Solving Simultaneous Differential Eqns

TABLE OF CONTENTS


```
(%i5) load(draw)$  set_draw_defaults(line_width=2, grid = [2,2], point_type = filled_circle,
          head_type = 'nofilled, head_angle = 20, head_length = 0.5,
            background_color = light_gray,  draw_realpart=false)$
           fpprintprec:5$   ratprint:false$     kill(all)$

(%i1) load ("Econ2.mac");
(%o1) c:/work5/Econ2.mac
```

1 Preface

Dowling19A.wxmx is one of a number of wxMaxima files available in the section
Economic Analysis with Maxima
on my CSULB webpage.

In Dowling19A.wxmx, we use Maxima to discuss solution methods for a set of first order ordinary differential equations. We use desolve, rk, and our own matrix methods. We end with a reconsideration of the inflation and unemployment model discussed in Dowling18C.wxmx, following Chiang and Wainwright's Ch. 19, Sec. 4.

We have changed some of the symbols used in particular problems. An approximate pdf translation (using Microsoft print to pdf) is available as Dowling19Afit.pdf. That pdf file can be searched using Ctrl-F.

Ted Woollett
https://home.csulb.edu/~woollett/
woollett@charter.net
Feb. 25, 2022

2 References

Fundamental Methods of Mathematical Economics, Alpha C. Chiang and Kevin Wainwright, 4th ed., 2005, McGraw-Hill

3 Don't Try Using ode2 for a Set of First Order ODE's

How not to solve a pair of first order ODE's: trying to use ode2:

```
(%i4) de1 : 'diff (y1,t) = 5*y1 - 0.5*y2 - 12;
      de2 : 'diff (y2,t) = -2*y1 + 5*y2 - 24;
      solns : ode2 ([de1, de2], [y1, y2], t);
```

(de1) $\frac{d}{d t} y1 = -0.5\, y2 + 5\, y1 - 12$

(de2) $\frac{d}{d t} y2 = 5\, y2 - 2\, y1 - 24$

(solns) $[\frac{d}{d t} y1 = -0.5\, y2 + 5\, y1 - 12, \frac{d}{d t} y2 = 5\, y2 - 2\, y1 - 24]$

not a proper differential equation

(solns) ***false***

4 Solutions Using desolve ([eqn_1, ..., eqn_n], [x_1, ..., x_n])

Solve the following system of first-order, autonomous, linear differential equations:
dy1/dt = 5*y1 - 0.5*y2 - 12, y1(0) = 12,
dy2/dt = - 2*y1 + 5*y2 - 24, y2(0) = 4.

```
(%i9) atvalue (y1(t), t = 0, 12)$
      atvalue (y2(t), t = 0, 4)$
      eqn1 : diff (y1(t), t) = 5*y1(t) - 0.5*y2(t) - 12;
      eqn2 : diff (y2(t), t) = - 2*y1(t) + 5*y2(t) - 24;
      soln : desolve ([eqn1, eqn2], [y1(t), y2(t)]);
```

(eqn1) $\frac{d}{d\,t}\,y1(t)=-0.5\,y2(t)+5\,y1(t)-12$

(eqn2) $\frac{d}{d\,t}\,y2(t)=5\,y2(t)-2\,y1(t)-24$

(soln) $[y1(t)=5\,\%e^{6\,t}+4\,\%e^{4\,t}+3, y2(t)=-10\,\%e^{6\,t}+8\,\%e^{4\,t}+6]$

```
(%i10) grind(%)$
```

[y1(t) = 5*%e^(6*t)+4*%e^(4*t)+3,y2(t) = (-10*%e^(6*t))+8*%e^(4*t)+6]$

Let y1ex and y2ex be Maxima expressions which depend on the value of t.

```
(%i11) [y1ex, y2ex] : map ('rhs, soln);
```

(%o11) $[5\,\%e^{6\,t}+4\,\%e^{4\,t}+3, -10\,\%e^{6\,t}+8\,\%e^{4\,t}+6]$

```
(%i12) at([y1ex, y2ex], t = 0);
```

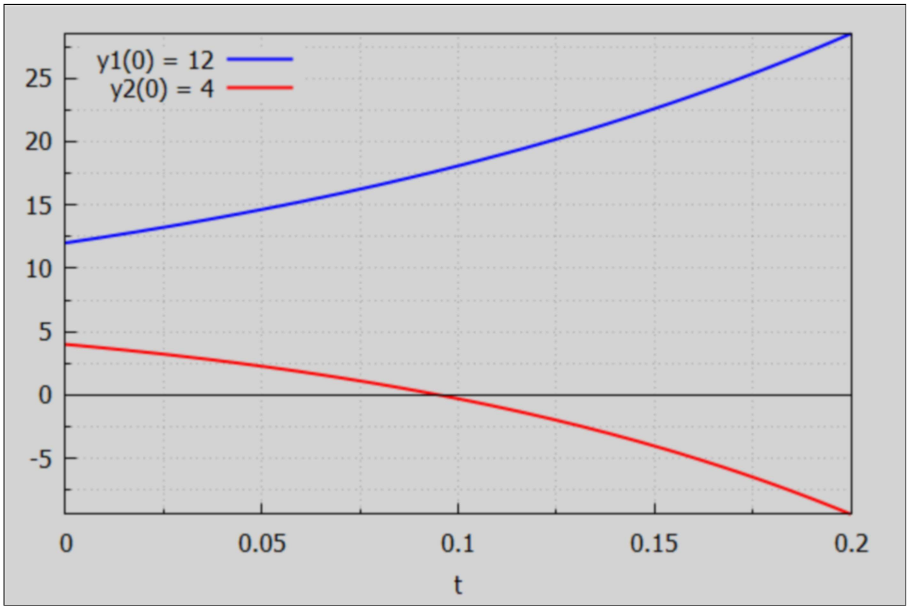
(%o12) $[12,4]$

It turns out that the eigenvalues (characteristic roots) of the coefficient matrix are (6, 4), and the "complementary solution" yc is a linear combination of exp(r1*t) and exp(r2*t), and for this set of 1st order ODE's, r1 = 6, r2 = 4 are both positive, leading to an unstable model/solution.

y1ex is always positive, and y2ex will go negative quickly from its initial value 4, since 6*t dominates 4*t in an exponent.

(%i14)
```
xmax : 0.2$
wxdraw2d (xlabel = "t", key_pos = top_left,
    key = "y1(0) = 12", explicit (y1ex, t, 0, xmax),
    color = red, key = "y2(0) = 4",  explicit (y2ex, t, 0, xmax), key = "",
    color = black, line_width = 1, explicit (0, t, 0, xmax));
```

(%t14)



(%o14)

(%i15)
```
find_root (y2ex, t, 0.05, 0.15);
```

(%o15) 0.095263

So y2ex is negative for t > 0.0953.

5 *Solutions Using Maxima's Runge-Kutta Routine rk*

The purely numerical Maxima function rk can be used for one or more first order ODE's.

For one ODE the syntax is
  results :   rk (dydt, y, y0, [t, t0, tlast, dt])
where dydt depends of y and t, y is the independent variable whose value at t0 is y0.
t is the independent variable, and a list (we have called results here):
    [[t0, y0], [t0 + dt, y(t + dt)], ... [tlast, y(tlast)]]
is returned corresponding to the requested interval dt. The returned list (suppose we call it results) can immediately be plotted using wxdraw2d (points (results)).

For two ODE's the syntax is
   results :   rk ([dudt, dvdt], [u, v], [u0, v0], [t, t0, tlast, dt])
where u and v are the two dependent variables, t is the independent variable, dudt and dvdt each in general depend on u, v, and t, and u0 and v0 are the respective values of u and v when t = t0.

For a pair of first order ODE's the list 'results' looks like:
   [[t0, u(t0), v(t0)], [t0 + dt, u(t0+dt), v(t0 + dt)], ..., [tlast, u(tlast), v(tlast)]].

One can then use the Maxima function makelist to form a (t,u) points list via:
  tu_points : makelist ([results[j][1], results [j][2]], j, 1, length (results)),
and form a (t,v) points list via:
  tv_points : makelist ([results[j][1], results [j][3]], j, 1, length (results)).

One can then use (for example):
  wxdraw2d (points (tu_points), color = red, points(tv_points)).

Solve the following system of first-order, autonomous, linear differential equations:
  dy1/dt = 5*y1 - 0.5*y2 - 12, y1(0) = 12,
  dy2/dt = - 2*y1 + 5*y2 - 24, y2(0) = 4
using the Maxima Runge-Kutta numerical integrator rk.

```
(%i16) results : rk ( [5*y1 - 0.5*y2 -12, -2*y1 + 5*y2 - 24], [y1, y2], [12, 4], [t, 0, 0.2, 0.01] );
(results) [[0.0,12.0,4.0],[0.01,12.472,3.7081],[0.02,12.971,3.3913],[
0.03,13.496,3.0478],[0.04,14.05,2.6756],[0.05,14.635,2.2726],[0.06,
15.252,1.8367],[0.07,15.902,1.3654],[0.08,16.589,0.85628],[0.09,
17.313,0.30657],[0.1,18.078,−0.28659],[0.11,18.885,−0.92626],[0.12,
19.736,−1.6157],[0.13,20.635,−2.3585],[0.14,21.585,−3.1583],[0.15,
22.586,−4.0191],[0.16,23.644,−4.9451],[0.17,24.761,−5.9409],[0.18,
25.941,−7.0113],[0.19,27.187,−8.1615],[0.2,28.503,−9.3968]]
```

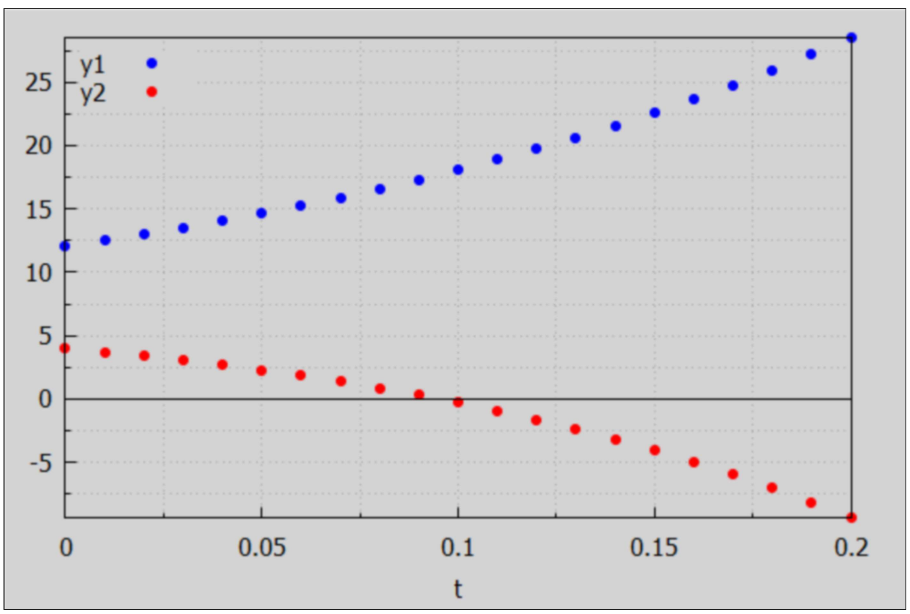
(%i17) ty1_pts : makelist ([results[j][1], results[j][2]], j, 1, length (results));

(ty1_pts) [[0.0,12.0],[0.01,12.472],[0.02,12.971],[0.03,13.496],[0.04,14.05],[0.05,14.635],[0.06,15.252],[0.07,15.902],[0.08,16.589],[0.09,17.313],[0.1,18.078],[0.11,18.885],[0.12,19.736],[0.13,20.635],[0.14,21.585],[0.15,22.586],[0.16,23.644],[0.17,24.761],[0.18,25.941],[0.19,27.187],[0.2,28.503]]

(%i18) ty2_pts : makelist ([results[j][1], results[j][3]], j, 1, length (results));

(ty2_pts) [[0.0,4.0],[0.01,3.7081],[0.02,3.3913],[0.03,3.0478],[0.04,2.6756],[0.05,2.2726],[0.06,1.8367],[0.07,1.3654],[0.08,0.85628],[0.09,0.30657],[0.1,−0.28659],[0.11,−0.92626],[0.12,−1.6157],[0.13,−2.3585],[0.14,−3.1583],[0.15,−4.0191],[0.16,−4.9451],[0.17,−5.9409],[0.18,−7.0113],[0.19,−8.1615],[0.2,−9.3968]]

(%i19) wxdraw2d (xlabel = "t", key_pos = top_left, key = "y1", points (ty1_pts), color = red, key = "y2", points (ty2_pts), key = "", color = black, line_width = 1, explicit (0, t, 0, 0.2))$

(%t19) 

The numerical Maxima rk method of solution can be used for any number of first order ODE's. Higher order ODE's can be turned into a set of first order ODE's.

6 *Solutions for dY/dt = A . Y(t) + B using Matrix Methods*

6.1 Review of Maxima Matrix Functions

To use matrix methods to find y1(t) and y2(t), we need to review Maxima's matrix syntax.

matrix(row_1,...,row_n), Create a rectangular matrix with rows: row_1,. . . ,row_n
zeromatrix (m, n), m rows, n columns matrix, with all elements = 0,
A + B, Sum of matrices A and B
A - B, Difference of matrices A and B
s*A, Multiply matrix A with scalar s
A . B, Matrix Product of matrices A and B
A^^n, n-th power of matrix A, i.e., A . A.· · · . A n times
A^^(-1) or invert (A), Inverse of matrix A
determinant (A), or det (A) from Econ2.mac, returns the determinant.
A[n], row n of matrix A
A[n, m], (n,m) element of matrix A = (row n, column m) element of matrix A
ident(n), Identity matrix of order n
transpose(A), Transpose of matrix A
matrixexp (A, t) "matrix exponential function" %e^^(t*A) for given matrix A
charpoly(A,x), Characteristic polynomial of A
eigenvalues(A) ==> [list of eigenvalues of matrix A, list of multiplicity of each eigenvalue]

eigenvectors(A) ==> [eigenvalues(A), list of corresponding eigenvector components]

Additional matrix functions defined in Econ2.mac are
cvec, mtrace, det, squarep, colVecSolve, ODE1S, ODE2S.

6.2 Maxima Function ODE1S (A, B, Y0) Syntax

The Maxima function ODE1S (A, B, Y0), defined in Econ2.mac, uses matrix methods to solve for the solution of dY/dt = A . Y(t) + B, in which Y is a matrix column vector with n elements depending on t, A is a square n x n matrix of numerical elements, and B is either a given n element matrix column vector (with constant numerical values), or if there is no given B column vector in the problem, B is replaced by either the number 0 or replaced by zeromatrix (n, 1). Finally Y0 is an n element numerical matrix column vector giving the desired initial values Y(0).

For a problem in which there is no B term, and we are solving dY/dt = A . Y(t), with Y(0) = Y0, we can either use ODE1S (A, 0, Y0) or ODE1S (A, zeromatrix (n, 1), Y0), in which
n = length(A) = length (Y0).

It is actually easier to just use (if no B in the problem):

```
Ys : matrixexp (A, t) . Y0,
   or  Ys : expand (matrixexp (A, t) . Y0 )    in such a problem.
```

6.3 Solution of dY(t)/dy = A . Y(t), with Y(0) given, using matrixexp (A, t)

We assume A is a 2 x 2 matrix with row1 = [6, 5], row2 = [1, 2], Y(t) is a 2 element column vector depending on t, and Y(0) = transpose (matrix ([4,1])) = cvec([4,1]). The Maxima function cvec, defined in Econ2.mac, creates a matrix column vector from a list of components.

This matrix ODE dY/dt = A . Y(t) then stands for the two component equations, with Y = cvec([y1, y2]):

dy1/dt = 6*y1 + 5*y2,
dy2/dt = y1 + 2*y2.

```
(%i23) A : matrix ([6, 5], [1, 2]);
       Y0 : cvec ([4,1]);
       Ys : expand (matrixexp (A, t) . Y0);
       at (Ys, t = 0);
```

(A) $\begin{pmatrix} 6 & 5 \\ 1 & 2 \end{pmatrix}$

(Y0) $\begin{pmatrix} 4 \\ 1 \end{pmatrix}$

(Ys) $\begin{pmatrix} \frac{25\ \%e^{7\,t}}{6} - \frac{\%e^{t}}{6} \\ \frac{5\ \%e^{7\,t}}{6} + \frac{\%e^{t}}{6} \end{pmatrix}$

(%o23) $\begin{pmatrix} 4 \\ 1 \end{pmatrix}$

6.4 Solution of dY(t)/dy = A . Y(t), with Y(0) given, using ODE1S

Since there is no given column vector B here, we can replace B by the number 0:

```
(%i25) Ys : ODE1S (A, 0, Y0);
       at (Ys, t = 0);
```

(Ys) $\begin{pmatrix} \frac{25\ \%e^{7\,t}}{6} - \frac{\%e^{t}}{6} \\ \frac{5\ \%e^{7\,t}}{6} + \frac{\%e^{t}}{6} \end{pmatrix}$

(%o25) $\begin{pmatrix} 4 \\ 1 \end{pmatrix}$

or we can use B --> zeromatrix (2,1).

(%i28) B : zeromatrix (2,1);
Ys : ODE1S (A, B, Y0);
at (Ys, t = 0);

(B) $\begin{pmatrix} 0 \\ 0 \end{pmatrix}$

(Ys) $\begin{pmatrix} \frac{25\,\%e^{7\,t}}{6} - \frac{\%e^{t}}{6} \\ \frac{5\,\%e^{7\,t}}{6} + \frac{\%e^{t}}{6} \end{pmatrix}$

(%o28) $\begin{pmatrix} 4 \\ 1 \end{pmatrix}$

6.5 Construction of the matrix exponential "by hand".

We can manufacture the "matrix exponential" e^^(t*A) "by hand" by defining a 2x2 matrix φi(t) such that Y(t) = k1*exp(r1*t)*V1 + k2*exp(r2*t)*V2, where k1 and k2 are constants to be determined, is written in the alternative form:
Y(t) = φi(t) . matrix([k1], [k2]) = φi(t) . cvec ([k1,k2]).
We then let φi0 = at (φi, t = 0), and define the matrix exponential as
EAt : φi (t). invert (φi0), which has the property: at (EAt, t = 0) = ident(2) = matrix([1,0],[0,1]).

(%i29) φi : matrix ([exp (r1*t)*V1[1,1], exp (r2*t)*V2[1,1]],
[exp (r1*t)*V1[2,1] , exp (r2*t)*V2[2,1]]);

(φi) $\begin{pmatrix} V1_{1,1}\,\%e^{r1\,t} & V2_{1,1}\,\%e^{r2\,t} \\ V1_{2,1}\,\%e^{r1\,t} & V2_{2,1}\,\%e^{r2\,t} \end{pmatrix}$

(%i30) φi0 : at (φi, t = 0);

(φi0) $\begin{pmatrix} V1_{1,1} & V2_{1,1} \\ V1_{2,1} & V2_{2,1} \end{pmatrix}$

Here is our "by hand" matrix exponential we call EAt:

(%i34) EAt : φi . invert (φi0), ratsimp;
at (EAt, t = 0);
Y0;
EAt . Y0, expand;

(EAt)
$$\begin{pmatrix} -\frac{V2_{1,1}\,V1_{2,1}\,\%e^{r2\,t}-V1_{1,1}\,V2_{2,1}\,\%e^{r1\,t}}{V1_{1,1}\,V2_{2,1}-V2_{1,1}\,V1_{2,1}} & \frac{V1_{1,1}\,V2_{1,1}\,\%e^{r2\,t}-V1_{1,1}\,V2_{1,1}\,\%e^{r1\,t}}{V1_{1,1}\,V2_{2,1}-V2_{1,1}\,V1_{2,1}} \\ -\frac{V1_{2,1}\,V2_{2,1}\,\%e^{r2\,t}-V1_{2,1}\,V2_{2,1}\,\%e^{r1\,t}}{V1_{1,1}\,V2_{2,1}-V2_{1,1}\,V1_{2,1}} & \frac{V1_{1,1}\,V2_{2,1}\,\%e^{r2\,t}-V2_{1,1}\,V1_{2,1}\,\%e^{r1\,t}}{V1_{1,1}\,V2_{2,1}-V2_{1,1}\,V1_{2,1}} \end{pmatrix}$$

(%o32)
$$\begin{pmatrix} -\frac{V2_{1,1}\,V1_{2,1}-V1_{1,1}\,V2_{2,1}}{V1_{1,1}\,V2_{2,1}-V2_{1,1}\,V1_{2,1}} & 0 \\ 0 & 1 \end{pmatrix}$$

(%o33)
$$\begin{pmatrix} 4 \\ 1 \end{pmatrix}$$

(%o34)
$$\begin{pmatrix} -\frac{4\,V2_{1,1}\,V1_{2,1}\,\%e^{r2\,t}}{V1_{1,1}\,V2_{2,1}-V2_{1,1}\,V1_{2,1}}+\frac{V1_{1,1}\,V2_{1,1}\,\%e^{r2\,t}}{V1_{1,1}\,V2_{2,1}-V2_{1,1}\,V1_{2,1}}+\frac{4\,V1_{1,1}\,V2_{2,1}\,\%e^{r1\,t}}{V1_{1,1}\,V2_{2,1}-V2_{1,1}\,V1_{2,1}}-\frac{V1_{1,1}\,V2_{1,1}\,\%e^{r1\,t}}{V1_{1,1}\,V2_{2,1}-V2_{1,1}\,V1_{2,1}} \\ -\frac{4\,V1_{2,1}\,V2_{2,1}\,\%e^{r2\,t}}{V1_{1,1}\,V2_{2,1}-V2_{1,1}\,V1_{2,1}}+\frac{V1_{1,1}\,V2_{2,1}\,\%e^{r2\,t}}{V1_{1,1}\,V2_{2,1}-V2_{1,1}\,V1_{2,1}}+\frac{4\,V1_{2,1}\,V2_{2,1}\,\%e^{r1\,t}}{V1_{1,1}\,V2_{2,1}-V2_{1,1}\,V1_{2,1}}-\frac{V2_{1,1}\,V1_{2,1}\,\%e^{r1\,t}}{V1_{1,1}\,V2_{2,1}-V2_{1,1}\,V1_{2,1}} \end{pmatrix}$$

which we compare with matrixexp (A, t):

(%i39)
```
A;
MEAt : matrixexp (A,t);
at (MEAt, t = 0);
Y0;
MEAt . Y0, expand;
```

(%o35) $\begin{pmatrix} 6 & 5 \\ 1 & 2 \end{pmatrix}$

(MEAt) $\begin{pmatrix} \frac{5\ \%e^{7\,t}+\%e^{t}}{6} & \frac{5\ \%e^{7\,t}-5\ \%e^{t}}{6} \\ \frac{\%e^{7\,t}-\%e^{t}}{6} & \frac{\%e^{7\,t}+5\ \%e^{t}}{6} \end{pmatrix}$

(%o37) $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

(%o38) $\begin{pmatrix} 4 \\ 1 \end{pmatrix}$

(%o39) $\begin{pmatrix} \frac{25\ \%e^{7\,t}}{6}-\frac{\%e^{t}}{6} \\ \frac{5\ \%e^{7\,t}}{6}+\frac{\%e^{t}}{6} \end{pmatrix}$

6.6 Solutions of dY(t)/dt = A . Y(t) + B, with Y(0) given, using ODE1S

Here we return to the same set of ODE's we solved using desolve in the first section, but now we use matrix methods.

Solve the following system of first-order, autonomous, linear differential equations:
dy1/dt = 5*y1 - 0.5*y2 - 12, y1(0) = 12,
dy2/dt = - 2*y1 + 5*y2 - 24, y2(0) = 4.

We convert these two differential equations into one matrix equation:
dY/dt = A . Y + B,
In which A is a 2 x 2 square matrix of coefficients:

(%i40)
```
A : matrix ([5, -0.5], [-2, 5] );
```

(A) $\begin{pmatrix} 5 & -0.5 \\ -2 & 5 \end{pmatrix}$

and B is a 2 element matrix column vector:

(%i41) B : cvec ([-12, -24]);

(B) $\begin{pmatrix}-12\\-24\end{pmatrix}$

The "intertemporal solution" Ye is found by assuming dY/dt = 0: A . Ye = - B, or
Ye = - invert (A) . B.

(%i42) Ye : - invert (A) . B;

(Ye) $\begin{pmatrix}3.0\\6.0\end{pmatrix}$

Define Y0 as the matrix column vector of initial values of the components of Y(t).
We then calculate Y(t), calling it Ys, using ODE1S (A, B, Y0).

```
(%i45) Y0 : cvec ([12, 4]);
       Ys : ODE1S (A, B, Y0);
       at (Ys, t = 0);
```

(Y0) $\begin{pmatrix}12\\4\end{pmatrix}$

(Ys) $\begin{pmatrix}5\ \%e^{6\,t}+4\ \%e^{4\,t}+3.0\\-10\ \%e^{6\,t}+8\ \%e^{4\,t}+6.0\end{pmatrix}$

(%o45) $\begin{pmatrix}12.0\\4.0\end{pmatrix}$

6.7 Matrix Details of the ODE1S Solution of dY(t)/dy = A . Y(t) + B

We go through some of the gory details of employing the standard Maxima matrix functions which we used to write the Maxima function ODE1S in Econ2.mac

We get both a list of eigenvalues of A and a list of corresponding eigenvectors (in a list form) by using the standard Maxima function eigenvectors (A).

```
(%i47) A;
       [ev, evec] : eigenvectors (A);
```

(%o46) $\begin{pmatrix}5 & -0.5\\-2 & 5\end{pmatrix}$

(%o47) [[[6,4],[1,1]],[[[1,−2]],[[1,2]]]]

The list ev contains two lists; the first is a list of two eigenvalues found, r1 = 6, r2 = 4.
The second list in ev is a list of the multiplicity of each eigenvalue found, here the multiplicity is 1 for both r1 and for r2, which is what we want in order to use ODE1S.

```
(%i48) ev;
(%o48) [[6,4],[1,1]]
```

```
(%i49) ev[1];
(%o49) [6,4]
```

Here is our r1:

```
(%i50) ev[1][1];
(%o50) 6
```

and here is our r2:

```
(%i51) ev[1][2];
(%o51) 4
```

The list evec is

```
(%i52) evec;
(%o52) [[[1,−2]],[[1,2]]]
```

which consists of two sublists and hence a "length" of 2:

```
(%i53) length(evec);
(%o53) 2
```

evec[1][1] gives a list of the elements of the eigenvector corresponding to r1 = 1.

```
(%i54) evec[1];
(%o54) [[1,−2]]
```

Here are the components of the eigenvector V1 corresponding to the eigenvalue r1:

```
(%i55) evec[1][1];
(%o55) [1,−2]
```

which we can turn into a matrix column vector using cvec (list) defined in Econ2.mac:

(%i56) cvec (evec[1][1]);

(%o56) $\begin{pmatrix} 1 \\ -2 \end{pmatrix}$

Here we define r1, and V1, and show A . V1 = r1*V1.

```
(%i60) r1 : 6;
V1 : cvec ([1, -2]);
A . V1;
r1*V1;
```

(r1) 6

(V1) $\begin{pmatrix} 1 \\ -2 \end{pmatrix}$

(%o59) $\begin{pmatrix} 6.0 \\ -12 \end{pmatrix}$

(%o60) $\begin{pmatrix} 6 \\ -12 \end{pmatrix}$

To see the actual way you would need to type this in from scratch, it helps to set display2d to false temporarily:

```
(%i63) display2d : false$
V1;
display2d : true$
```

(%o62) matrix([1],[-2])

You need to be careful about getting the first and second elements of the vector V1 using Maxima's list notation:

```
(%i65) V1[1,1];
V1[2,1];
```

(%o64) 1

(%o65) −2

Here we repeat the above with r2 and V2:

```
(%i69) r2 : 4;
       V2 : cvec ([1, 2]);
       A . V2;
       r2*V2;
```

(r2) 4

(V2) $\begin{pmatrix} 1 \\ 2 \end{pmatrix}$

(%o68) $\begin{pmatrix} 4.0 \\ 8 \end{pmatrix}$

(%o69) $\begin{pmatrix} 4 \\ 8 \end{pmatrix}$

```
(%i72) V2;
       V2[1,1];
       V2[2,1];
```

(%o70) $\begin{pmatrix} 1 \\ 2 \end{pmatrix}$

(%o71) 1

(%o72) 2

We then write Y(t) as the sum of the particular solution Ye and the complementary solution containing two constants k1 and k2, to be determined by the initial conditions.

```
(%i74) [k1, k2];
       Y : Ye + k1*exp(r1*t)*V1 + k2*exp(r2*t)*V2;
```

(%o73) $[k1,k2]$

(Y) $\begin{pmatrix} k1\ \%e^{6\,t}+k2\ \%e^{4\,t}+3.0 \\ -2\ k1\ \%e^{6\,t}+2\ k2\ \%e^{4\,t}+6.0 \end{pmatrix}$

To apply the initial conditions, require Y(0) = Y0. Here we use our Maxima function colVecSolve (C1, C2) (defined in Econ2.mac), where C1 and C2 are each a matrix column vector.

```
(%i76) Y0;
       solns : colVecSolve (at (Y, t = 0), Y0);
```

(%o75) $\begin{pmatrix} 12 \\ 4 \end{pmatrix}$

(solns) $[k2=4,k1=5]$

Now specialize the indefinite solution Y written down above, using solns:

(%i77) Ys : at (Y, solns);

(Ys) $\begin{pmatrix} 5\,\%e^{6\,t}+4\,\%e^{4\,t}+3.0 \\ -10\,\%e^{6\,t}+8\,\%e^{4\,t}+6.0 \end{pmatrix}$

Here is our code for colVecSolve:

```
(%i80) display2d : false$
       fundef (colVecSolve);
       display2d : true$
```

```
(%o79) colVecSolve(%B,%C):=block([%eqnL:[],num:length(%B)],
            if length(%C) # num
               then return(" column vecs must be same length"),
            for j thru num do %eqnL:cons(%B[j,1] = %C[j,1],%eqnL),
            solve(%eqnL)[1])
```

7 Stability and Phase Diagrams for a System of Two ODE's

We follow Dowling Sec. 19.5 "Stability and Phase Diagrams for Simultaneous Differential Equations", together with Chiang & Wainwright Sec. 19.5 "Two Variable Phase Diagrams".

In our Dowling Ch. 16, Sec. 6, we discussed the use of phase diagram analysis for a single ordinary differential equation of the form dx/dt = f(x(t)), and there we made plots of vx = dx/dt (on the vertical axis) versus x (on the horizontal axis), to discuss stability of solution questions.

In this section, we instead make a plot of y2 on the vertical axis and y1 on the horizontal axis. At every instant t, there will be a point in this (y1, y2) plane which represents the momentary state of the two (dependent) variable system.

We start with the matrix form
  dY/dt = A . Y(t) + B,
in which A = matrix ([a11, a12], [a21, a22]), and B = matrix ([b1], [b2]),
so the single matrix equation stands for the set of two ode's:
  dy1/dt = a11*y1 + a12*y2 + b1,
  dy2/dt = a21*y1 + a22*y2 + b2.

Quoting Dowling, p. 439, (with a few editorial changes)
"Given a system of linear autonomous differential equations, the intertemporal equilibrium level will be asymptotically stable, i.e., Y(t) willl converge to Ye as t --> ∞, if and only if both characteristic roots are negative. In the case of complex roots, the real parts must be negative. If all the roots are positive, the system will be unstable. A 'saddle point equilibrium', in which the roots have opposite signs, will usually be unstable. An exception to the latter rule is the case in which the initial conditions y10 and y20 satisfy
 y20 = (r1 - a11)*(y10 - y1e)/a12 + y2e,
where r1 = the negative root, we have what is called a 'saddle path', y10 and y20 happen to be on the saddle path, and y1(t) and y2(t) will then converge to their intertemporal equilibrium level (see Example 10)."

"A 'phase diagram' for a system of two differential equations, linear or nonlinear, graphs y2 on the vertical axis and y1 on the horizontal axis. The (y1, y2) plane is called the 'phase plane'. Construction of a phase diagram is easiest explained in terms of an example."

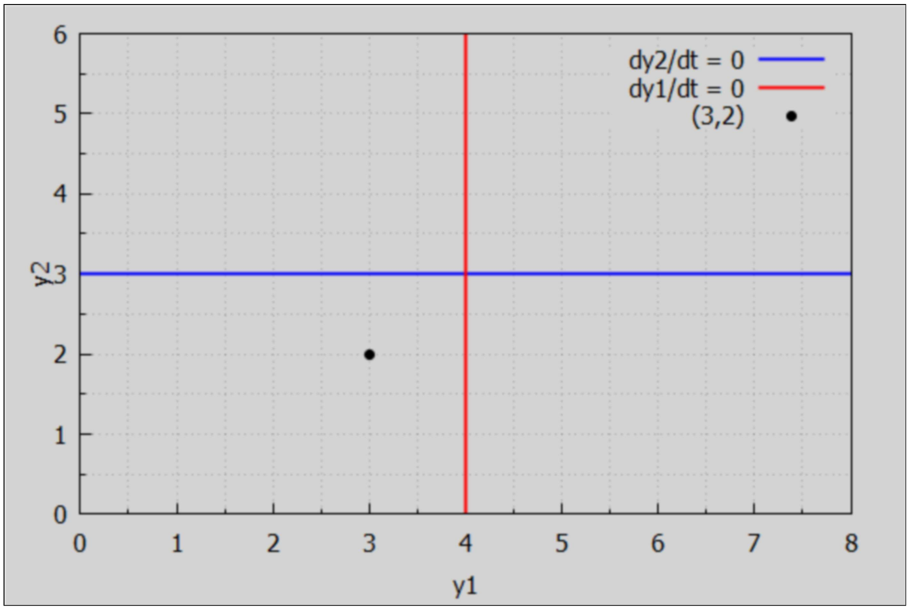
7.1 Example 9: Convergent Phase Diagram Case Analysis

Given the system of linear autonomous differential equations
 dy1/dt = - 4 y1 + 16,
 dy2/dt = - 5 y2 + 15,
a phase diagram is used below to test the stability of the model. Since neither dependent variable is a function of the other dependent variable in this simple model, each equation can be graphed independently.

1. Determine the intertemporal equilibrium values y1e, y2e. y1e is a solution for which dy1/dt = 0, so y1e = 4. Likewise y2e is a solution for which dy2/dt = 0, or y2e = 3.

The line y2 = 3 (the "y2 isocline") divides the phase plane (y1,y2) into two isosectors, one above the isocline and one below. The intersection of the isoclines defines the intertemporal equilibrium Ye = (4,3).

(%i81) wxdraw2d (xlabel = "y1", ylabel = "y2", xrange = [0, 8], yrange = [0, 6], key = "dy2/dt = 0",
explicit (3, x, 0, 8), color = red, key = "dy1/dt = 0", parametric (4, yy, yy, 0, 6),
color = black, key = "(3,2)", points ([[3, 2]]))$

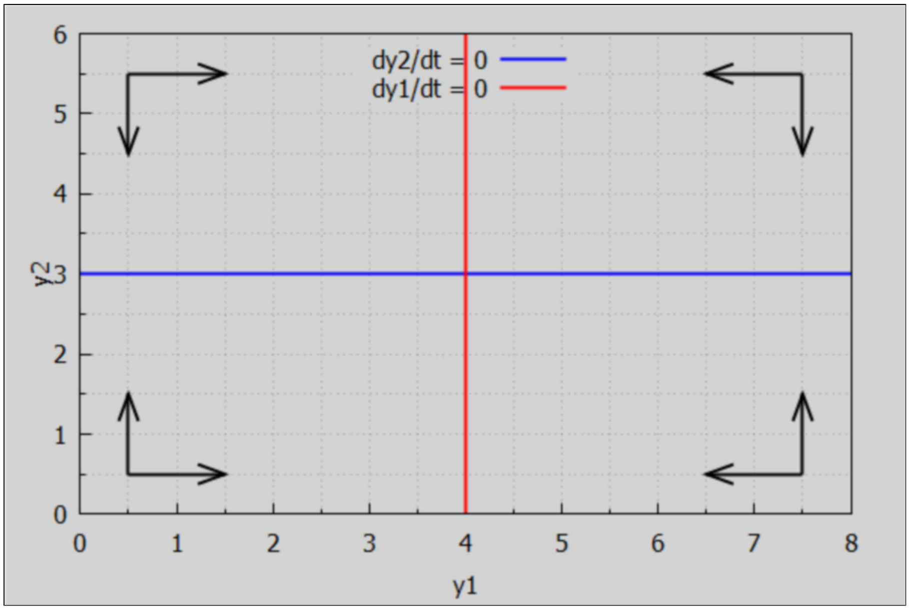
(%t81) 

2. Determine the motion around the y1 isocline, using arrows of horizontal motion.
To the left of the y1 isocline, y1 < 4 and dy1/dt will be positive (y1 will be increasing), using the first ode. To the right of the y1 isocline, y1 > 4 and dy1/dt will be negative (y1 will be decreasing).
This motion of the system point will occur for any value of y2

3. Determine the motion around the y2 isocline. Above the y2 isocline, y2 > 3 and dy2/dt will be negative (y2 will be decreasing), using the second ode. Below the y2 isocline, y2 < 3 and dy2/dt will be positive (y2 will be increasing), for any value of y1.

We can put a corner arrow feature in the quadrants of the phase plane to remind ourselves of what we have concluded so far.

(%i82)
```
wxdraw2d (xlabel = "y1", ylabel = "y2", xrange = [0, 8], yrange = [0, 6],
    key_pos = top_center,key = "dy2/dt = 0",  explicit (3, x, 0, 8),
    color = red, key = "dy1/dt = 0",    parametric (4, yy, yy, 0, 6),
    color = black, key = "", head_length = 0.3,vector ([0.5,5.5],[1,0]),
    vector ([0.5,5.5], [0, -1]), vector ([7.5,5.5],[-1,0]),  vector ([7.5,5.5], [0, -1]),
    vector ([0.5,0.5],[0,1]), vector ([0.5,0.5], [1, 0]),
    vector ([7.5,0.5],[-1,0]),  vector ([7.5,0.5], [0, 1]))$
```

(%t82) 

To draw convergent system point motion in the lower quadrants, we use the standard form of a parabola having vertex at (x0, y0), having axis of symmetry be the y-axis (vertical) and opening down: y = - a*(x - x0)^2 + y0, with a > 0, so we take y = - 0.2*(x-4)^2+3, and we place a small arrow at x = (2 and 6), y = 2.2, since:

(%i83) `at (-0.2*(x-4)^2+3, x = 2);`

(%o83) 2.2

(%i84) `at (-0.2*(x-4)^2+3, x = 6);`

(%o84) 2.2

Likewise we use a similar parabola formula in the two upper quadrants, but replace -0.2 with 0.2 so the parabola opens in the upward direction. Note that:

(%i85) `at (0.2*(x-4)^2+3, x = 2);`

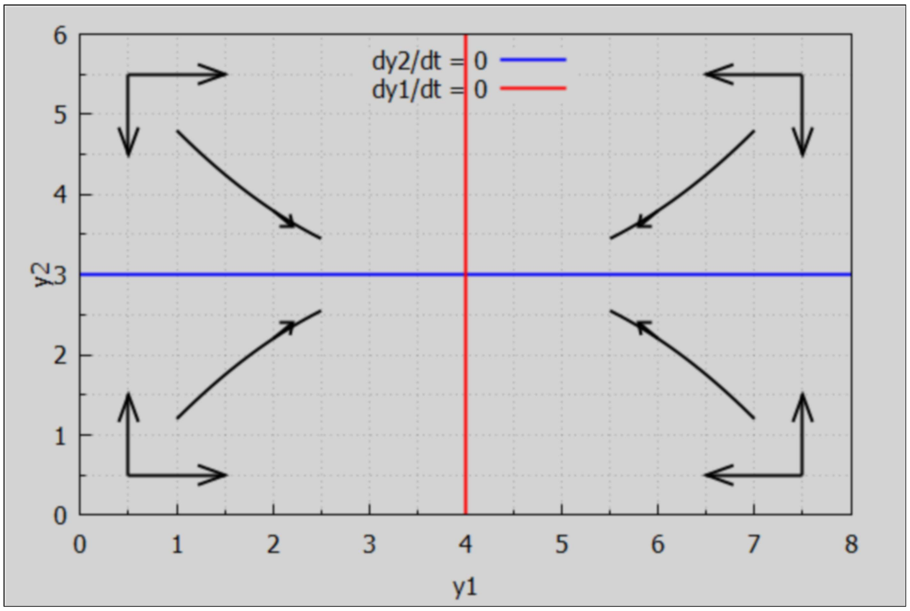
(%o85) 3.8

(%i86) `at (0.2*(x-4)^2+3, x = 6);`

(%o86) 3.8

(%i87)
```
wxdraw2d (xlabel = "y1", ylabel = "y2", xrange = [0, 8], yrange = [0, 6],
    key_pos = top_center,key = "dy2/dt = 0",  explicit (3, x, 0, 8),
    color = red, key = "dy1/dt = 0",    parametric (4, yy, yy, 0, 6),
    color = black, key = "", head_length = 0.3,vector ([0.5,5.5],[1,0]),
    vector ([0.5,5.5], [0, -1]), vector ([7.5,5.5],[-1,0]),  vector ([7.5,5.5], [0, -1]),
    vector ([0.5,0.5],[0,1]), vector ([0.5,0.5], [1, 0]),
    vector ([7.5,0.5],[-1,0]),  vector ([7.5,0.5], [0, 1]),
    explicit (- 0.2*(x - 4)^2 + 3, x, 1, 2.5), vector([2,2.2],[0.2,0.2]),
    explicit (- 0.2*(x - 4)^2 + 3, x, 5.5, 7), vector ([6,2.2],[-0.2,0.2]),
    explicit (0.2*(x-4)^2 + 3, x, 1, 2.5), vector ([2, 3.8], [0.2, -0.2]),
    explicit (0.2*(x-4)^2 + 3, x, 5.5, 7), vector ([6, 3.8],[-0.2, -0.2]))$
```

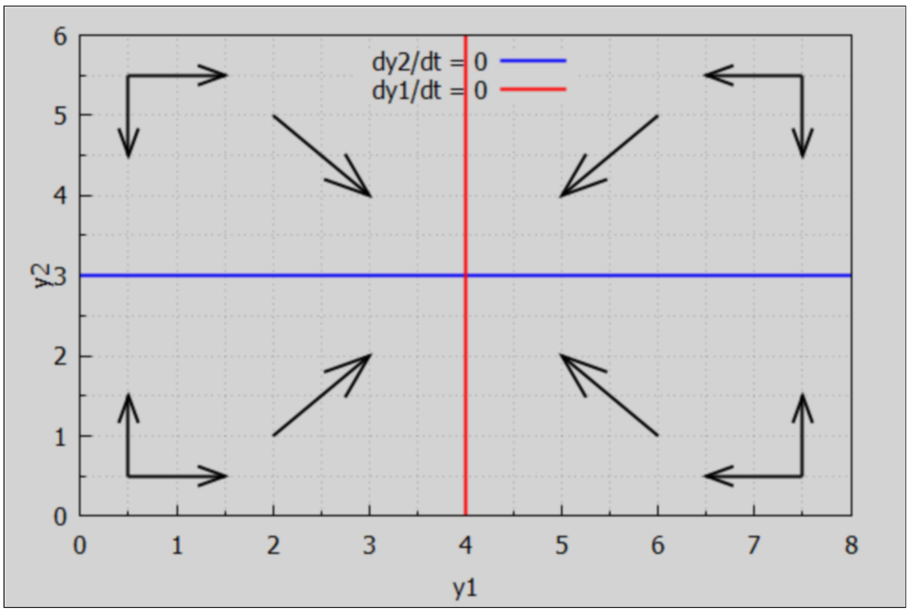
(%t87)



It is, of course, easier to use vector ([x0,y0], [dx,dy]) in each of the quadrants once to indicate the general system motion in that quadrant.

```
(%i88) wxdraw2d (xlabel = "y1", ylabel = "y2", xrange = [0, 8], yrange = [0, 6],
    key_pos = top_center,key = "dy2/dt = 0",  explicit (3, x, 0, 8),
    color = red, key = "dy1/dt = 0",    parametric (4, yy, yy, 0, 6),
    color = black, key = "", head_length = 0.3,
    /* corner vectors */
    vector ([0.5,5.5],[1,0]), vector ([0.5,5.5], [0, -1]),
    vector ([7.5,5.5],[-1,0]),  vector ([7.5,5.5], [0, -1]),
    vector ([0.5,0.5],[0,1]), vector ([0.5,0.5], [1, 0]),
    vector ([7.5,0.5],[-1,0]),  vector ([7.5,0.5], [0, 1]),
    /* quadrant vectors */
    head_length = 0.5, vector ( [2,1],[1,1] ), vector ( [2,5],[1,-1]), vector ([6, 1],[-1, 1]),
    vector ([6,5],[-1,-1]))$
```

(%t88)



Without committing ourselves to initial values and a definite solution, we can formulate Example 9 in a matrix form: dY/dt = A . Y + B, and use the Maxima function eigenvalues with the matrix A.

```
(%i90) A : matrix ([-4, 0], [0, -5]);
       eigenvalues (A);
```

(A) $\begin{pmatrix} -4 & 0 \\ 0 & -5 \end{pmatrix}$

(%o90) [[−5,−4],[1,1]]

Both eigenvalues are real and negative, indicating the complementary solution has the form k1*V1*exp(-5*t) + k2*V2*exp(-4*t), in which V1 and V2 are the corresponding eigenvectors. This complementary solution converges to zero, leaving the equilibrium solution Ye (implied by dY/dt = 0).

(%i92) B : cvec ([16, 15]);
Ye : - invert (A) . B;

(B) $\begin{pmatrix}16\\15\end{pmatrix}$

(Ye) $\begin{pmatrix}4\\3\end{pmatrix}$

So having the matrix A determines the eigenvalues r1 and r2, leading to general answers about whether or not the system will reach the nominal equilibrium values Ye (implied by A and B).

Even if we have no available method to solve for the eigenvalues of a pair of ode's, we can use the phase plane analysis to predict stability or instability of a given model.

7.2 Example 10: Saddle Point Equilibrium Case

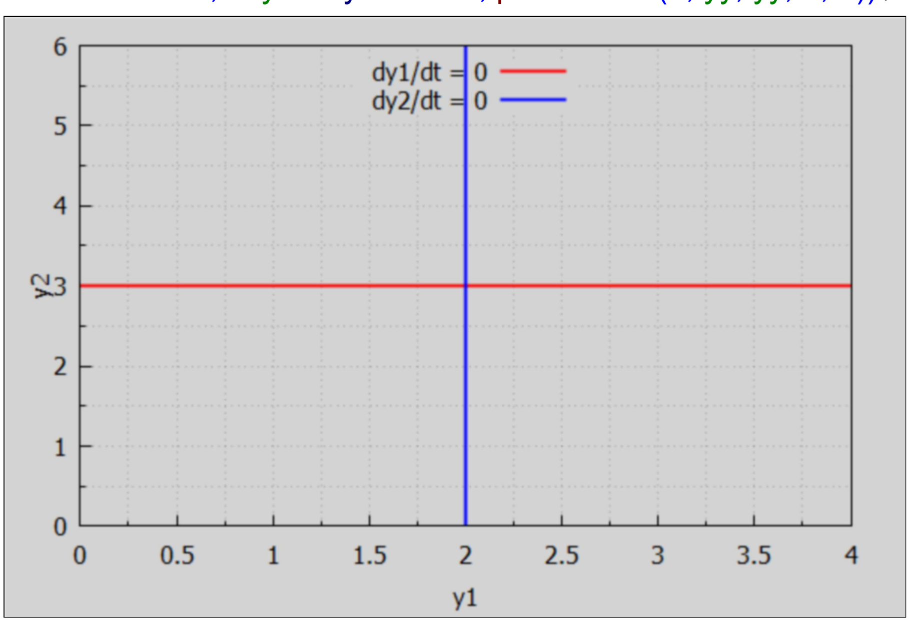
Given the system of linear autonomous differential equations
dy1/dt = 2 y2 - 6,
dy2/dt = 8 y1 - 16,
a phase diagram is used below to test the stability of the model.

1. Determine the isoclines from the given pair of ode's.

The y1 isocline is the horizontal line y2 = 3; on this line dy1/dt = 0.
The y2 isocline is the vertical line y1 = 2; on this line dy2/dt = 0.

(%i93) wxdraw2d (xlabel = "y1", ylabel = "y2", xrange = [0, 4], yrange = [0, 6], key_pos = top_center,
color = red, key = "dy1/dt = 0", explicit (3, x, 0, 4),
color = blue, key = "dy2/dt = 0", parametric (2, yy, yy, 0, 6))$

(%t93)



2. Determine the motion around the y1 isocline, on which dy1/dt = 0.
Recall the given odes: dy1/dt = 2 y2 - 6, dy2/dt = 8 y1 - 16,

Above the y1 isocline, y2 > 3, and dy1/dt > 0, so y1 increases with time t, thus arrows of motion point to the right.

Below the y1 isocline, y2 < 3, and dy1/dt < 0, so y1 decreases with time, arrows of motion point to the left.

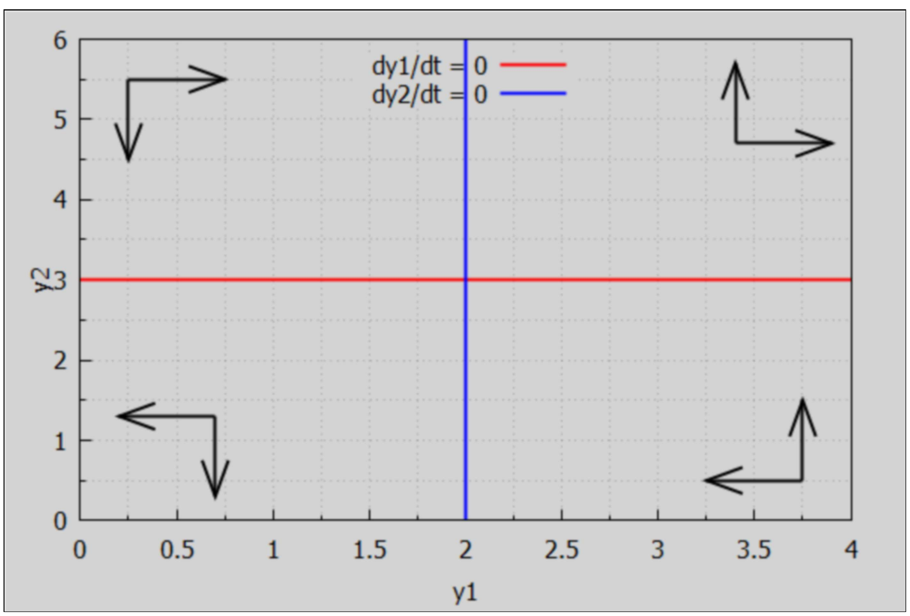
3. Determine the motion around the y2 isocline, on which dy2/dt = 0.

To the left of the y2 isocline, y1 < 2, dy2/dt < 0, y2 is decreasing with time, arrows of motion point downward.
To the right of the y2 isocline, y1 > 2, dy2/dt > 0, y2 is increasing with time, arrows of motion point upward.

4. Show this system motion with quadrant arrows of motion.

(%i94)
```
wxdraw2d (xlabel = "y1", ylabel = "y2", xrange = [0, 4], yrange = [0, 6], key_pos = top_center,
   color = red, key = "dy1/dt = 0", explicit (3, x, 0, 4),
   color = blue, key = "dy2/dt = 0", parametric (2, yy, yy, 0, 6),
   color = black, key = "", head_length = 0.2,
   vector ([0.25,5.5],[0.5, 0]), vector ([0.25,5.5], [0, -1]),
   vector ([3.4,4.7],[0.5,0]),  vector ([3.4,4.7], [0, 1]),
  /* vector ([0.5, 1],[- 0.9,0]), vector ([0.5, 1], [0, - 1]), */
   vector ( [0.7, 1.3], [- 0.5, 0] ), vector ( [0.7, 1.3], [0, - 1]),
   vector ([3.75,0.5],[-0.5,0]),  vector ([3.75,0.5], [0, 1]))$
```

(%t94)



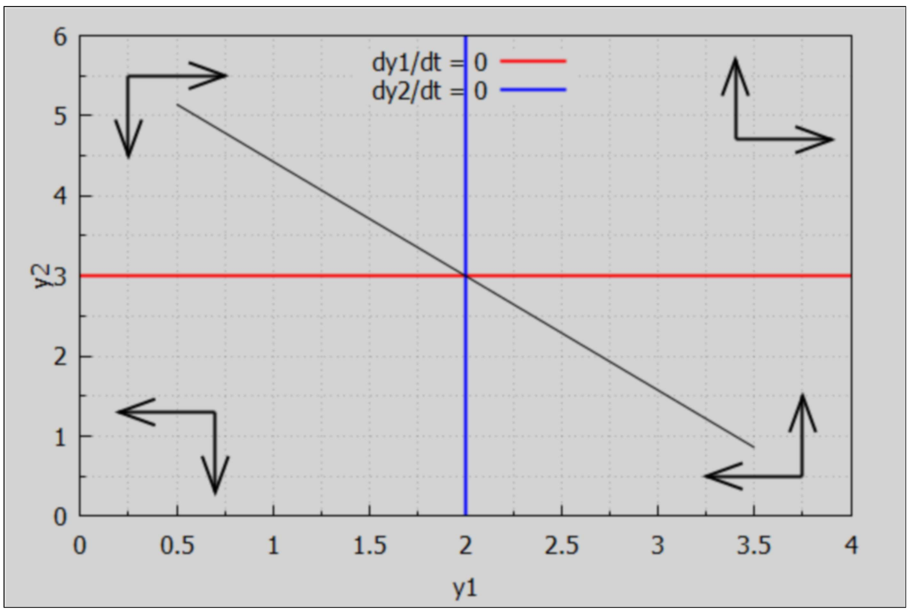
Add a diagonal line.

(%i95) solve ([5.5 = a*0.25 + b, 0.5 = a*3.75 + b], [a, b]);

(%o95) $[[a=-\frac{10}{7},b=\frac{41}{7}]]$

(%i96)
```
wxdraw2d (xlabel = "y1", ylabel = "y2", xrange = [0, 4], yrange = [0, 6], key_pos = top_center,
   color = red, key = "dy1/dt = 0", explicit (3, x, 0, 4),
   color = blue, key = "dy2/dt = 0", parametric (2, yy, yy, 0, 6),
   color = black, key = "", head_length = 0.2,
   vector ([0.25,5.5],[0.5, 0]), vector ([0.25,5.5], [0, -1]),
   vector ([3.4,4.7],[0.5,0]),  vector ([3.4,4.7], [0, 1]),
  /* vector ([0.5, 1],[- 0.9,0]), vector ([0.5, 1], [0, - 1]), */
   vector ( [0.7, 1.3], [- 0.5, 0] ), vector ( [0.7, 1.3], [0, - 1]),
   vector ([3.75,0.5],[-0.5,0]),  vector ([3.75,0.5], [0, 1]),
   line_width = 1, explicit (- 10*x/7 + 41/7, x, 0.5, 3.5))$
```

(%t96)



We use two parabolas opening up and down, having a vertical axis coinciding with the line x = 2, one with vertex at (x0 = 2, y0 = 4) opening up, using the standard form
   y = a*(x-x0)^2 + y0 with a > 0.
We use the same formula for a parabola opening down with vertex at (x0 = 2, y0 = 2) and with a < 0.

(%i97) at (0.8*(x-2)^2 + 4, x = 3);

(%o97) 4.8

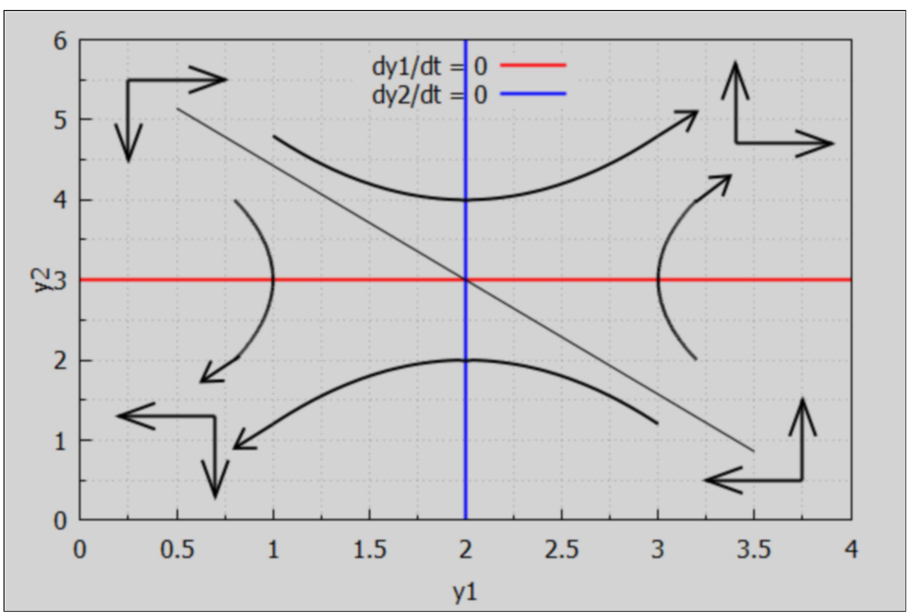
(%i98) at (-0.8*(x - 2)^2 + 2, x = 1);

(%o98) 1.2

We use two more parabolas having the standard form
x = a*(y - y0)^2 + x0
with vertex at (x0, y0), opening to the right for a > 0, opening to the left for a < 0.
To plot these two curves we use the draw2d function
implicit (f(x,y), x, x1, x2, y, y1, y2),
in which f (x,y) is either an expression depending on x and y, or an equation depending on x and y.

To place a small vector on the ends of these latter two curves, we right click the plot, select "popout interactively", and expand the result to full screen. We can then read off the coordinates of the cursor position on the screen in the lower left corner. Those numbers become a first approximation to the numbers (xs, ys) in vector ([xs, ys], [dx, dy]). Some trial and error is needed to get a reasonable looking arrow (and it will be clear we didn't spend enough time on this!)

```
(%i99) wxdraw2d (xlabel = "y1", ylabel = "y2", xrange = [0, 4], yrange = [0, 6], key_pos = top_center,
    color = red, key = "dy1/dt = 0", explicit (3, x, 0, 4),
    color = blue, key = "dy2/dt = 0", parametric (2, yy, yy, 0, 6),
    color = black, key = "", head_length = 0.2,
    /* corner vectors */
    vector ([0.25,5.5],[0.5, 0]), vector ([0.25,5.5], [0, -1]),
    vector ([3.4,4.7],[0.5,0]),  vector ([3.4,4.7], [0, 1]),
    vector ( [0.7, 1.3], [- 0.5, 0] ), vector ( [0.7, 1.3], [0, - 1]),
    vector ([3.75,0.5],[-0.5,0]),  vector ([3.75,0.5], [0, 1]),
    /* diagonal line */
    line_width = 1, explicit (- 10*x/7 + 41/7, x, 0.5, 3.5),
    /*  four system motion curves  */
    line_width = 2, explicit (0.8*(x-2)^2 + 4, x, 1, 3), vector ([3,4.8], [0.2,0.3]),
    explicit (-0.8*(x - 2)^2 + 2, x, 1,3), vector ([1,1.2], [-0.2,-0.3]),
    implicit (x = 0.2*(y-3)^2 + 3, x, 2.8, 3.5, y, 2, 4), vector ([3.193,3.97],[0.18,0.33]),
    implicit (x = - 0.2*(y - 3)^2 + 1, x, 0.5, 1.2, y, 2, 4), vector ([0.828,2.05],[-0.2,-0.32]))$
```

(%t99) 

We see that the system is unstable, irregardless of in which of the four quadrants the system point (y1(t), y2(t)) is located at t = 0, even in the NorthWest and SouthEast quadrants.

Without committing ourselves to initial values and a definite solution, we can formulate Example 10 in a matrix form: dY/dt = A . Y + B, and use the Maxima function eigenvalues with the matrix A and also calculate Ye using A and B.

```
(%i103) A : matrix ([0, 2], [8, 0]);
        B : cvec ([-6, -16]);
        Ye : - invert (A) . B;
        eigenvalues (A);
```

(A) $\begin{pmatrix} 0 & 2 \\ 8 & 0 \end{pmatrix}$

(B) $\begin{pmatrix} -6 \\ -16 \end{pmatrix}$

(Ye) $\begin{pmatrix} 2 \\ 3 \end{pmatrix}$

(%o103) [[−4,4],[1,1]]

The eigenvalues are +/- 4, indicating the complementary solution has the form
k1*V1*exp(4*t) + k2*V2*exp(-4*t),
in which V1 and V2 are the corresponding eigenvectors. This complementary solution grows exponentially large for 4*t >> 1.

The diagonal line in our figure is a "saddle path". Only if the initial conditions fall on the saddle path will the steady-state equilibrium prove to be stable.

Quoting Dowling again on this point:

"A 'saddle point equilibrium', in which the roots have opposite signs, will usually be unstable. An exception to the latter rule is the case in which the initial conditions y10 and y20 satisfy
y20 = (r1 - a11)*(y10 - y1e)/a12 + y2e,
where r1 = the negative root, and we have what is called a 'saddle path', y10 and y20 happen to be on the saddle path, and y1(t) and y2(t) will then converge to their intertemporal equilibrium level (see Example 10)."

In Ex. 10 the negative root is r1 - -4, y1e = 2, y2e = 3, A[1,1] = a11 = 0, A[1,2] = a12 = 2, so we need y2(0) = 7 - 2*y1(0) initial conditions.

Let's try y10 = 3, so we need y20 = 7 - 2*3 = 1, in the SouthEast quadrant.

==============================
===============================

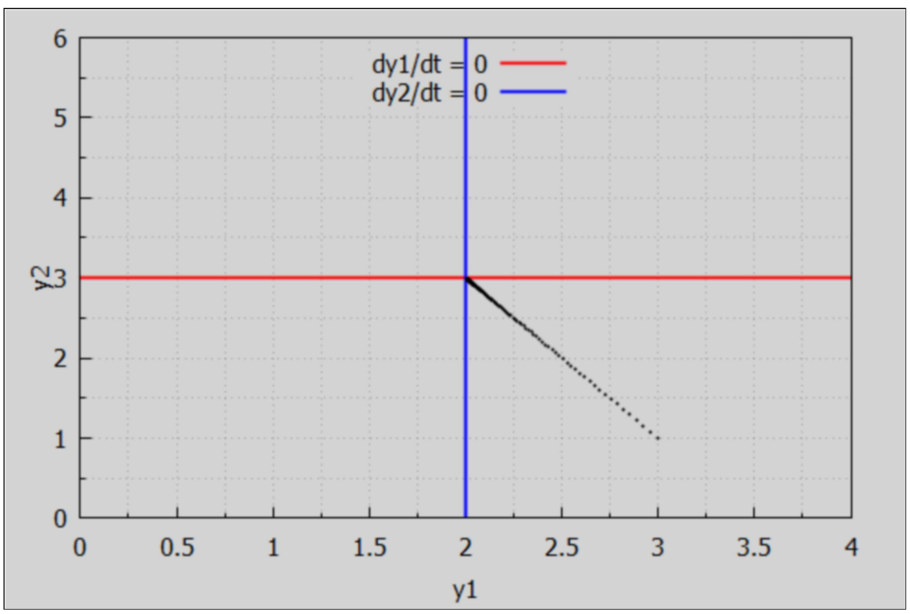
(%i106)
```
kill(y1,y2)$
R : rk ([2*y2 - 6, 8*y1 - 16], [y1, y2], [3, 1], [t, 0, 1.2, 0.01])$
fll(R);
```

(%o106) [[0.0,3.0,1.0],[1.2,2.0082,2.9835],121]

(%i108)
```
y1y2_pts : makelist ([R[j][2], R[j][3] ], j, 1, length (R))$
fll (y1y2_pts);
```

(%o108) [[3.0,1.0],[2.0082,2.9835],121]

(%i109)
```
wxdraw2d (xlabel = "y1", ylabel = "y2", xrange = [0, 4], yrange = [0, 6], key_pos = top_center,
    color = red, key = "dy1/dt = 0", explicit (3, x, 0, 4),
    color = blue, key = "dy2/dt = 0", parametric (2, yy, yy, 0, 6),
    key = "", color = black,  point_size = 0.2,  points (y1y2_pts) )$
```

(%t109) 

This plot of our numerical integration of the pair of ode's shows that by choosing y10 and y20 carefully, we get a convergent solution.

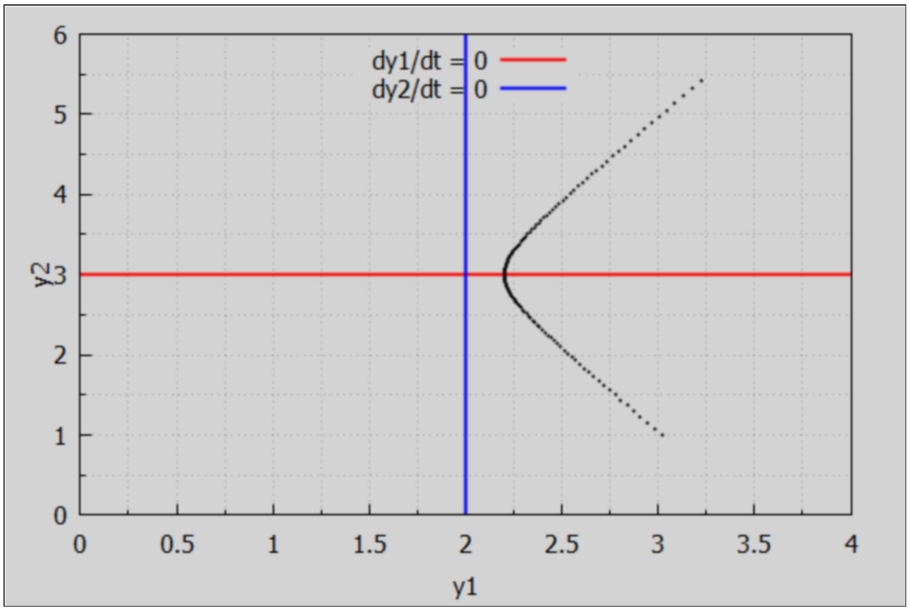
Let's choose a slightly different starting point for the system, with the same set of ode's.

(%i112)
```
kill(y1,y2)$
R : rk ([2*y2 - 6, 8*y1 - 16], [y1, y2], [3.02, 1], [t, 0, 1.2, 0.01])$
fll(R);
```

(%o112) [[0.0,3.02,1.0],[1.2,3.2234,5.4136],121]

```
(%i114) y1y2_pts : makelist ([R[j][2], R[j][3] ], j, 1, length (R))$
        fll (y1y2_pts);
(%o114) [[3.02,1.0],[3.2234,5.4136],121]

(%i115) wxdraw2d (xlabel = "y1", ylabel = "y2", xrange = [0, 4], yrange = [0, 6], key_pos = top_center,
          color = red, key = "dy1/dt = 0", explicit (3, x, 0, 4),
          color = blue, key = "dy2/dt = 0", parametric (2, yy, yy, 0, 6),
          key = "", color = black,  point_size = 0.2,  points (y1y2_pts) )$
```

(%t115)



which shows that a small fluctuation in initial conditions results in a non-convergent model.

8 *Solutions for A1 . dY/dt = A2 . Y(t) + B Using desolve*

We consider Dowling's example, Sec. 19.2, Ex. 3, page 432,

dy1/dt = - 3 y1 + 1.5 y2- 2.5 dy2/dt + 2.4,

dy2/dt = 2 y1 - 5 y2 + 16,

with initial conditions y1(0) = 14, y2(0) = 15.4, which Dowling solves using matrix methods.

We first use desolve, which doesn't require any matrix considerations.

(%i120) atvalue (y1(t), t = 0, 14)$
atvalue (y2(t), t = 0, 15.4)$
eqn1 : diff (y1(t), t) = - 3*y1(t) + 1.5*y2(t) - 2.5*diff (y2(t), t) + 2.4;
eqn2 : diff (y2(t), t) = 2*y1(t) - 5*y2(t) + 16;
soln : desolve ([eqn1, eqn2], [y1(t), y2(t)]);

(eqn1) $\frac{d}{d\,t}\,y1(t)=-2.5\left(\frac{d}{d\,t}\,y2(t)\right)+1.5\,y2(t)-3\,y1(t)+2.4$

(eqn2) $\frac{d}{d\,t}\,y2(t)=-5\,y2(t)+2\,y1(t)+16$

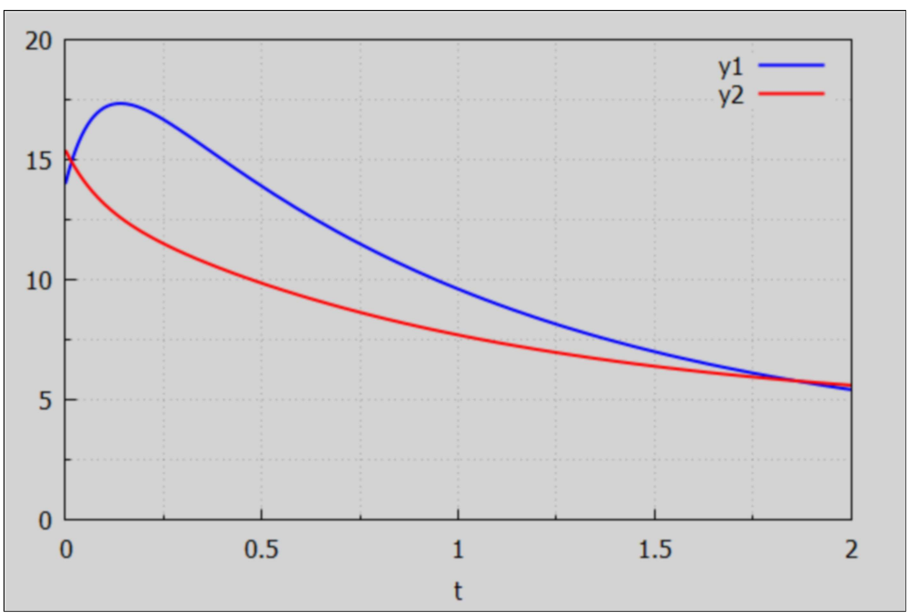
(soln) $[y1(t)=18\,\%e^{-t}-7\,\%e^{-12\,t}+3,y2(t)=9\,\%e^{-t}+2\,\%e^{-12\,t}+\frac{22}{5}]$

(%i121) [y1, y2] : map ('rhs, %);

(%o121) $[18\,\%e^{-t}-7\,\%e^{-12\,t}+3,9\,\%e^{-t}+2\,\%e^{-12\,t}+\frac{22}{5}]$

It is clear from the exponential damping in the above solutions, that in the long run, y1 ~ 3 and y2 ~ 22.5 = 4.4.

(%i123) tmax : 2$
wxdraw2d (xlabel = "t", yrange = [0, 20],
 key = "y1",
   explicit (y1, t, 0, tmax), color = red, key = "y2",
   explicit (y2, t, 0, tmax));

(%t123) 

(%o123)

(%i124) t1max : find_root (diff (y1, t), t, 0.01, 0.25);

(t1max) 0.14004

```
(%i125) y1max : at (y1, t = t1max);
(y1max) 17.344
```

Noting that 22/5 = 4.4, this is the solution Dowling gets for Ex. 3 on pages 432-434 by hand using matrix methods.

9 Solutions for A1 . dY/dt = A2 . Y(t) + B Using rk Methods

We consider Dowling's example, Sec. 19.2, Ex. 3, page 432,
dy1/dt = - 3 y1 + 1.5 y2- 2.5 dy2/dt + 2.4,

dy2/dt = 2 y1 - 5 y2 + 16,

with initial conditions y1(0) = 14, y2(0) = 15.4, which Dowling solves using matrix methods.

By substituting the rhs of dy2/dt (line two) for dy2/dt in line one, we get the pair of first order ODE's
dy1/dt = - 8 y1 + 14 y2 - 37.6,

dy2/dt = 2 y1 - 5 y2 + 16,
and use the rk Maxima numerical Runge_Kutta integration routine:

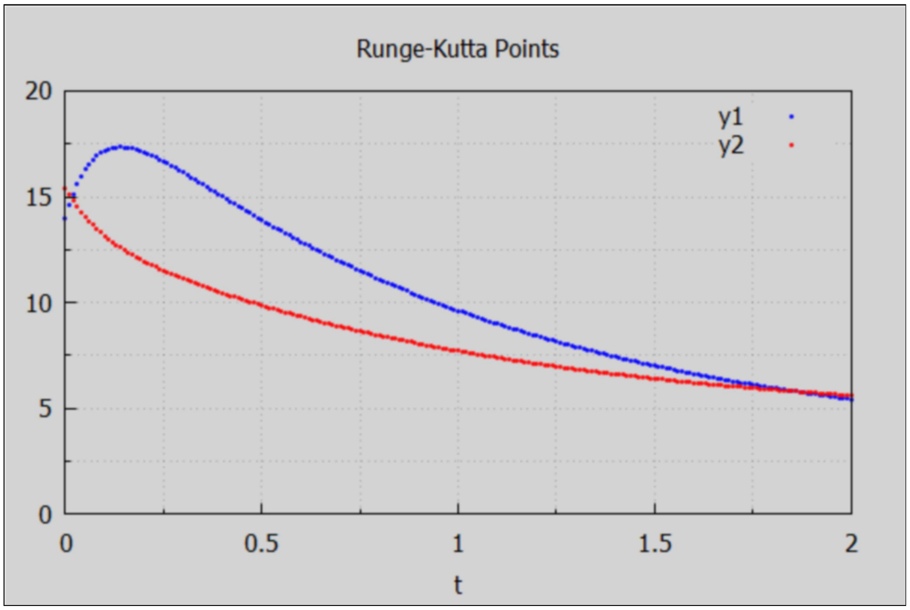
```
(%i127) kill (y1, y2)$
        results : rk ( [ - 8*y1 + 14*y2 - 37.6,  2*y1 - 5*y2 + 16], [y1, y2], [14, 15.4], [t, 0, 2, 0.01])$
```

For long lists, it is convenient to use fll(L), defined in Econ2.mac, which shows you the first element of the list, the last element of the list, and the length of the list.

```
(%i128) fll (results);
(%o128) [[0.0,14.0,15.4],[2.0,5.436,5.618],201]

(%i129) ty1_pts : makelist ([ results[j][1], results[j][2] ], j, 1, length (results) )$

(%i130) fll (ty1_pts);
(%o130) [[0.0,14.0],[2.0,5.436],201]

(%i131) ty2_pts : makelist ([ results[j][1], results[j][3] ], j, 1, length (results) )$

(%i132) fll (ty2_pts);
(%o132) [[0.0,15.4],[2.0,5.618],201]
```

```
(%i133) wxdraw2d (xlabel = "t", yrange = [0, 20],
        title = "Runge-Kutta Points", key = "y1",
        point_size = 0.3, points (ty1_pts),  color = red, key = "y2",
        points (ty2_pts))$
```

(%t133)



10 Solutions for A1 . dY/dt = A2 . Y(t) + B Using Matrix Methods

We use the same example used above with desolve and rk, but define the 2 x 2 matrices A1 and A2, and the matrix 2 element column vector B, with Y = cvec ([y1, y2]) the unknown solution, to conform to the matrix equation A1 . dY/dt = A2 . Y(t) + B.

We then call the Maxima function ODE2S (A1, A2, B, Y0), defined in Econ2.mac, which uses the matrix inverse to reduce the given pair of ode's to the form dY/dt = D . Y + E.

```
(%i138) A1 : matrix ([1, 2.5], [0, 1]);
        A2 : matrix ([-3, 1.5], [2, -5]);
        B : cvec ([2.4, 16]);
        Y0 : cvec ([14, 15.4]);
        Ye : - invert(A2) . B;
```

(A1) $\begin{pmatrix} 1 & 2.5 \\ 0 & 1 \end{pmatrix}$

(A2) $\begin{pmatrix} -3 & 1.5 \\ 2 & -5 \end{pmatrix}$

(B) $\begin{pmatrix} 2.4 \\ 16 \end{pmatrix}$

(Y0) $\begin{pmatrix} 14 \\ 15.4 \end{pmatrix}$

(Ye) $\begin{pmatrix} 3.0 \\ 4.4 \end{pmatrix}$

```
(%i139) Ys : ODE2S(A1, A2, B, Y0);
```

(Ys) $\begin{pmatrix} 18\ \%e^{-t} - 7\ \%e^{-12\ t} + 3.0 \\ 9\ \%e^{-t} + 2\ \%e^{-12\ t} + 4.4 \end{pmatrix}$

lme is our alias for list_matrix_entries (defined in Econ2.mac).

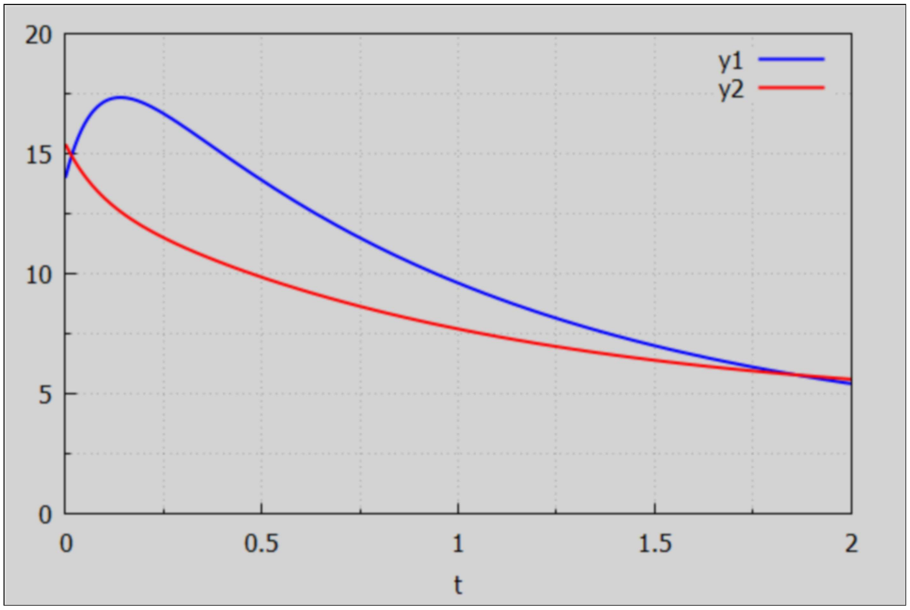
```
(%i140) [y1, y2] : lme (Ys);
```

(%o140) $[18\ \%e^{-t} - 7\ \%e^{-12\ t} + 3.0, 9\ \%e^{-t} + 2\ \%e^{-12\ t} + 4.4]$

These solutions agree with our previous results with this problem.

```
(%i142) tmax : 2$
        wxdraw2d (xlabel = "t", yrange = [0, 20],
         key = "y1",
           explicit (y1, t, 0, tmax), color = red, key = "y2",
           explicit (y2, t, 0, tmax));
```

(%t142)



(%o142)

10.1 Dowling Prob 19.4: Example of using ODE2S (A1, A2, B,Y0)

Solve the following system of nonlinear, autonomous, first-order differential equations in which one or more derivative is a function of another derivative.

dy1/dt = 4 y1 + y2 + 6,
dy2/dt = 8 y1 + 5 y2 - dy1/dt - 6,
with
y1(0) = 9, y2(0) = 10.

Define matrices A1, A2, and B, such that in matrix form we have
A1 . dY/dt = A2 . Y(t) + B.

```
(%i148) A1 : matrix ([1, 0], [1, 1]);
        A2 : matrix ([4, 1], [8, 5]);
        B : cvec ([6, -6]);
        Y0 : cvec ([9, 10]);
        Ye : - invert (A2) . B;
        Ys : ODE2S (A1, A2, B, Y0);
```

(A1) $\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$

(A2) $\begin{pmatrix} 4 & 1 \\ 8 & 5 \end{pmatrix}$

(B) $\begin{pmatrix} 6 \\ -6 \end{pmatrix}$

(Y0) $\begin{pmatrix} 9 \\ 10 \end{pmatrix}$

(Ye) $\begin{pmatrix} -3 \\ 6 \end{pmatrix}$

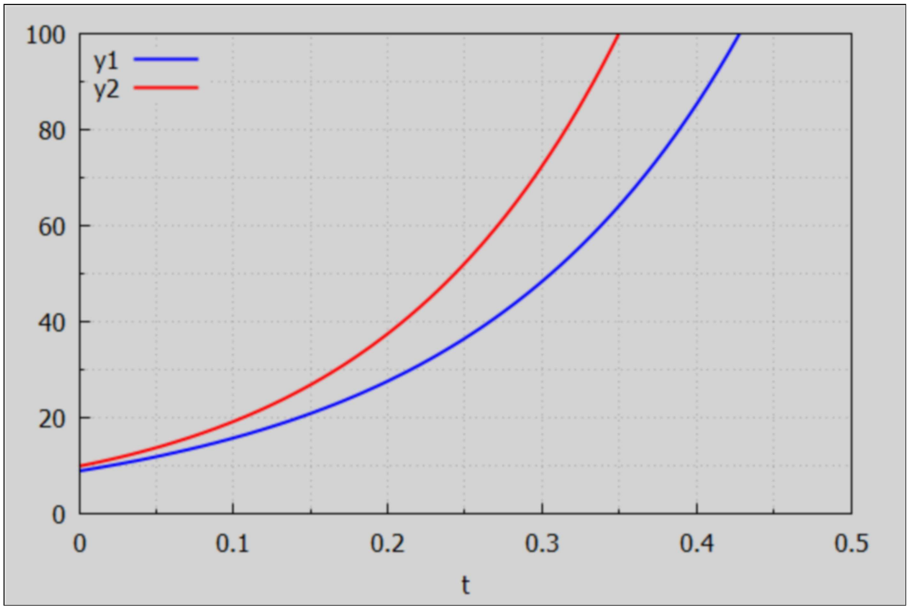
(Ys) $\begin{pmatrix} 7\ \%e^{6\,t}+5\ \%e^{2\,t}-3 \\ 14\ \%e^{6\,t}-10\ \%e^{2\,t}+6 \end{pmatrix}$

```
(%i149) [y1, y2] : lme (Ys);
```

(%o149) $[7\ \%e^{6\,t}+5\ \%e^{2\,t}-3, 14\ \%e^{6\,t}-10\ \%e^{2\,t}+6]$

(%i151) tmax : 0.5$
```
wxdraw2d (xlabel = "t", yrange = [0, 100],
key_pos = top_left, key = "y1",
   explicit (y1, t, 0, tmax), color = red, key = "y2",
   explicit (y2, t, 0, tmax));
```

(%t151)



(%o151)

10.2 Dowling Prob 19.5

Solve the following system of nonlinear, autonomous, first-order differential equations in which one or more derivative is a function of another derivative.

dy1/dt = - y1 + 4 y2 - 0.5 dy2/dt - 1,
dy2/dt = 4 y1 - 2 y2 - 10,
with
y1(0) = 4.5, y2(0) = 16.

```
(%i157) A1 : matrix ([1, 0.5], [0, 1]);
        A2 : matrix ([-1, 4], [4, -2]);
        B : cvec ([- 1, - 10]);
        Y0 : cvec ([4.5, 16]);
        Ye : - invert (A2) . B;
        Ys : ODE2S (A1, A2, B, Y0);
```

(A1) $\begin{pmatrix} 1 & 0.5 \\ 0 & 1 \end{pmatrix}$

(A2) $\begin{pmatrix} -1 & 4 \\ 4 & -2 \end{pmatrix}$

(B) $\begin{pmatrix} -1 \\ -10 \end{pmatrix}$

(Y0) $\begin{pmatrix} 4.5 \\ 16 \end{pmatrix}$

(Ye) $\begin{pmatrix} 3 \\ 1 \end{pmatrix}$

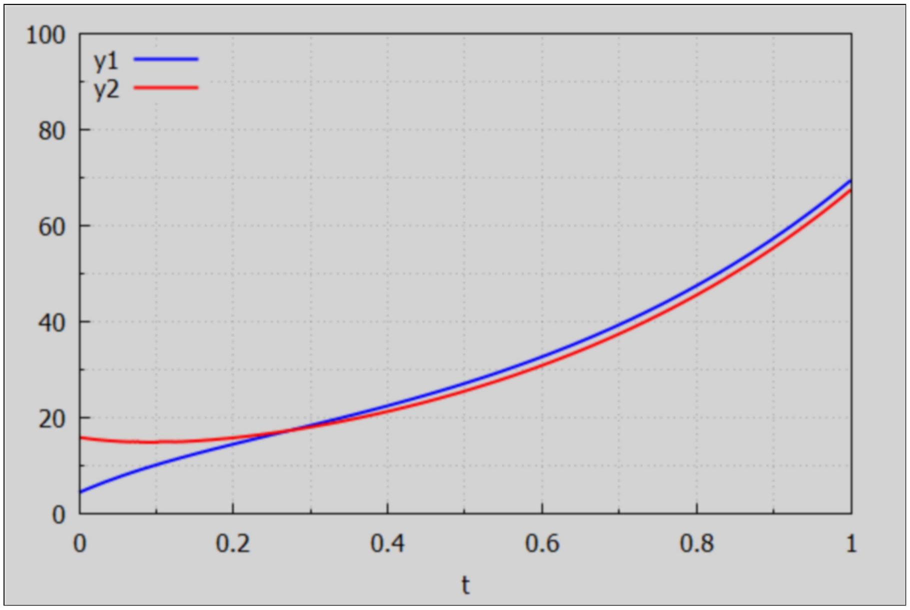
(Ys) $\begin{pmatrix} 9\,\%e^{2\,t} - \frac{15\,\%e^{-7\,t}}{2} + 3.0 \\ 9\,\%e^{2\,t} + 6\,\%e^{-7\,t} + 1.0 \end{pmatrix}$

```
(%i158) [y1, y2] : lme (Ys);
```

(%o158) $[9\,\%e^{2\,t} - \frac{15\,\%e^{-7\,t}}{2} + 3.0, 9\,\%e^{2\,t} + 6\,\%e^{-7\,t} + 1.0]$

```
(%i160) tmax : 1$
wxdraw2d (xlabel = "t", yrange = [0, 100],
key_pos = top_left, key = "y1",
   explicit (y1, t, 0, tmax), color = red, key = "y2",
   explicit (y2, t, 0, tmax));
```

(%t160)



(%o160)

11 Inflation - Unemployment Model as a Set of ODE's

In our Dowling18C.wxmx work we considered both the continuous time and discrete time models of the interaction between inflation and unemployment. The continuous time treatment was based on Chiang and Wainwright's Ch. 16, Sec. 5, which combined a set of three equations (including two ode's) into one second order ode in order to get a solution. The process of arriving at a soluble second order ode was somewhat painful.

Here we follow Chiang and Wainwright's Ch. 19, Sec. 4, "The Inflation-Unemployment Model Once More", in a relatively painless method.

With T the constant labor productivity, our equation relating the rate of inflation p, unemployment U, and inflation expectation π is:

$p(t) = \alpha - T - \beta*U(t) + g*\pi(t), \quad (0 < g <= 1), \quad (\alpha, \beta > 0).$ (1)

The "adaptive expectations hypothesis" is the equation

$d\pi/dt = j*(p - \pi), \quad (0 < j <= 1),$ (2)

which simply says that if the present rate of inflation is greater than the present expected rate of inflation, the expected rate of inflation should increase with time. Likewise, if p falls short of π, then π is revised in the downward direction. The driving force is the difference between the actual and the expected rate of inflation.

Let M(t) be the nominal money balance (the amount of money in the economy) and let μ(t) = (1/M)*dM/dt be the "rate of growth of the money balance". A simple model which relates unemployment U, inflation p, and μ is:

dU/dt = -k*(μ - p), (k > 0). (3)

This relation says that if the money growth rate is larger than the inflation rate, then unemployment will decrease with time, and conversely, if the inflation rate is greater than the money growth rate, unemployment will tend to increase. In Dowling 18C we used m instead of μ to represent the rate of growth of the money balance.

The difference (μ - p) is the rate of growth of "real money". Eqn. (3) asserts an "interaction" between the rate of inflation p and the "time path of unemployment U(t)", dU/dt.

If we substitute (1) into (2) and (3), the latter become the pair of equations:

dπ/dt = - j*(1- g)*π(t) - j*β*U(t) + j*(α - T), (4)

dU/dt = k*g*π(t) - k*β*U(t) + k*(α - T - μ). (5)

which we write in our standard matrix form

dY/dt = A . Y + B, (6)

in which Y = cvec ([π, U]) is a matrix column vector whose first element is the expected inflation π and whose second element is the unemployment U. Once we have solutions for π(t) and U(t), we can use the solutions in Eqn (1) and obtain a solution for the actual rate of inflation p(t).

11.1 Numerical Example 1: Complex Roots

A word of warning about our use of the Maxima symbol π in our equations. Inside the wxMaxima graphical user interface (notebook) the Greek symbol π (accessed by first pressing the excape key and then typing the two letters 'pi') is interpreted by the Maxima engine as %pi, which represents the ratio of the perimeter of a circle to its diameter, and is a transcendental number with the approximate value 3.14159...

The use of the symbol π in calculus equations should be avoided because Maxima interprets it as a constant number.

```
(%i165) grind (π)$
diff (π, t);
fpprintprec : 0$
float (π);
fpprintprec : 5$
```

%pi$

```
(%o162) 0
(%o164) 3.141592653589793
```

We consider a numerical example of the above model of inflation-unemployment, using the parameter values β = 3, g = 1, j = 3/4, k = 1/2, μ = 2, α - T = 1/6.
We assume initial values: π(0) = 3 and U(0) = 1.

```
(%i167) A : matrix([-j*(1-g), - j*β], [k*g, -k*β]);
        B : cvec ([ j*(α - T), k*(α - T - μ)]);
```

(A) $$\begin{pmatrix} -(1-g)\,j & -j\,\beta \\ g\,k & -k\,\beta \end{pmatrix}$$

(B) $$\begin{pmatrix} j\,(\alpha-T) \\ k\,(-\mu+\alpha-T) \end{pmatrix}$$

In the list case1 we include all the parameter values except for α - T. We use ratsubst to handle that substitution.

```
(%i168) case1 : [β = 3, g = 1, j = 3/4, k = 1/2, μ = 2];
```

(case1) $$[\beta=3, g=1, j=\frac{3}{4}, k=\frac{1}{2}, \mu=2]$$

```
(%i170) A : at (A, case1);
        B : at (B, case1);
```

(A) $$\begin{pmatrix} 0 & -\frac{9}{4} \\ \frac{1}{2} & -\frac{3}{2} \end{pmatrix}$$

(B) $$\begin{pmatrix} \frac{3\,(\alpha-T)}{4} \\ \frac{\alpha-T-2}{2} \end{pmatrix}$$

```
(%i171) B : ratsubst (1/6, α - T, B);
```

(B) $$\begin{pmatrix} \frac{1}{8} \\ -\frac{11}{12} \end{pmatrix}$$

```
(%i172) Ye : - invert(A) . B;
```

(Ye) $$\begin{pmatrix} 2 \\ \frac{1}{18} \end{pmatrix}$$

(%i173) eigenvalues (A), expand;

(%o173) $[[-\frac{3\,\%i}{4}-\frac{3}{4},\frac{3\,\%i}{4}-\frac{3}{4}],[1,1]]$

We have complex eigenvalues, and the real parts are negative, hence convergence.

(%i174) Y0 : cvec ([3, 1]);

(Y0) $\begin{pmatrix} 3 \\ 1 \end{pmatrix}$

(%i175) Ys : ODE1S (A, B, Y0);

(Ys) $$\begin{pmatrix} -\frac{(11\,\%i-6)\,\%e^{-\frac{(3\,\%i+3)\,t}{4}}}{12}+\frac{(11\,\%i+6)\,\%e^{\frac{(3\,\%i-3)\,t}{4}}}{12}+2 \\ -\frac{(\%i+1)\,(11\,\%i-6)\,\%e^{-\frac{(3\,\%i+3)\,t}{4}}}{36}-\frac{(\%i-1)\,(11\,\%i+6)\,\%e^{\frac{(3\,\%i-3)\,t}{4}}}{36}+\frac{1}{18} \end{pmatrix}$$

We know πs and Us are real numbers, so don't worry about the messy look of these expressions. They could be made to look simpler, as

exp(-3*t/4)*(A1*cos(3*t/4) + A2*sin(3*t/4))

by using the Maxima function demoivre, but we just want to make a plot.

lme is our alias for list_matrix_entries, defined in Econ2.mac.

(%i176) [πs, Us] : lme (Ys);

(%o176) $$[-\frac{(11\,\%i-6)\,\%e^{-\frac{(3\,\%i+3)\,t}{4}}}{12}+\frac{(11\,\%i+6)\,\%e^{\frac{(3\,\%i-3)\,t}{4}}}{12}+2,-\frac{(\%i+1)\,(11\,\%i-6)\,\%e^{-\frac{(3\,\%i+3)\,t}{4}}}{36}-\frac{(\%i-1)\,(11\,\%i+6)\,\%e^{\frac{(3\,\%i-3)\,t}{4}}}{36}+\frac{1}{18}]$$

With our parameter choices, p(t) = 1/6 - 3*U(t) + π(t).

(%i177) ps : 1/6 - 3*Us + πs;

(ps) $$-3\left(-\frac{(\%i+1)\,(11\,\%i-6)\,\%e^{-\frac{(3\,\%i+3)\,t}{4}}}{36}-\frac{(\%i-1)\,(11\,\%i+6)\,\%e^{\frac{(3\,\%i-3)\,t}{4}}}{36}+\frac{1}{18}\right)-\frac{(11\,\%i-6)\,\%e^{-\frac{(3\,\%i+3)\,t}{4}}}{12}+\frac{(11\,\%i+6)\,\%e^{\frac{(3\,\%i-3)\,t}{4}}}{12}+\frac{13}{6}$$

(%i178) at ([πs, Us, ps], t = 0), expand;

(%o178) $$\left[3,1,\frac{1}{6}\right]$$

(%i179) limit(πs, t, inf);

(%o179) 2

(%i180) limit(ps, t, inf);

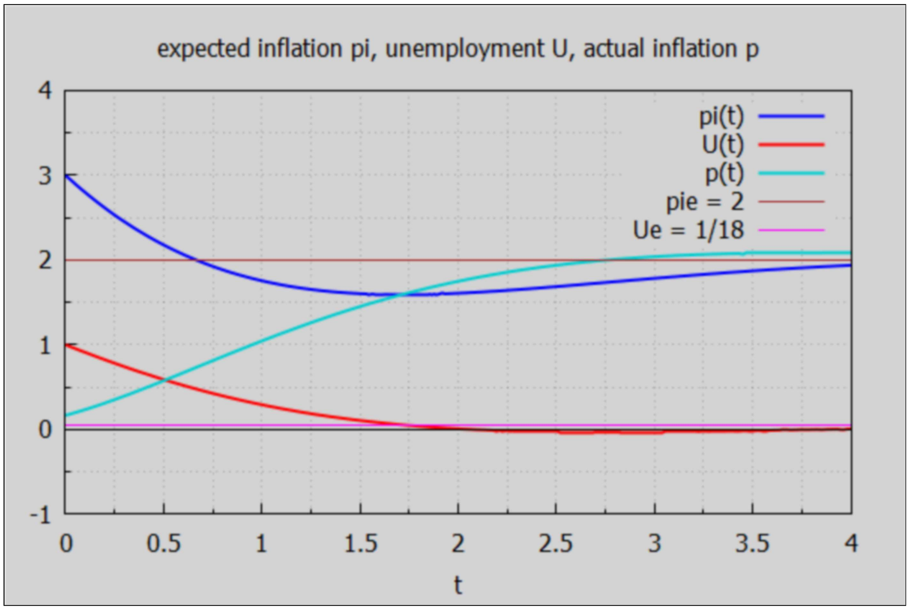
(%o180) 2

(%i181) limit(Us, t, inf);

(%o181) $$\frac{1}{18}$$

```
(%i183) tmax : 4$
        wxdraw2d (xlabel = "t", yrange = [-1, 4],
          title = "expected inflation pi, unemployment U, actual inflation p",
          key = "pi(t)", explicit (πs, t, 0, tmax),
          color = red,  key = "U(t)",  explicit (Us, t, 0, tmax),
          color = dark_turquoise, key = "p(t)", explicit (ps, t, 0, tmax),
          key = "",  color = black, line_width = 1, explicit (0, t, 0, tmax),
          color = brown, key = "pie = 2", explicit (2, t, 0, tmax),
          color = magenta, key = "Ue = 1/18", explicit (1/18, t, 0, tmax))$
```

(%t183)



11.2 Numerical Example 2: Real Roots

As a second numerical example of our same inflation-unemployment model, we take the allowed parameter values: $(\alpha - T) = 1/6$, $\beta = 2$, $g = 1/3$, $j = 1/4$, $k = 1/2$ and $\mu = 2$.

We then have the three starting equations

$$p(t) = 1/6 - 2*U(t) + \pi/3, \qquad (7)$$

$$d\pi/dt = (1/4)*(p(t) - \pi(t)), \qquad (8)$$

$$dU/dt = -(1/2)*(2 - p(t)). \qquad (9)$$

Replacing p(t) from (7) in (8) and (9) results in the pair of first order ode's:

$$d\pi/dt = -\pi/6 - U/2 + 1/24 \qquad (10)$$

$$dU/dt = \pi/6 - U - 11/12. \qquad (11)$$

With Y(t) the matrix column vector cvec ([π(t), U(t)]), the pair of equations (10) and (11) can be written as one matrix equation in the form
dY/dt = A . Y + B.

```
(%i186) A : matrix ( [-1/6, -1/2], [1/6, -1]);
        B : cvec ( [1/24, - 11/12]);
        Ye : - invert (A) . B;
```

(A) $$\begin{pmatrix} -\frac{1}{6} & -\frac{1}{2} \\ \frac{1}{6} & -1 \end{pmatrix}$$

(B) $$\begin{pmatrix} \frac{1}{24} \\ -\frac{11}{12} \end{pmatrix}$$

(Ye) $$\begin{pmatrix} 2 \\ -\frac{7}{12} \end{pmatrix}$$

A little sideshow here:

```
(%i190) depends ([_pi, U], t)$
        Y : cvec ( [_pi, U]);
        diff (Y, t ) = A . Y + B;
        kill (Y, _pi, U)$
```

(Y) $$\begin{pmatrix} \_pi \\ U \end{pmatrix}$$

(%o189) $$\begin{pmatrix} \frac{d}{d\,t}\_pi \\ \frac{d}{d\,t}U \end{pmatrix} = \begin{pmatrix} -\frac{\_pi}{6}-\frac{U}{2}+\frac{1}{24} \\ \frac{\_pi}{6}-U-\frac{11}{12} \end{pmatrix}$$

Back to the main event.

```
(%i191) eigenvalues (A), numer;
(%o191) [[-0.8838,-0.28287],[1,1]]
```

We have two real distinct eigenvalues of A, hence a convergent model.

We assume π(0) = 3 and U(0) = 1 as initial conditions.

(%i192) Y0 : cvec ([3, 1]);

(Y0) $\begin{pmatrix} 3 \\ 1 \end{pmatrix}$

(%i193) Ys : ODE1S (A, B, Y0);

(Ys)

$$\begin{pmatrix} \frac{(9\sqrt{13}+26)\ \%e^{-\frac{(\sqrt{13}+7)\ t}{12}}}{52} - \frac{(9\sqrt{13}-26)\ \%e^{\frac{(\sqrt{13}-7)\ t}{12}}}{52} + 2 \\ \frac{(\sqrt{13}+5)\ (9\sqrt{13}+26)\ \%e^{-\frac{(\sqrt{13}+7)\ t}{12}}}{312} + \frac{(\sqrt{13}-5)\ (9\sqrt{13}-26)\ \%e^{\frac{(\sqrt{13}-7)\ t}{12}}}{312} - \frac{7}{12} \end{pmatrix}$$

The symbol πs is safe to use in an assignment statement.

(%i195) kill(πs, Us, ps)$
grind (πs)$

πs$

lme is our alias for list_matrix_entries, defined in Econ2.mac.

(%i197) [πs, Us] : Ime (Ys);
ps : 1/6 - 2*Us + πs/3;

(%o196) $$\left[\frac{(9\sqrt{13}+26)\,\%e^{-\frac{(\sqrt{13}+7)\,t}{12}}}{52}-\frac{(9\sqrt{13}-26)\,\%e^{\frac{(\sqrt{13}-7)\,t}{12}}}{52}+2,\ \frac{(\sqrt{13}+5)\,(9\sqrt{13}+26)\,\%e^{-\frac{(\sqrt{13}+7)\,t}{12}}}{312}+\frac{(\sqrt{13}-5)\,(9\sqrt{13}-26)\,\%e^{\frac{(\sqrt{13}-7)\,t}{12}}}{312}-\frac{7}{12}\right]$$

(ps) $$-2\left(\frac{(\sqrt{13}+5)\,(9\sqrt{13}+26)\,\%e^{-\frac{(\sqrt{13}+7)\,t}{12}}}{312}+\frac{(\sqrt{13}-5)\,(9\sqrt{13}-26)\,\%e^{\frac{(\sqrt{13}-7)\,t}{12}}}{312}-\frac{7}{12}\right)+\frac{\frac{(9\sqrt{13}+26)\,\%e^{-\frac{(\sqrt{13}+7)\,t}{12}}}{52}-\frac{(9\sqrt{13}-26)\,\%e^{\frac{(\sqrt{13}-7)\,t}{12}}}{52}+2}{3}+\frac{1}{6}$$

Since we are only going to use these numerical solutions to make a plot, we will not attempt any simplifications in πs, Us, or ps. But we can look at values at t = 0 and t --> ∞.

(%i198) at ([πs, Us, ps], t = 0), expand;

(%o198) $\left[3,1,-\frac{5}{6}\right]$

(%i199) limit (πs, t, inf);

(%o199) 2

(%i200) limit (Us, t, inf);

(%o200) $-\frac{7}{12}$

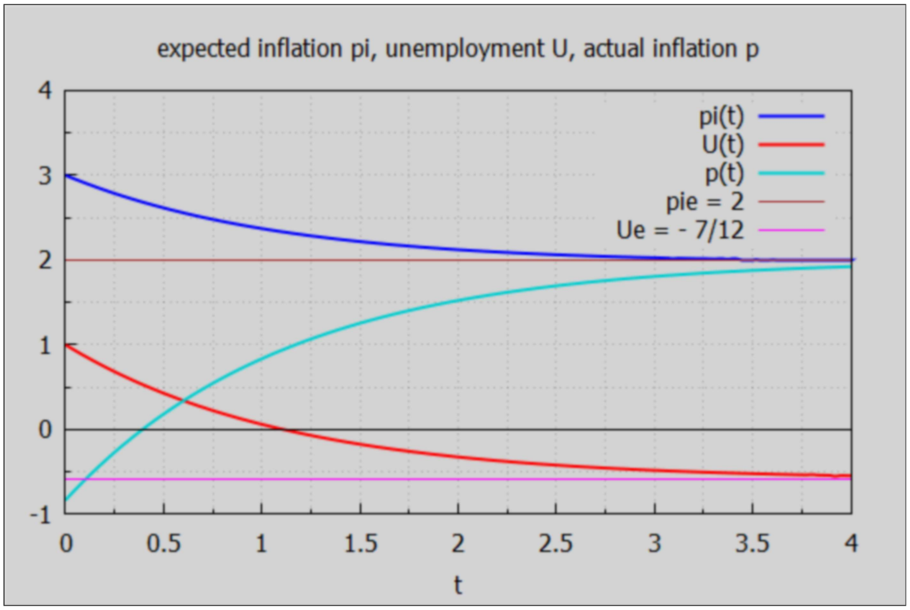
(%i201) limit (ps, t, inf);

(%o201) 2

Both the expected rate of inflation π(t) and the actual rate of inflation p(t) approach μ = 2% in the long run. The unemployment rate U(t) approaches -7/12% ~ - 0.58% in the long run.

```
(%i203) tmax : 4$
        wxdraw2d (xlabel = "t", yrange = [-1, 4],
          title = "expected inflation pi, unemployment U, actual inflation p",
          key = "pi(t)", explicit (πs, t, 0, tmax),
          color = red,  key = "U(t)",  explicit (Us, t, 0, tmax),
          color = dark_turquoise, key = "p(t)", explicit (ps, t, 0, tmax),
          key = "",  color = black, line_width = 1, explicit (0, t, 0, tmax),
          color = brown, key = "pie = 2", explicit (2, t, 0, tmax),
          color = magenta, key = "Ue = - 7/12", explicit (-7/12, t, 0, tmax))$
```

(%t203)



With real negative roots the convergence is steady rather than oscillatory.

11.3 General Proof of Convergence of Inflation-Unemployment Model

```
(%i204) A : matrix([-j*(1-g), - j*β], [k*g, -k*β]);
```

(A) $$\begin{pmatrix} -(1-g)\,j & -j\,\beta \\ g\,k & -k\,\beta \end{pmatrix}$$

```
(%i205) ev : eigenvalues (A);
```

(ev) $$[[-\frac{\sqrt{k^2\,\beta^2+(-2\,g-2)\,j\,k\,\beta+(g^2-2\,g+1)\,j^2}+k\,\beta+(1-g)\,j}{2}, \frac{\sqrt{k^2\,\beta^2+(-2\,g-2)\,j\,k\,\beta+(g^2-2\,g+1)\,j^2}-k\,\beta+(g-1)\,j}{2}],[1,1]]$$

(%i206) [r1, r2] : [ev[1][1], ev[1][2]];

(%o206) $$\left[-\frac{\sqrt{k^2\beta^2+(-2g-2)jk\beta+(g^2-2g+1)j^2}+k\beta+(1-g)j}{2}, \frac{\sqrt{k^2\beta^2+(-2g-2)jk\beta+(g^2-2g+1)j^2}-k\beta+(g-1)j}{2}\right]$$

First look at the sum of the roots: r1 + r2, call it r1pr2.

(%i207) r1pr2 : r1 + r2, ratsimp;

(r1pr2) $(g-1)j-k\beta$

Recall our assumptions about the model parameters:

0 < g <= 1, α, β > 0, 0 < j <= 1, k > 0.

This means k*β > 0, and j*(1 - g) >= 0, so (r1 + r2) < 0 always.

The sum (r1 + r2) must then always be negative.

With the eigenvalue equation taking the form
r^2 + a1*r + a2 = 0,
if we have a case in which a1^2 < 4*a2, r1 and r2 are complex. This means they are complex conjugates of each other, and each has the same real part but their complex parts are opposite in sign. So r1 + r2 = 2*real(r1) < 0, hence the real part is negative, which implies convergence of the complementary solution.

Next look at the product of the roots, call it r1r2.

(%i208) r1r2 : r1*r2, ratsimp;

(r1r2) $jk\beta$

Given our restrictions on the model parameters j, k, and β, r1* r2 is always a positive number.

Assume we have a case in which a1^2 > 4*a2, so that we have two distinct real roots. Since the product of the roots must be positive, they must both have the same sign. Since their sum must always be negative, the two real roots of the same sign must both be negative, and we again have convergence of the complementary solution in the case of distinct real roots.