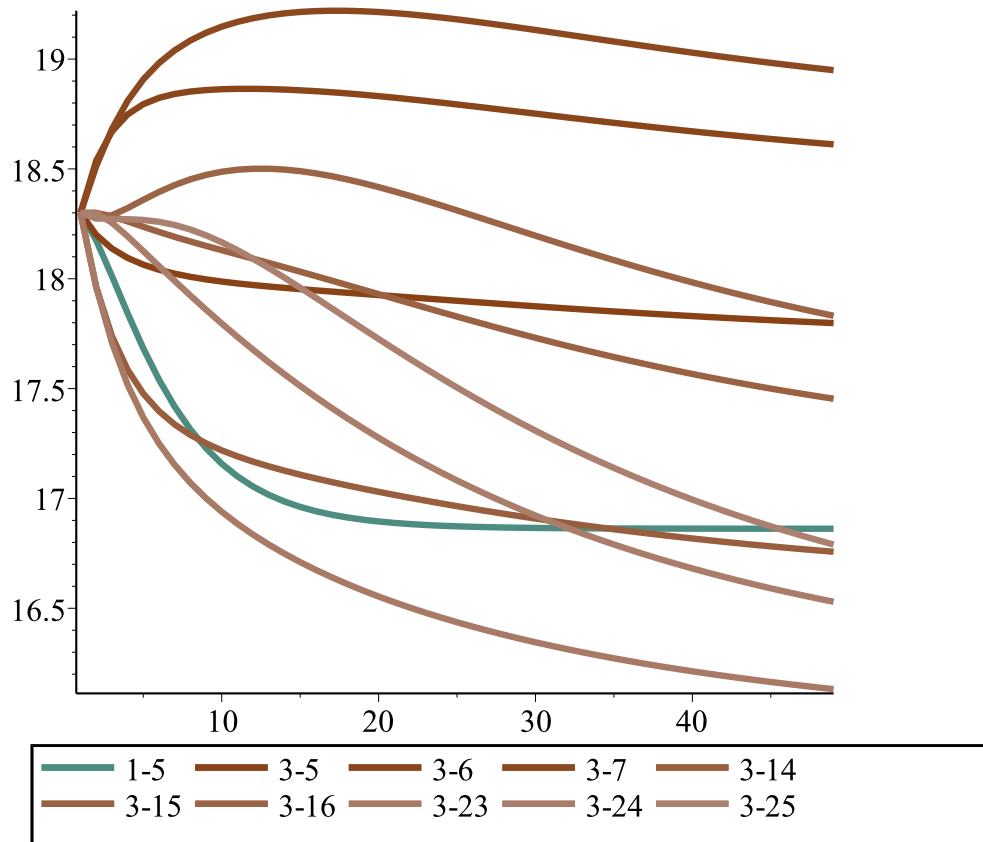


The following commands would compare cells 6,7,8,9 and 10 for $r=1$ with the respective subcells.

```
> display(dibT1[6],dibT2[7],dibT2[8],dibT2[13],dibT2[14]):
display(dibT1[7],dibT2[9],dibT2[10],dibT2[15],dibT2[16]):
display(dibT1[8],dibT2[17],dibT2[18],dibT2[23],dibT2[24]):
display(dibT1[9],dibT2[19],dibT2[20],dibT2[25],dibT2[26]):
display(dibT1[10],dibT2[21],dibT2[22],dibT2[27],dibT2[28]):
```

Comparing cell 5 for $r=1$ with $r=3$:

```
> display(dibT1[5],dibT3[5],dibT3[6],dibT3[7],dibT3[14],dibT3[15],
dibT3[16],dibT3[23],dibT3[24],dibT3[25]);
```



BLOCK II

Let us suppose now that we want to install double glasses in the South walls. There are several ways to implement this circumstance. One of them (you can propose other options though) is adding a new cell neighbouring all the previous South cells, thus incorporating a new variable T_{esp} into the system.

$$\begin{array}{c}
 T_N \\
 \begin{array}{|c|c|c|}
 \hline
 T_{1,1} & T_{1,2} & T_{1,3} \\
 \hline
 T_{2,1} & T_{2,2} & T_{2,3} \\
 \hline
 & T_{\text{esp}} & \\
 \hline
 \end{array} \\
 T_S
 \end{array}
 \begin{array}{c}
 T_W \quad T_E
 \end{array}$$

(a) State the new set of equations and compute the heat distribution after 12 and 24 hours taking $T_{\text{esp}}^0 = 18.4^\circ\text{C}$.

We create the matrix taking into account the new cell provided by the presence of the double glass,

modifying the function MatrizA accordingly. We call Tcristal to the added cell.

```
> MatrizAcrystal:=proc(nFil,nCol)
  local Ecuaciones,Incognitas,i,j,NorteDeLaCelda,SurDeLaCelda,
  EsteDeLaCelda,OesteDeLaCelda,NuevaEcuacion,Ecuacioncristal,A,b,
  fM0:
  Ecuaciones:=[TN,TS,TE,T0]:Incognitas:=[TN,TS,TE,T0]:
  fM0:=Matrix(1..nFil,1..nCol):
  for i from 1 to nFil do
  for j from 1 to nCol do
    Incognitas := [op(Incognitas),T[i,j]];
    NorteDeLaCelda := `if`(i=1, TN, T[i-1,j]);
    SurDeLaCelda := `if`(i=nFil, Tcristal, T[i+1,j]);
    OesteDeLaCelda := `if`(j=1, T0, T[i,j-1]);
    EsteDeLaCelda := `if`(j=nCol, TE, T[i,j+1]);
    NuevaEcuacion := T[i,j]/2+(NorteDeLaCelda+ SurDeLaCelda +
    OesteDeLaCelda + EsteDeLaCelda)/8;
    Ecuaciones := [op(Ecuaciones),NuevaEcuacion];
  end do;
end do;
Ecuacioncristal:=(1/2)*Tcristal+(1/4)*TS:
for j from 1 to nCol do
Ecuacioncristal:=Ecuacioncristal+(1/4/nCol)*T[nFil,j]:
od;
#Ecuacioncristal:=(1/2)*Tcristal+(1/8)*(T[nfil][1]+T[nfil][2]+T
[nfil][3]+TS):
Ecuaciones:=[op(Ecuaciones),Ecuacioncristal]:
Incognitas:=[op(Incognitas),Tcristal]:
A,b:=GenerateMatrix(Ecuaciones, Incognitas):
A:
end:
```

(b) Compare them with those obtained in Block I (e).

For this purpose, we create matrices, vectors and functions similar to those used in Block I.

```
> Acrystal:=MatrizAcrystal(2,3);
```

$$Acrystal := \begin{bmatrix} 11 \times 11 \text{ Matrix} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran_order} \end{bmatrix}$$

(5)

```
> Tinicristal:=<20.0,11.4,15.6,15.6,18.3,18.1,18.4,18.3,18.5,18.2,
18.4>;
```

Tinicristal := $\left[\begin{array}{l} 1 \dots 11 \text{ Vector}_{column} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran_order} \end{array} \right]$

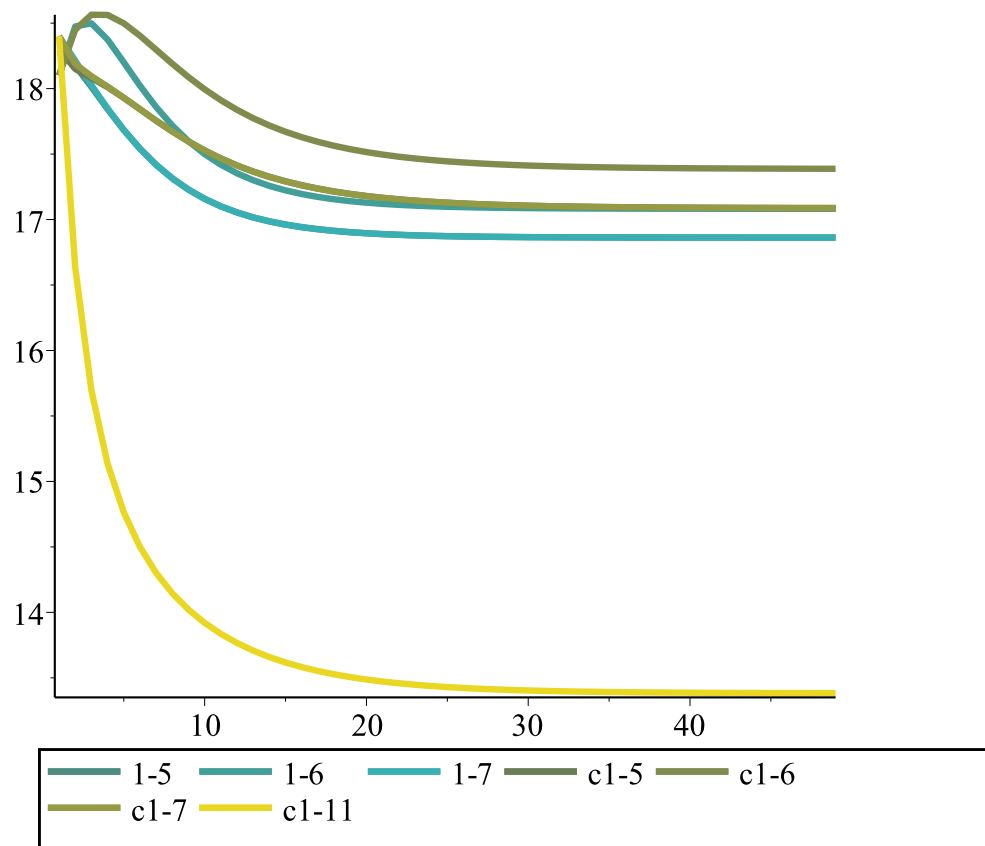
(6)

```
> Tc1[0]:=Tinicristal:
  for j from 1 to Nhours do
    Tc1[j]:=Acristal.Tc1[j-1]:
  od:

  for l from 1 to 11 do
    dibTc1[l]:=
    listplot(
      [seq(Tc1[k][l],k=0..Nhours)],
      color=ColorTools:-Color("RGB", [l/12.0, 0.2+0.7*l/12., 0.5-0.4*
        l/12.]),
      legend=typeset("c1-%1",l),
      thickness=3
    ):
  od:
```

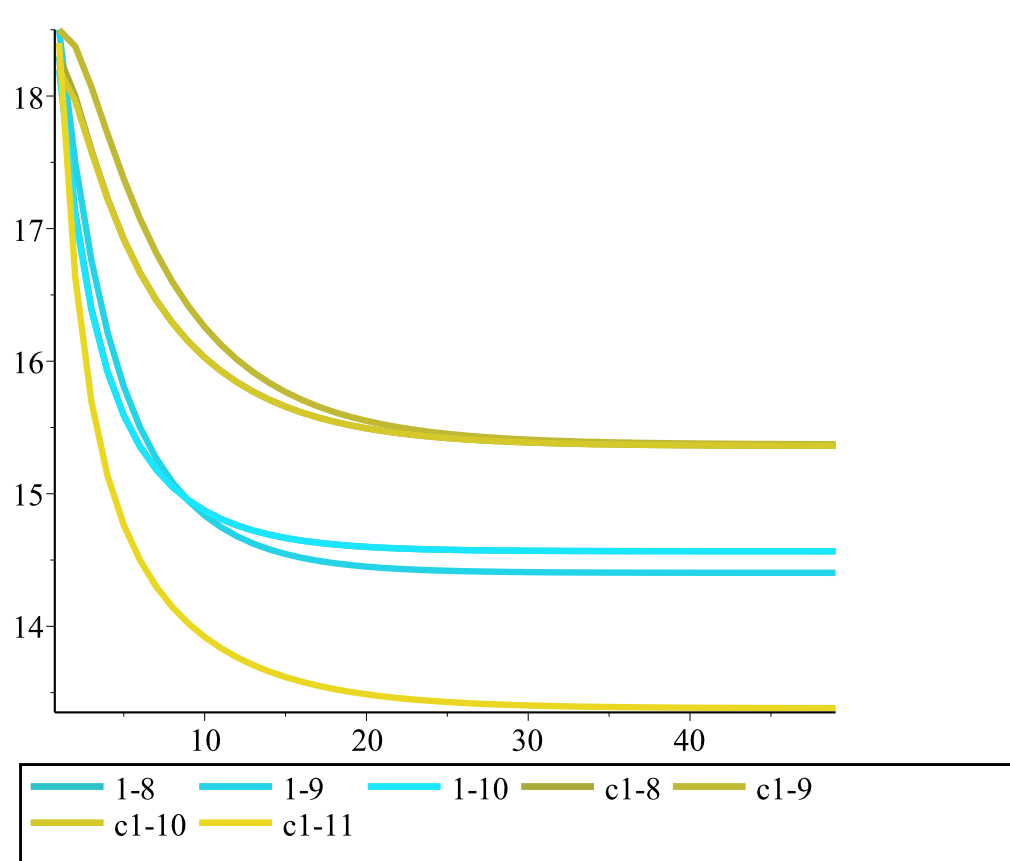
Comparative changes in the north interior cells together with the "cristal cell" (yellowish):

```
> display(dibT1[5],dibT1[6],dibT1[7],dibTc1[5],dibTc1[6],dibTc1[7],
  dibTc1[11]);
```



Comparative changes in the south interior cells together with the "cristal cell" (yellowish). We can appreciate a raise in the temperature in the south cells, meaning that the "cristal cell" is really insulating them.

```
> display(dibT1[8],dibT1[9],dibT1[10],dibTc1[8],dibTc1[9],dibTc1[10],dibTc1[11]);
```



>

BLOCK III

(optional)} Simulate the heat evolution when introducing on the South wall the variable temperature profile used in Activity I.1: $T_A(t) = 8 + 8 \sin(\pi (t-11)/12)$, in Celsius degrees, where t units are hours. For the resolution, see the file LNA-Activity.m and the corresponding output (movie) in the folder of Linear Numerical Algebra

```
> restart:
  with(LinearAlgebra):
  with(plots):
```

Program to generate the tridiagonal matrix required to solve the 1D heat equation using the Implicit Euler Method.

```
> MatTriDiag:=proc(n,r)
  local i,j,M0:
  M0:=Matrix(1..n-1,1..n-1):
  for i from 1 to n-1 do
    for j from 1 to n-1 do
      M0[i,j]:=0:
    end do;
  end do;
  for i from 1 to n-2 do
    M0[i,i]:=1+2*r;
    M0[i,i+1]:=-r;
    M0[i+1,i]:=-r;
  end do;
  M0[n-1,n-1]:=1+2*r:
  M0:
end:

> f:=(x,L)->3*(1+8*x*(x-L)/L^2):
g1:=t->3:
g2:=t->3:
```

We define a function depending on n, m, alpha L and T whose output is the approximate solution of the heat equation with these parameters.

```
> HeatEquation1D:=proc(n,m,alpha,L,T)
  local r,u,b,v,i,j,i1,j1,A;

  r:=alpha*(T/m)*n^2/L^2:#here you should compute your r in terms
  of your assinal thermal diffusivity value
  u:=Matrix(n+1,m+1):
  b:=Vector(1..n-1):
  v:=Vector(1..n-1):

  for j from 1 to m+1 do
    u[1,j]:=evalf(g1((j-1)*T/m)):
    u[n+1,j]:=evalf(g2((j-1)*T/m)):
  od:

  for i from 2 to n do
```

```

u[i,1]:=evalf(f((i-1)*L/n,L)):
od:

A:=MatTriDiag(n,r):

for j1 from 1 to m-1 do

  for i1 from 1 to n-1 do
    b[i1]:=evalf(u[i1+1,j1]):
  od:
  b[1]:=evalf(b[1]+r*u[1,j1+1]):
  b[n-1]:=evalf(b[n-1]+r*u[n+1,j1+1]):

  v:=LinearSolve(A,b):

  for i1 from 1 to n-1 do
    u[i1+1,j1+1]:=v[i1]:
  od:

od:
evalm(u):
end:

```

We compute 3 solutions ($n=10,100,1000$) and $m=100$ and store in 3 components of a vector called uSol (each component of uSol contains a complete $(n+1) \times (m+1)$ u-matrix):

```

> uSol[10]:=HeatEquation1D(10,100,0.1,2,24):
  uSol[100]:=HeatEquation1D(100,100,0.1,2,24):
  uSol[1000]:=HeatEquation1D(1000,100,0.1,2,24):

```

Next, we compute the maximum differences between uSol[10] and uSol[100] after m_0 time-steps.

```

> max10_100:=0:
  m0:=100:
  for j from 2 to 10 do
    mm:=abs(uSol[10][j,m0]-uSol[100][10*(j-1)+1,m0]):
    if (mm>max10_100) then
      max10_100:=mm:
    fi;
  od:
  max10_100;

```

0.000955905083464881

(1)

Next, we compute the maximum differences between uSol[100] and uSol[1000] after m_0 time-steps.

```

> max100_1000:=0:
  for j from 2 to 100 do

```

```

mm:=abs(uSol[100][j,m0]-uSol[1000][10*(j-1)+1,m0]):
if (mm>max100_1000) then
max100_1000:=mm:
fi;
od:
max100_1000;

```

0.00000938110654624680

(2)

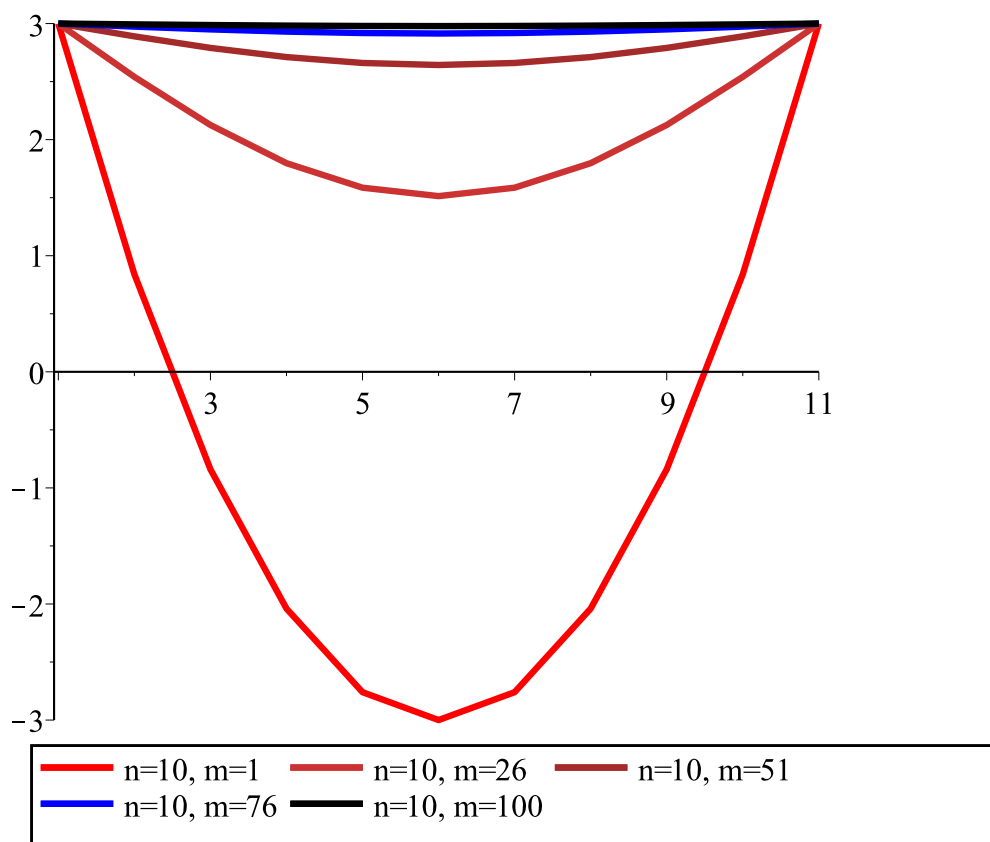
We proceed to represent the results graphically.

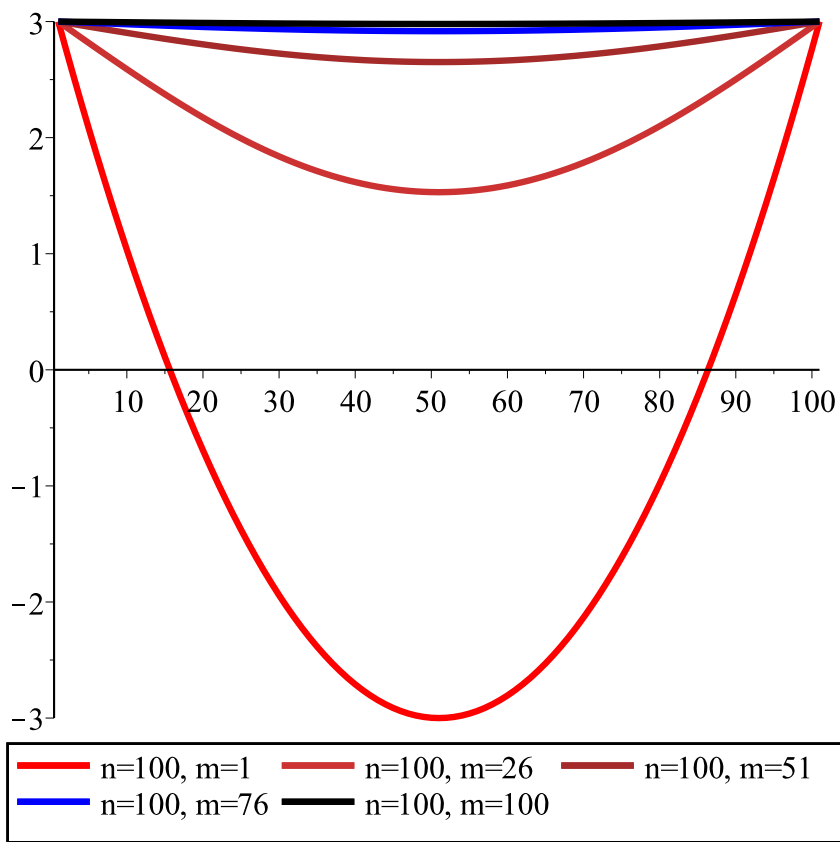
We first plot the temperature distribution at 5 different times for each n.

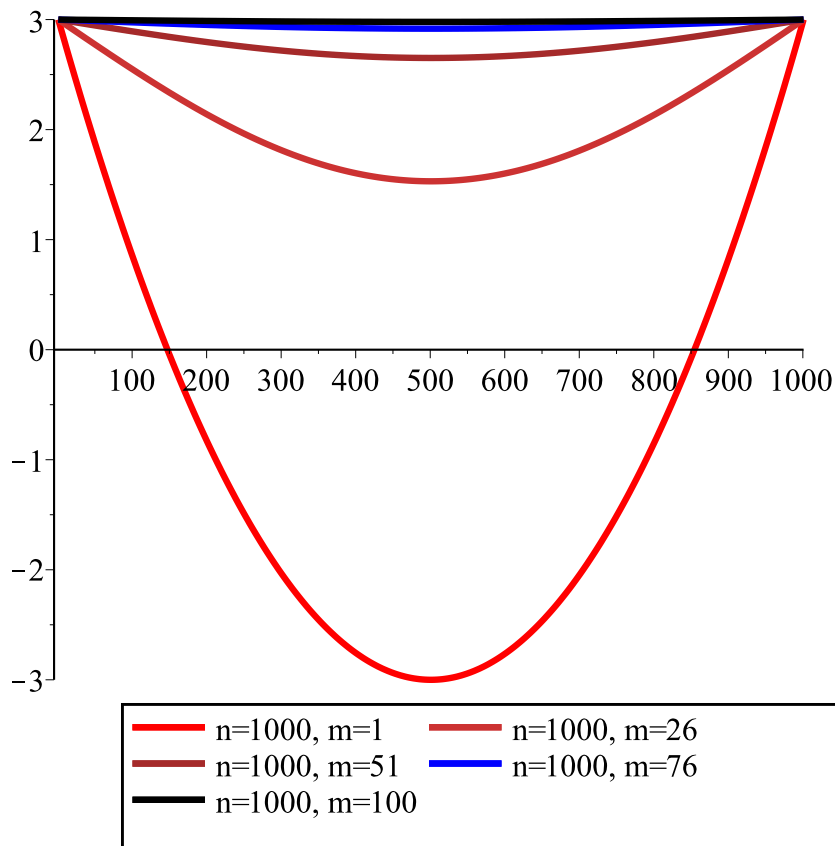
```

> dibU:=(n,j,c)->listplot([seq(uSol[n][i,j], i=1..n+1)],thickness=
3,color=c,legend=typeset("n=%1, m=%2",n,j));
display(dibU(10,1,red),dibU(10,26,orange),dibU(10,51,brown),dibU
(10,76,blue),dibU(10,100,black));
display(dibU(100,1,red),dibU(100,26,orange),dibU(100,51,brown),
dibU(100,76,blue),dibU(100,100,black));
display(dibU(1000,1,red),dibU(1000,26,orange),dibU(1000,51,
brown),dibU(1000,76,blue),dibU(1000,100,black));
dibU := (n,j,c) → plots:-listplot([seq(uSoln,i,j, i = 1 .. n + 1)], thickness = 3, color = c, legend
= typeset("n=%1, m=%2", n, j) )

```

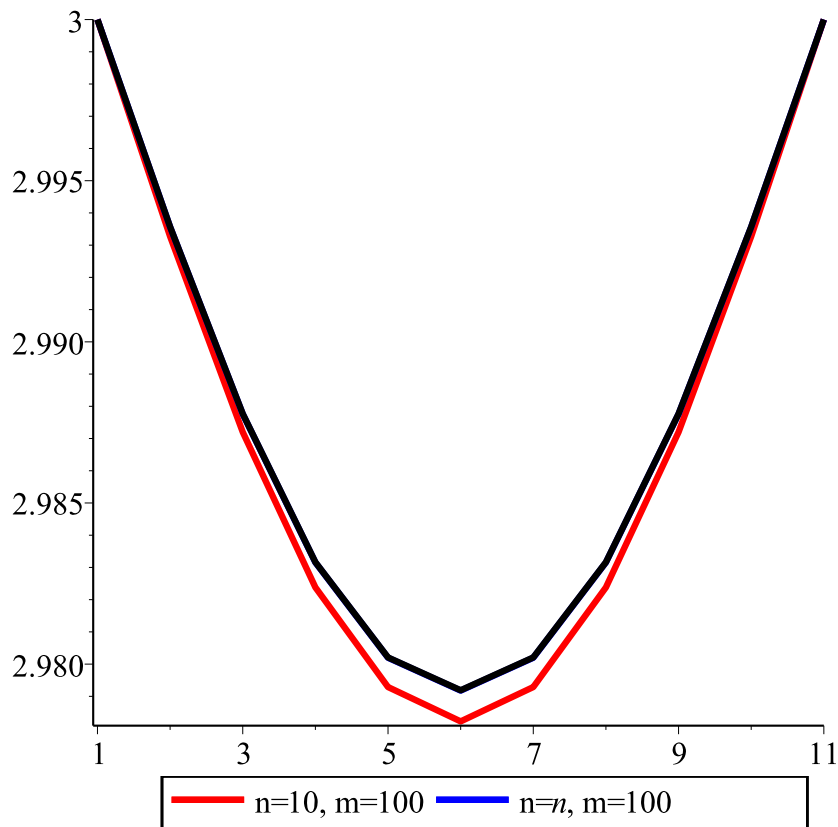






Now, we want to compare the 3 simulations after m_0 time-steps to visualize the differences that we have already computed. To overcome the problem of the different number of nodes (we cannot merge the three corresponding "dibU" plots), we define a function that plot the solution for $n = n_{\text{Base}} \cdot \text{factorScale}$ with the same number of nodes than the solution for n_{Base} . The resulting plot shows the three final temperature distributions.

```
> dibUScale:=(nBase,factorScale,j,c)->listplot([seq(uSol[nBase*  
factorScale][factorScale*(i-1)+1,j], i=1..nBase+1)],thickness=3,  
color=c,legend=typeset("n=%1, m=%2",n,j)):  
  
display(dibU(10,100,red),dibUScale(10,10,100,blue),dibUScale(10,  
100,100,black));
```



Finally, we plot whole simulation for each pair (n,m) using the listplot3d command. You can change n to visualize the 3 solutions.

```
> n:=10:
m:=100:
listplot3d([seq([seq(uSol[n][i,j],i=1..n+1)],j=1..m)],labels=["j
(time index)","i (space index)","u (temperature)"],
labeldirections=[horizontal, horizontal, vertical],axes=frame,
tickmarks=[[1="0",m="T"],[1="0",n+1="L"],default],title="Heat
transfer on a rod");
```

Heat transfer on a rod

