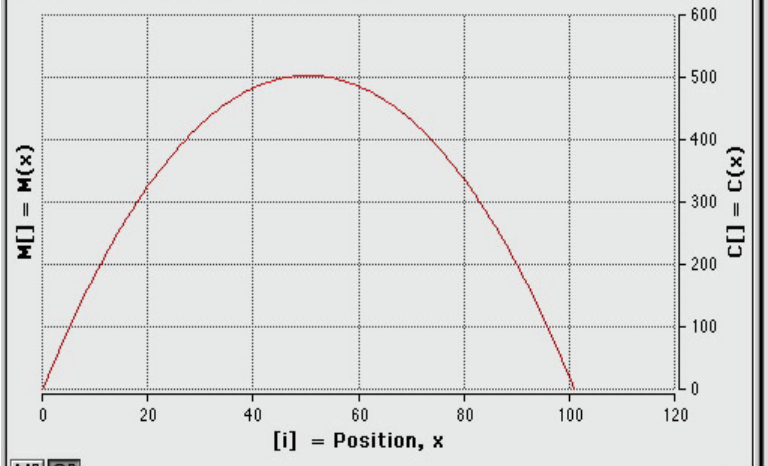
File Edit Flowchart Model Compute Graph Parameters Window Help Tue 10:57

Linear Array - Run 1: C[] vs. TIME

Run 0 F L P Ic Z Run 1: 5000 steps in 0.8 seconds



M[] C[]

Linear Array - Sliders

10x n = 100

10x q = 1.3

10x P = 2.7

Linear Array - Equations

Run

```
METHOD RK4

STARTTIME = 0
STOPTIME = 500
DT = 0.1


d/dt (M [1..n]) = q + P*(C [i-1] -C[i] ) + P*(C [i+1] -C[i] )
C [1..n]  = M[i]/V ;concentration
Init M[1..n] =0 ; initial condition

C[0] = 0 ;boundary condition at the left
C[n+1] = 0 ; boundary condition at the right

q = 1 ; production rate of M in each cell
P = 2 ; permeability of cell walls
V = 0.75 ; volume of a cell
n = 51 ; number of compartments
```

Linear Array - Parameters

Run Runge-Kutta 4

Reset

STARTTIME	0
STOPTIME	500
DT	0.1
DTOUT	* 1
q	* 1.3
P	* 2.7
V	0.75
n	* 100

Linear Array - Notes

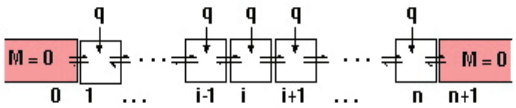
Geneva 16 B I U

Using indexed variables in a linear array of cells

Features:

- Indexed variables (arrays)
- Plotting as a function of position (index)
- Batch Runs

Description of the model:



At time = 0, a linear array of n + 2 cells (compartments) begin producing metabolite **M** at a constant rate **q**. M diffuses between cells with permeability coefficient **P**. The two end cells are slit open so that they are in equilibrium with an infinite bath where M = 0. Cell Concentrations and volumes are denoted by **C** and **V** respectively. The model computes C and M in all cells for time > 0.

Model Equations

Refer to the "Array Equations" section in the Equation Help window (Help menu) for an explanation of array syntax.

The kinetic equations for the inner 50 compartments are defined in terms of the number of moles, M, in each compartment:

d/dt (M [1..n]) = q + P*(C [i-1] -C[i]) + P*(C [i+1] - C[i])
Init M[1..n] =0

With equal volumes V in each compartment, concentrations are defined as

C [1..n] = M[i] / V

and the first and last compartments are reserved for the boundary conditions:

C[0] = 0
C[n+1] = 0

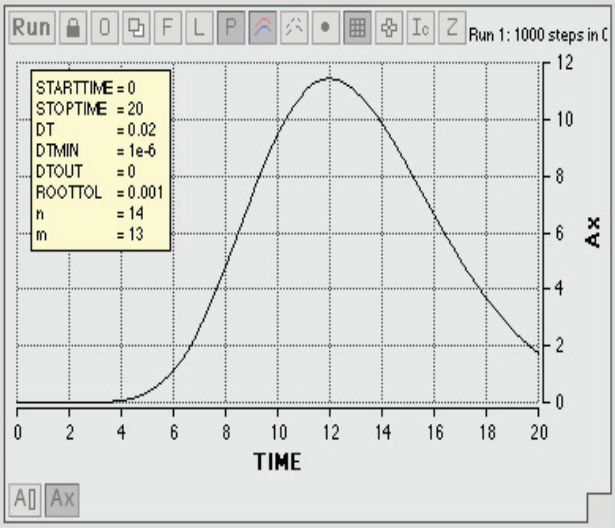
File Edit Flowchart Model Compute Graph Parameters Window Help

Use Indexed Variables (Arrays) - Equations

Run

```
METHOD RK4

STARTTIME = 0
STOPTIME = 20
DT = 0.02

d/dt (A[1]) = -k[1]*A[1]  ; boundary condition at n = 1
init A[1] = 100

d/dt (A[2..n-1]) = k[i-1]*A[i-1]-k[i]*A[i]
init A[2..n]=0

d/dt (A[n]) = k[n-1]*A[n-1]   ; boundary condition at n

k[1..n] =1 {set all rate constants equal}
n = 14 {number of compartments}

Ax = A[m] {defined to look at each A[i] separately}
m= 6
```

Use Indexed Variables (Arrays) - Run 1:Ax vs.TIME



Use Indexed Variables (Arrays) - Sliders

10x m = 13

10x n = 14

Use Indexed Variables (Arrays) - Notes

Arial 18 B I U

Using Indexed Variables (Arrays)

Choose "Equation Help" from the Help menu and look over the section titled "Array Equations".

Note 1: The symbols **i, j,** and **k** are reserved for the running index; they are the only symbols that can be used for this purpose and are available only within the right-hand side of an array equation.

Note 2: The left-hand side of an array equation uses the range operator (two dots "..") to define the extent of each dimension (the first and last index).

Example: The reaction sequence **A1 → A2 → ... → Ai → ... → An**

- The first reactant, A1, is not produced and the last reactant, **An**, is not consumed so they are handled separately by the equations:

```
d/dt (A[1]) = -k[1]*A[1]
d/dt (A[n]) = k[n-1]*A[n-1]
```

- The rest of the reactions i = 2...(n-1) are described by the equation

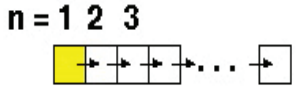
```
d/dt (A[2..n-1]) = k[i-1]*A[i-1] - k[i]*A[i]
```

where k[i] = rate constant for the loss of the ith reactant

- The redundant initial conditions also use indices, as do the rate constants (see Equation window)

- Run the model. The entire indexed set (designated by **A[]** is plotted). Use the legend to distinguish the different plots

- Notice the last two statements which define the variable **Ax = A[m]** (with m arbitrarily set to 6). Clicking the Ax button on the graph will plot A[m] = A[6] for the default. We then define a slider for m to make it easy to obtain individual plots for any member of the set.



File Edit Flowchart Model Compute Graph Parameters Window Help Tue 11:06

Use Multidimensional Arrays - Equations

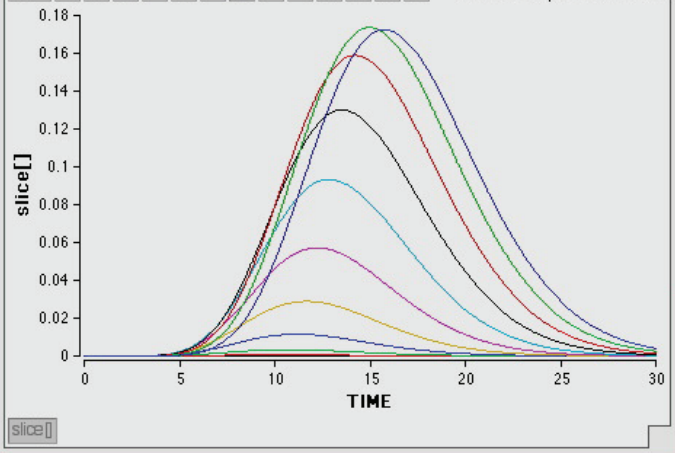
Run

```
{dimensions of matrix}
m = 10
n = 10

{initial material in top-left corner}
INIT S[0..m,0..n] = 0   ;Sets entire matrix to zero
INIT S[0,0] = 10        ;Changes top-left cell

{propagation constants}

a = 2/(sqrt(2)+4)   ;Horizontal/Vertical
b = 1/(sqrt(8)+1)   ;Diagonal

{flow in southeast direction only}

d/dt(S[1..m,1..n]) = a*S[i, j-1] + a*S[i-1, j]+b*S[i-1, j-1] - S[i, j] ;
internal compartments

d/dt(S[0,1..n]) = a*S[i, j-1] - S[i, j] ; m = 0 boundary conditions

d/dt(S[1..m,0]) = a*S[i-1, j] - S[i, j]; n = 0 boundary conditions

d/dt(S[0,0]) = -S[i, j] ;

{Cross section across bottom of sheet}
slice[0..n] = S[m, i]
```

Use Multidimensional Arrays - Run 1: slice[] vs. TIME

Run | 0 | F | L | P | Z

Run 1: 1500 steps in 0.133 seconds



Use Multidimensional Arrays - Notes

Geneva | 14 | B | I | U

Using Multidimensional Arrays

Read the "Array Equations" section in the Equation Help window (Help menu) for basic information about array equations.

This model illustrates flow of material over a surface. The surface "S" is modelled as a 10x10 array of cells. Each cell is a differential equation that accepts material from **three** of its neighboring cells (above, left, and above-left diagonal). The model begins with material in the top-left cell only. As time advances, material flows in a southeast direction over the surface and eventually falls off the right and bottom edges.

1. Initialize the array to zero: INIT S[0..m,0..n] = 0.
2. Place a source in the upper left reservoir: INIT S[0,0] = 10
3. Define the flows between the reservoirs
4. Define the internal compartment conservation laws
5. Define the boundary conditions along m = 0 and n = 0.
6. Define the source in the (0, 0) reservoir.

The last equation in the model. slice[0..n] = S[m, i], copies the row of cells at the bottom of the sheet into a one-dimensional array (vector). This slicing technique lets you plot only the portion of the array you're interested in.

Note that the size of the matrix can be easily changed by adjusting the values of m and m in the **Parameter** window.