Continuum Physics II
Examination Assignment

Finite difference simulation of the baking process

Andrew N. Jackson - 5th May 1996

Introduction

The brief for this assignment can be outlined as follows. I am to put myself in the position of having been employed by a firm of commercial bakers to investigate the factors underlying the successful baking of cakes. For a simplified cake (ie having made certain assumptions about it), I will consider the factors which influence the baking time and final condition of the cake, and using this information go on to sketch out a plan for a simulation model. From this model I can then devise an appropriate flow diagram for the actual code structure.

Contents

Baking Theory

The cake-baking process can be broken down into three main stages:

1 - Rising

As the temperature of the cake mixture rises, the gas cells (beaten in during the preparation of the cake-mix) expand and the chemical rising agent, if any, releases carbon dioxide (eg baking powder). In home-baked cakes, the uneven heating of the *liquid* cake-mix can cause considerable convection of the material, up until it begins to set. However, this does not occur in commercial cake mixes, as added vegetable gums increase the batter viscosity and so prevent differential movement.

2 - Setting

As the temperature increases further, the cake-mix is set into it's permanent shape by starch gelatinisation, and flour, egg, and milk protein coagulation (at 60°C and 71°C respectively). Basically, the starch and protein molecules line up along the walls of gas cells and then, as the protein chains form (giving structure), water is squeezed out from between the proteins and absorbed by the starch granules (which can absorb up to ten times their own weight of water). The now softened starch granules will become more solid as the cake is allowed to cool after baking.

3 - Browning

Once the mixture has solidified, the flavour-enhancing browning reactions can begin to take place. Browning only occurs significantly in the areas where moisture has been driven from the cake, in effect, where the cake temperature is at least 100°C (see method/further-assumptions section, page 5), and so browning first occurs in the sides and surface of the cake. The products of the browning process diffuse inwards, improving the flavour of the cake, but if left too long the browning process will become charring and burning.

Our aim is to find the oven temperatures and baking times that will lead to a well baked cake. Too low a temperature and the batter solidifies too slowly, allowing the gas cells to expand too far and begin to coalesce, producing a heavy and rough textured cake. At too high a temperature, the outside may well char and burn before the whole of the mixture has set, leading to a 'wet-spot' in the cake.

The Model: Assumptions From The Theory

In this section I will outline the assumptions and simplifications that can be made in order to form a reasonable finite difference simulation of the baking process.

1 - The Oven:

In any commercial operation, large ovens must be used in order to bake the number of products required, and so we can assume that we are using a very wide and good quality oven. This means that while there will be some vertical temperature gradient, there will be a negligible degree of horizontal variation, and so all the isotherms within the oven environment will be horizontal (see fig. 1). This means we can assume that the upper surface of the cake and the lower surface of the tin to each be exposed to a fixed temperature throughout the baking process..

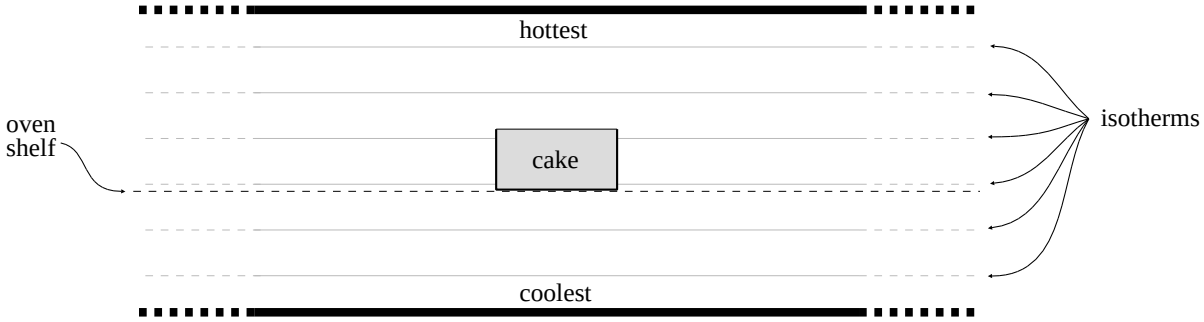

Figure 1: The large oven approximation (isotherms assume linear temperature variation).

If the form of the temperature variation with vertical position is known, then the temperature gradient along the sides of the tin can also be specified. The simplest model for this would be a linear temperature variation constructed from the temperatures at the upper and lower oven surfaces, and I would expect such a model to perform reasonably well.

2 - The Baking Tin:

A number of assumptions are made about the baking tin. Firstly that it is cylindrical, for reasons explained later, and also that it is thin and made from dull, rough metal (or glass). This means that we can assume that the heat of the oven is transmitted instantaneously to the cake-mix, and so the model behaves like a cylindrical arrangement of cake mixture that has no baking tin at all, but still manages to preserve it's cylindrical form. The specification of the material the cake tin is made of is required in order for possible comparison with experimental results, as such tins can require as much as 20% less baking time than a shiny surfaced tin.

3 - The Cake Mixture:

The primary assumptions concerning the cake-mix are that it is homogenous and sufficiently viscous for there to be a negligible degree of heat transportation by convection. Assumptions are also made concerning the chemical behaviour outlined in the theory.

Firstly we suppose that we are given a chemical pathway for each of the rising, setting and browning stages, and that we are also given a function or look-up table that tells us how the rate the cake-mix progresses along this path varies as a function of other system parameters (see fig. 2)

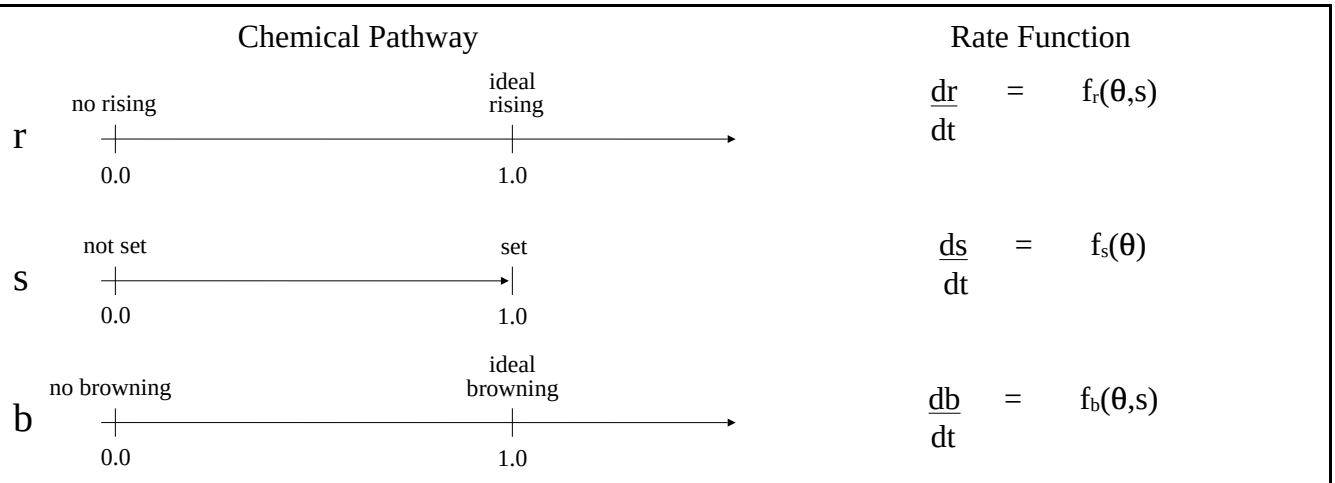

Figure 2: General outline of the chemical pathways and their rate functions.

In other words three variables are use to represent the degree of rising, setting and browning of a part of the cake, and are given (for example) the symbols r, s and b. Each of these has a rate function which varies with temperature (θ), the exact form of which would be determined experimentally (this would take into account the way the starch and protein solidification occurs at 60°C and 71°C, and how the browning rate increases dramatically once the temperature goes past 100°C). As well as this, we must take into account the fact that setting restricts rising, and that browning only occurs in cake-mix that has already set. This means that the rate functions for r and b are functions of both θ and s, where dr/dt becomes zero and db/dt becomes non-zero as s reaches 1.0. This approach also assumes that the values of r and b which correspond to a cake that has risen too far and charred instead of browned have been determined and can be compared with our simulation results. A further assumption is that s cannot be greater than 1.0, ie that there is no such thing as a cake which has set too much.

At first I thought that a single chemical pathway and rate function would be sufficient, but interactions between the three processes means that this would overlook certain possibilities. Such a scheme would assume that a cake always rises to the ideal amount before it begins to set, and could allow significant browning to occur below 100°C, both of which could lead to misleading results.

Further assumptions are that:

- The degree of expansion during the rising stage is assumed to be negligible in comparison with the initial volume of the cake. This means that, within the model, the rising stage will translate into a texture forming stage and no physical expansion will be simulated.

- No significant overall water diffusion occurs while it is in the liquid phase (below 100°C), and the way the cake dries out should be accounted for by the rate function for the browning process. While a *model* for simulation of the diffusion of water through the cake is no more complex than the heat diffusion model, I suggest that it is wise not to include a water-diffusion simulation into the cake model. The main reason for this is that the experimental facts required to back up such a simulation would be difficult to acquire; for example, we would need to know the volume of *free* water per unit volume of cake-mix, which is not only a function of the degree of solidification, s, (because solidification absorbs/ties down water molecules, thus reducing the amount of water that is able to diffuse) but also requires us to know how much water was in the initial cake-mix, where no water is added as a specific, lone ingredient (ie we need the water content of milk, eggs, butter etc). Of course, we would also need to know the diffusivity of water molecules in cake-mix (a function of some or all of θ, r, s and b). Given that the water has to be vaporised before significant browning occurs, I feel that this process is modelled to reasonable accuracy by the rate function approach outlined above.
- The specific thermal capacity per unit volume and the thermal conductivity of the cake-mix are known as functions of the state variables of the system/environment. ie we have $c(\theta,r,s,b)$ and $\kappa(\theta,r,s,b)$. These characteristics may not vary significantly with some or indeed all of θ, r, s and b, but I would leave that for whoever is conducting the experiments to discover, as the information I have to hand sheds no light on this matter. As I will outline in the next section, the thermal capacity and conductivity can be combined to form the thermal diffusivity, ie a single function can be used to represent both these physical characteristics (given the symbol $\chi(\theta,r,s,b)$ in this assignment).

4 - Symmetry:

The above assumptions of tin shape and homogenous cake-mix allow us to break down the three-dimensional nature of the model down to two-dimensions. Instead of modelling the whole of the cake we can ignore the angular variable and use the model shown in fig.3 below.

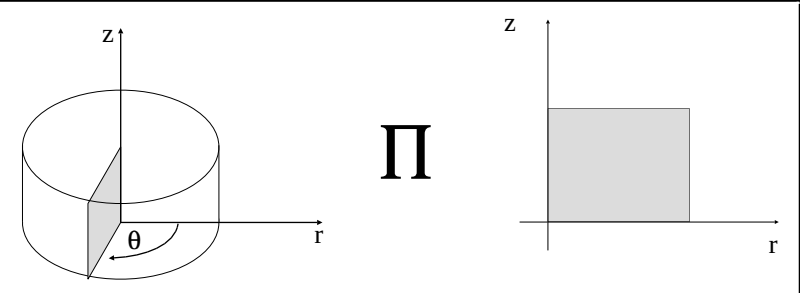

Figure 3: Simplification of the model from 3D to 2D via symmetry.

The Model: Finite Difference Formulation

It is possible to set up the difference equations either by simple differencing or by the controlled volume approach, each give the same result. In this situation of straightforward diffusion through a cartesian grid (fig.2), the simple differencing approach gives us our result easily enough. In order to derive a suitable finite-difference form for a problem, we need to find a mathematical expression which encapsulates the physics of the problem. For the general diffusion case,such as our heat diffusion problem, we use the following parabolic differential equation:

$$c(e)\frac{de}{dt} = \left(k(e) \quad e\right) \qquad \text{- (1)}$$

Or, in integral form:

$$\frac{d}{dt}\int_V c(e)e\,dV - \int_S k(e) \quad e.dS = 0 \qquad \text{- (2)}$$

In other words, the rate of change of the total quantity of ε inside the volume V is balanced by the flow of ε through the surface of that volume. Of course, in our simulation ε will correspond to the cake temperature.This form should maintain positivity, ie no negative temperatures should form from a positive initial state (ε(r)>0 for all r), unless they are due to poor simulation behaviour.

If we use a grid like the one shown in fig.4, and replace the capacity (c) and the conduction (κ) with the diffusivity (χ), then we can break eqn(1) down into the following two dimensional form:

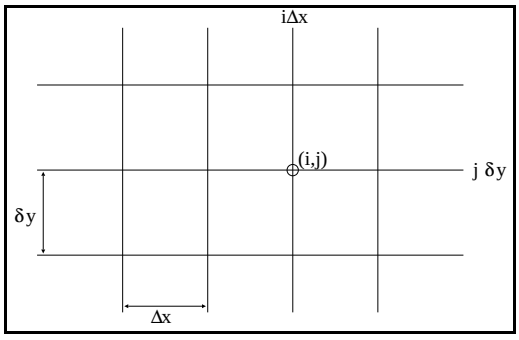

Figure 4: General form of a cartesian finite-difference grid.

$$\frac{de}{dt} = c\left[\frac{d^2e}{dx^2} + \frac{d^2e}{dy^2}\right] \qquad \text{- (3)}$$

Now, by simple differencing of this equation we get:

$$\frac{e_{i,j}^{n+1} - e_{i,j}^{n}}{Dt} = c(\tilde{e})\left(\frac{\tilde{e}_{i+1,j} - 2\tilde{e}_{i,j} + \tilde{e}_{i-1,j}}{Dx^2} + \frac{\tilde{e}_{i,j+1} - 2\tilde{e}_{i,j} + \tilde{e}_{i,j-1}}{dy^2}\right) \qquad \text{- (4)}$$

Where, $\tilde{e}_{i,j} = q e_{i,j}^{n+1} + (1-q)e_{i,j}^{n}$

with Dx and dy as the spatial finite difference steps (as fig.4), and with Dt as the finite difference time step.

This formulation allows us to decide at what time we calculate the spatial derivatives (fluxes) via the implicitness parameter, θ:

θ	=	0	:	Explicit	[1st order]
θ	=	1	:	Fully implicit	[1st order]
θ	=	½	:	Centred time or Crank-Nicholson form	[2nd order]

Equation 4 can now be expanded from the `ε terms into a set of separable $\varepsilon^n$ and $\varepsilon^{n+1}$ terms. However, there is a slight complication in that $\chi$ is a function of `ε, and, once expanded, does not present itself in a separable form. This means that for the purposes of derivation we ignore it's properties as a function of `ε and just treat it as a known constant (ie we use $\chi(e^n_{i,j})$, it's value at the start of the time step of n to n+1). Once the $e^n_{i,j}$ and $e^{n+1}_{i,j}$ parameters have been separated, equation 4 becomes:

$$e^{n+1}_{i,j} - qcDt\left(\frac{e^{n+1}_{i+1,j}-2e^{n+1}_{i,j}+e^{n+1}_{i-1,j}}{Dx^2}+\frac{e^{n+1}_{i,j+1}-2e^{n+1}_{i,j}+e^{n+1}_{i,j-1}}{dy^2}\right) = e^n_{i,j} + (1-q)cDt\left(\frac{e^n_{i+1,j}-2e^n_{i,j}+e^n_{i-1,j}}{Dx^2}+\frac{e^n_{i,j+1}-2e^n_{i,j}+e^n_{i,j-1}}{dy^2}\right) \quad \text{- (5)}$$

For the diffusion equation, the Crank-Nicholson method (at θ = ½) is known to be stable (even when it is not accurate) for all values of Dt. For this reason I would suggest using the C-N method for this problem. Equation 5 can now be expressed in the following general form:

$$- A_{i,j}\,\varepsilon_{i,j+1} - B_{i,j}\,\varepsilon_{i+1,j} - C_{i,j}\,\varepsilon_{i,j} - D_{i,j}\,\varepsilon_{i-1,j} - E_{i,j}\,\varepsilon_{i,j-1} = F_{i,j} \quad \text{- (6)}$$

As the data parameters for all the values of i and j are assembled (ie all i from 0 to I and all j from 0 to J), we can simplify the solution of this matrix problem by breaking the two-dimensional array $\varepsilon_{i,j}$ down to form the one dimensional array E_k, where k = i + (j - 1)I. This means that equation 6 reduces to the form:

$$M_{k,l}\,E_l = F_k \quad \text{- (7)}$$

Where $M_{k,l}$ is a well defined sparse matrix consisting of five diagonal lines corresponding to each of A, B, C, D and E in equation 6. While holding θ = 0 or θ = 1 would make solution of the problem easier, both introduce limits on the time-step we can use and so we need to use one of the iterative matrix solution methods. For our problem it would probably be best to use one of the family of sparse matrix solvers, which work as follows: Given a matrix M, find an approximate M^{-1}, call it $`M^{-1}$, which has roughly the same simple sparsity pattern. This means we can perform $`M^{-1}.F$ very quickly, allowing us to iterate to convergence using $F - M.`M^{-1}.F$, which will become zero as $`M^{-1}$ becomes M^{-1}. For example, the ICCG (Inerative Cholesky Conjugate Gradient) method works well for symmetric matrices, expressing M in the form LL^T and then using the maximum gradient in the error so that the iterations converge rapidly. However, before any solution is possible, we must first define the domain of the simulation and the boundary conditions that apply to it.

We must first define the dimensionless finite-difference domain. In order to make the simulation more general, we can translate the simulation into a spatially dimensionless form. This means we essentially simulate a cake of unit radius and unit height, the results from which can then be scaled/redimensioned to the proportions of the specific cake we are dealing with. For our cake simulation, it is not possible to use a dimensionless form for temperature because the oven is not at a single temperature, but has a vertical gradient, the severity of which would completely alter the results of the simulation. In a single temperature situation, the cake baking would always follow the same pattern, and the rate of baking would simply depend on the difference between the cake's initial temperature and the oven environment temperature. In other words an alteration in the temperature scale of the problem could be be compensated for by using a related alteration in the time scale. With a strong temperature gradient, the cake would bake in a completely different and much more uneven manner than with no temperature gradient and so there is no simple simulation solution that can be scaled (spatially or temporally) to cover all the possible outcomes (with the possible exception of a solution whose degree of temporal scaling is a function of vertical position, but his would be very complicated and perhaps not even possible within the bounds of finite-difference simulation).

I should mention here that when using the spatially dimensionless arrangement we must remember to scale the diffusivity function (χ) from it's normal units (m^2s^{-1}) into the spatially dimensionless form using the relationship between the cake's height and width and the simulation space units (ie for a cake 12 cm in radius (R) and 8 cm in height (H), one vertical simulation unit = 0.08 m and one horizontal space unit = 0.12 m). This of course means that we must scale the diffusivity differently depending on whether we are looking at vertical or horizontal diffusion (this would be accounted for by simply substituting $\Delta X = \Delta x/R$ and $\delta Y = \delta Y/H$ into equation 5 and using a new coordinate system based on X and Y)

Figure 5 illustrates the general form of the grid that would be used for this simulation with I=6 and J= 6, which is somewhat coarser than the grid that the program would actually use.

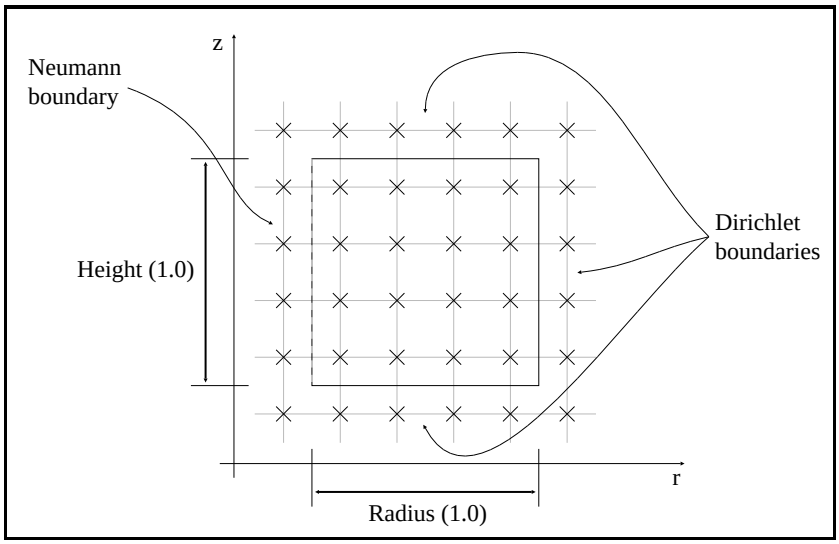

Figure 5: Illustration of the finite-difference domain.

The figure also identifies the boundary conditions that apply to our simulation, of which there are two kinds. Along the points that correspond to the outside edges of the cake, we have Dirichlet boundary conditions, where the value at that grid point is known, in this case it is the oven temperatures as defined by the oven temperature gradient function. Along those points which correspond to the vertical axis of the cake we have a Neumann boundary condition, because the simulation is reflective over that

axis and so there is no net flux of heat across this boundary. The corner points, which are not directly connected to the points inside the cake, are not important, and while they do exist within the matrix formulation of the problem, they are just set to zero and should not interfere with the solution. However, the Dirichlet and Neumann boundary conditions must be incorporated into matrix equation 7 otherwise it will not be solvable. In the case of the Dirichlet points, the exact temperature values, as calculated from the definition of the oven environment, can be substituted into the matrix equation. The Neumann condition is similar but instead of placing an exact value into the matrix, a value of temperature is copied from the points on the inner edge of the inside the cake to their corresponding Neumann boundary points.

While these measures allow us to solve the thermal diffusion problem, I should also outline the finite difference form of the rising, setting and browning processes. As stated earlier, the rates these processes move at have the form:

$$\frac{dr}{dt} = f_r(\theta, s)$$

$$\frac{ds}{dt} = f_s(\theta)$$

$$\frac{db}{dt} = f_b(\theta, s)$$

And so the problem requires us to simultaneously solve three simple differential equations. As we can only evaluate the functions for the current temperature (ie at n, not n+1), in a similar fashion to the behaviour of χ previously, these equations have the following finite difference form (where ε and θ are synonymous):

$$r^{n+1}_{i,j} = r^{n}_{i,j} + f_r(\theta^{n}_{i,j}, s^{n}_{i,j}) \,.\, Dt$$

$$s^{n+1}_{i,j} = s^{n}_{i,j} + f_s(\theta^{n}_{i,j}) \,.\, Dt$$

$$b^{n+1}_{i,j} = b^{n}_{i,j} + f_b(\theta^{n}_{i,j}, s^{n}_{i,j}) \,.\, Dt$$

And so the state of each point within the cake can be calculated as time passes and as the heat of the oven diffuses into the cake.

Flow Chart of the General Code Structure

Initialisation Procedure:

Define required data structures (including $M_{I\times Jmax,I\times Jmax}$, $E_{I\times Jmax}$ and $F_{I\times Jmax}$).

≅

Read in cake defining parameters (either from a data file or from the user).
This may include data which describes characteristics such as χ if they do not have a well defined functional form.
Also read in oven environment parameters.

≅

Initialise state counters (time=0, rising=0, setting=0, browning=0).

≅

Define the mesh for the problem in terms of data structures ($x_{I,J}$, $y_{I,J}$), including a structure which show s whether a particular point is internal, or if it is a boundary point, and if so whether Dirichlet or Neumann (ie of the form bound(I,J) = 0,1,2 for internal, Dirichlet, Neumann). The temperatures for the Dirichlet boundaries can be defined from the oven environment parameters via the $y_{I,J}$ structure.

≅

Given the above parameters, assemble the initial temperature distribution matrix, E_k,
where k=i+(j-1)*I as before.

≅

Define the time between presentation of results (t_{step}).

≅

Main Program Loop:

Assemble the $M_{k,l}$ and F_k matrices from the r ,s, b, E_k and χ data, and then r eset the E_k matrix ready for solution.

≅

Scanning through all the grid points, pick out those which lie on a boundary and insert the Dirichlet/Neumann boundary conditions into
the thermal diffusion matricies.

≅

Solve matrix equation 7 ($M_{k,l}\ E_l = F_k$) using a sparse matrix solver such as ICCG.

≅

Use the new temperature distribution, E_k, to follow the progress of
the differential equations describing the rising, setting and

browning process using the finite difference form mentioned earlier.

≅

Increase the time counter.

≅

Has t_{step} passed since the last calculation loop began?
If NO then loop back over main calculation...
If YES then:

≅

Present the current state of the cake to the user. This would probably consist of three cake plots, one for each of the rising, setting and browning processes. For the 0.0 to 1.0 (to > 1.0) progress scales that r, s and b represent (along with over-rising and over-browning scale data), a palette of colours would be defined so that each element would be coloured in to illistrate the cakes status (probably a blue to green to red colour selection).

≅

Prompt user to ask whether to continue the the simulation for another t_{step} seconds.
If YES then loop back for next t_{step} period...
If NO then:

≅

END