As discussed in class, graphs and graph based models can conveniently be represented using objects from linear algebra: vectors and matrices. In the following, let us try to be as general as we can, so let's assume that we have N isotopes that decay into each other (and choose N=5 here), then the rest of the program should look like the following. I've added an alternative graph below that shows the various lines with different colors -- it uses the same idea of how we plotted the various trajectories in programs like the solar system plots.

$N := 5;$

$$5$$

$k := vector\left(N, \left[\ln(2), \frac{\ln(2)}{2}, \frac{\ln(2)}{3}, \ln(2), 0\right]\right);$

$$\begin{bmatrix} \ln(2) & \frac{1}{2}\ln(2) & \frac{1}{3}\ln(2) & \ln(2) & 0 \end{bmatrix} \tag{2}$$

$P := matrix(N, N, [[0, 1, 0, 0, 0], [0, 0, 1, 0, 0], [0, 0, 0, .9, .1], [0, 0, 0, 0, 1], [0, 0, 0, 0, 0]]);$

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0.9 & 0.1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3}$$

$ode := seq(\mathrm{D}(x[i])(t) = -k[i] \cdot x[i](t) + sum(P[j, i] \cdot k[j] \cdot x[j](t), j = 1..N), i = 1..N);$

$$\mathrm{D}(x_1)(t) = -\ln(2)\, x_1(t), \mathrm{D}(x_2)(t) = -\frac{1}{2}\ln(2)\, x_2(t) + \ln(2)\, x_1(t), \mathrm{D}(x_3)(t) = -\frac{1}{3}\ln(2)\, x_3(t) + \frac{1}{2}\ln(2)\, x_2(t), \mathrm{D}(x_4)(t) = -\ln(2)\, x_4(t) + 0.3000000000\ln(2)\, x_3(t), \mathrm{D}(x_5)(t) = 0.03333333333\ln(2)\, x_3(t) + \ln(2)\, x_4(t) \tag{4}$$

$initialAmounts := vector(N, [1, 0, 0, 0, 0]);$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{5}$$

$ic := seq(x[i](0) = initialAmounts[i], i = 1..N);$

$$x_1(0) = 1, x_2(0) = 0, x_3(0) = 0, x_4(0) = 0, x_5(0) = 0 \tag{6}$$

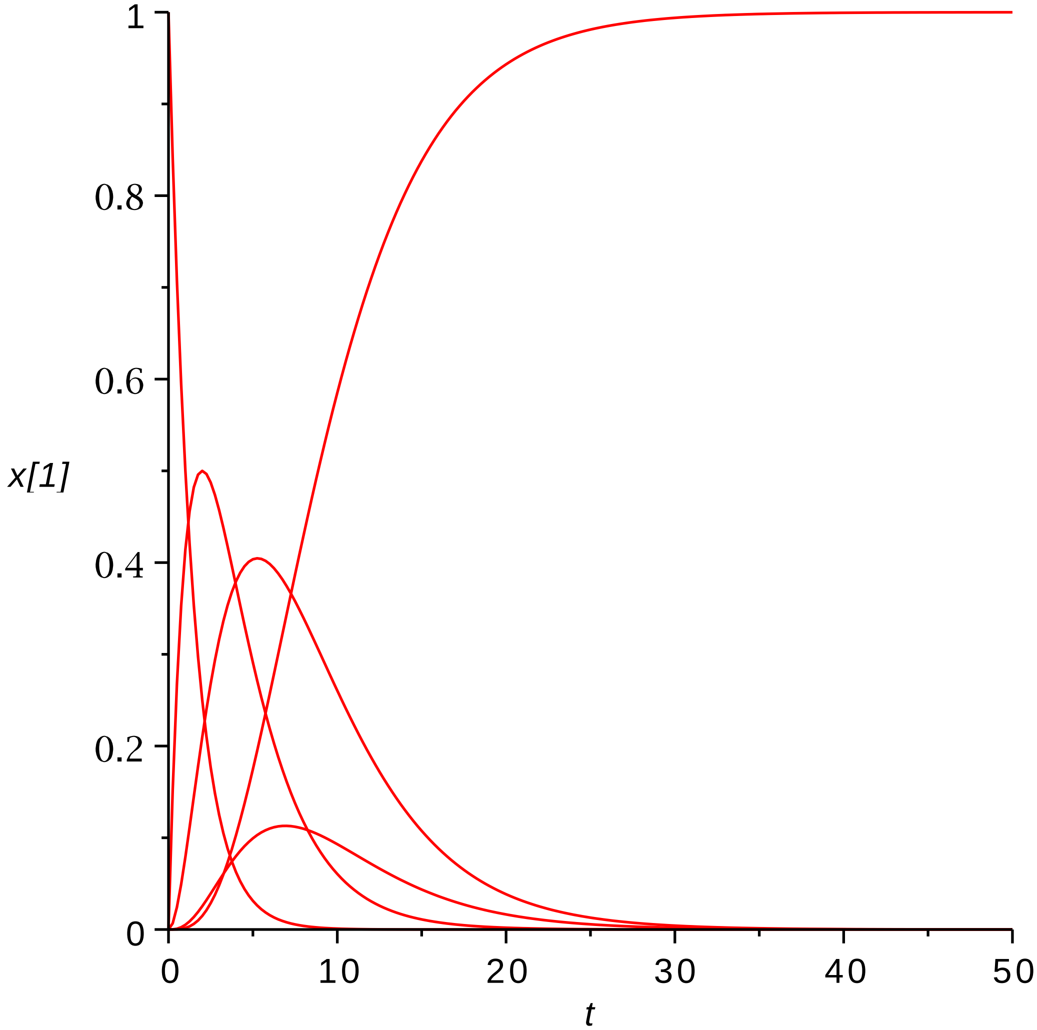
$solution := dsolve(\{ode, ic\}, numeric);$

$$\mathbf{proc}(x\_rkf45)\ \ldots\ \mathbf{end\ proc} \tag{7}$$

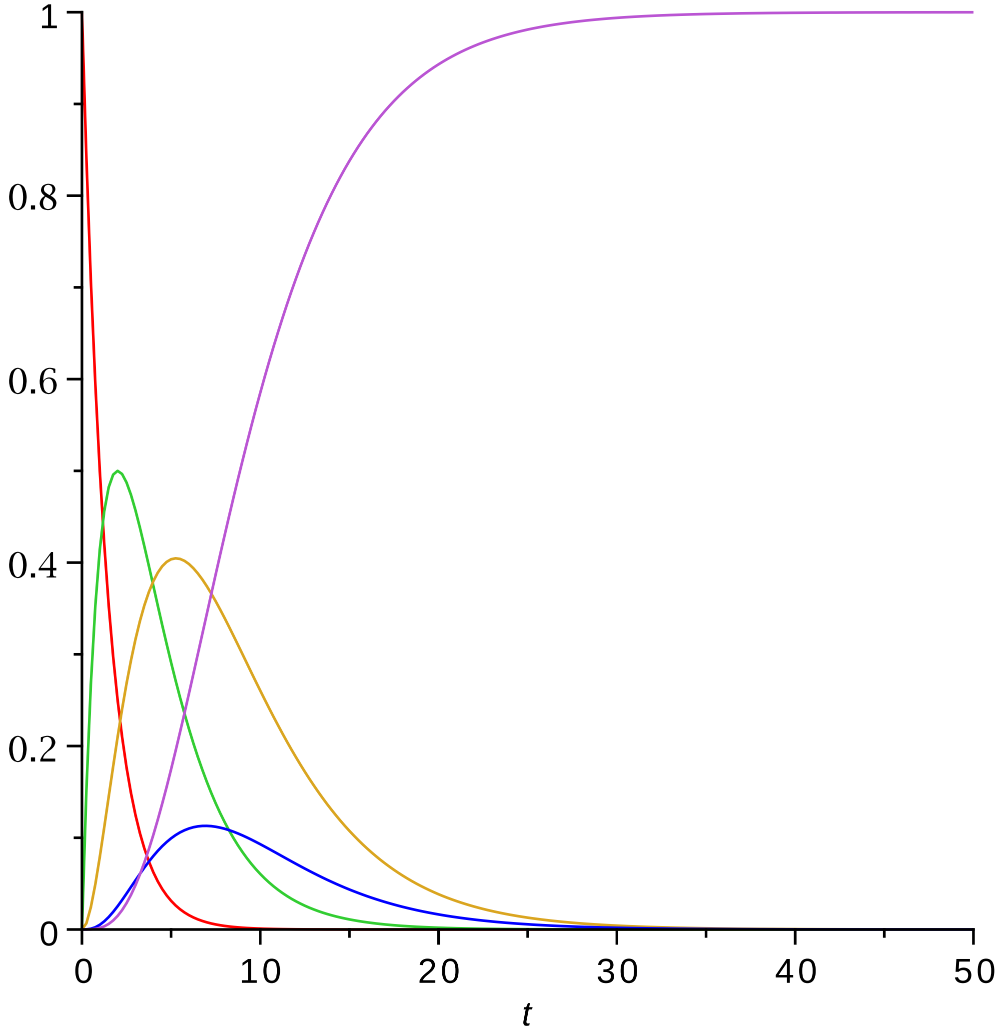
$myplots := seq(plots[odeplot](solution, [t, x[i](t)], t = 0..50), i = 1..N);$

$$PLOT(\ldots), PLOT(\ldots), PLOT(\ldots), PLOT(\ldots), PLOT(\ldots) \tag{8}$$

$plots[display](\{myplots\});$



$plots[odeplot](solution, [seq([t, x[i](t)], i = 1..N)], t = 0..50);$



1
0.8
0.6
0.4
0.2
0
0
10
20
30
40
50
t