Fall 2003 Math 308/501–502
9 Matrix Methods for Linear Systems
9.5 Homogeneous Linear Systems with Constant Coefficients
Fri, 14/Nov ©2003, Art Belmonte

Summary

Today we'll take a broad view of how to solve a linear system $\mathbf{x}' = \mathbf{A}\mathbf{x}$ with constant coefficients. That is, the $n \times n$ coefficient matrix $\mathbf{A}$ has constant elements.

Definitions

An **eigenvalue** of $\mathbf{A}$ is a number r such that $\mathbf{A}\mathbf{v} = r\mathbf{v}$ for some nonzero vector $\mathbf{v}$, which is called an **eigenvector** associated with the eigenvalue r. Collectively, r and $\mathbf{v}$ are called an **eigenpair**. Equivalently, we have $(\mathbf{A} - r\mathbf{I})\mathbf{v} = \mathbf{0}$, where $\mathbf{I}$ is the $n \times n$ identity matrix and $\mathbf{0}$ the $n \times 1$ column vector of zeros. Thus $\mathbf{v}$ is a nontrivial vector in nullspace of $\mathbf{A} - r\mathbf{I}$, a.k.a, the **eigenspace** of r, a subspace of $\mathbb{R}^n$ or $\mathbb{C}^n$. Observe that any nonzero multiple of an eigenvector $\mathbf{v}$ is also an eigenvector associated with eigenvalue r.

The **characteristic polynomial** of $\mathbf{A}$ is $p(r) = \det(\mathbf{A} - r\mathbf{I})$. The **characteristic equation** of $\mathbf{A}$ is $p(r) = 0$. Its roots (i.e., the values for which the polynomial is zero) are precisely the eigenvalues of $\mathbf{A}$. These are also known as **characteristic roots**.

Theorem

If r is an eigenvalue of a matrix $\mathbf{A}$ with assoicated eigenvector $\mathbf{v}$, then the function $\mathbf{x}(t) = e^{rt}\mathbf{v}$ is a solution of the system $\mathbf{x}' = \mathbf{A}\mathbf{x}$ satisfying the vector initial condition $\mathbf{x}(0) = \mathbf{v}$.

Roots various

1. If the eigenvalues $r_k,\ k = 1, \ldots, n$, of a matrix $\mathbf{A}$ are *distinct real roots*, then a fundamental solution set for $\mathbf{x}' = \mathbf{A}\mathbf{x}$ is given by
$$\mathbf{x}_k(t) = e^{r_k t}\mathbf{v}_k,\ k = 1, \ldots, n,$$
where $\mathbf{v}_k$ is an eigenvector associated with r_k.
2. If some of the eigenvalues are complex numbers, exponential solutions given by the preceding theorem are complex-valued. We'll learn how to extract real-valued solutions from these in Section 9.6.
3. If the eigenvalues (be they real or complex) are *not* distinct (i.e., there are repeated roots), then this *may* present complications. The ultimate resolution to this difficulty is found in the notion of *generalized eigenvectors*. But that will come later in the chapter. . .

Hand Examples

541/2

Find the eigenvalues and eigenvectors of $\mathbf{A} = \begin{bmatrix} 6 & -3 \\ 2 & 1 \end{bmatrix}$.

Solution

- To find the characteristic polynomial $p(r)$, we compute the determinant of the matrix $\mathbf{A} - r\mathbf{I}$, where $\mathbf{I}$ is the identity matrix having the same size as $\mathbf{A}$.
$$p(r) = \det \begin{bmatrix} 6-r & -3 \\ 2 & 1-r \end{bmatrix} = 6 - 7r + r^2 + 6 = r^2 - 7r + 12$$
- The eigenvalues of A are roots of the characteristic equation, $0 = p(r) = r^2 - 7r + 12 = (r-3)(r-4)$, whence $r = 3, 4$.
- For $r = 3$, compute the reduced row echelon form of $\mathbf{A} - r\mathbf{I}$.
$$\begin{bmatrix} 3 & -3 \\ 2 & -2 \end{bmatrix} \overset{\text{rref}}{\rightarrow} \begin{bmatrix} 1 & -1 \\ 0 & 0 \end{bmatrix}$$
In order for $(\mathbf{A} - r\mathbf{I})\mathbf{v} = \mathbf{0}$, we must have $v_1 - v_2 = 0$ or $v_1 = v_2$. Hence $3 \leftrightarrow \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ is an eigenpair.
- Similarly, for $r = 4$, we have
$$\mathbf{A} - r\mathbf{I} = \begin{bmatrix} 2 & -3 \\ 2 & -3 \end{bmatrix} \overset{\text{rref}}{\rightarrow} \begin{bmatrix} 1 & -\frac{3}{2} \\ 0 & 0 \end{bmatrix},$$
whence $4 \leftrightarrow \begin{bmatrix} 3 \\ 2 \end{bmatrix}$ is an eigenpair.
- We thus have a full set of linearly independent eigenvectors.

541/6

Find the eigenvalues and eigenvectors of $\mathbf{A} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$.

Solution

- Solve $0 = p(r) = \det(\mathbf{A} - r\mathbf{I}) = \det \begin{bmatrix} -r & 1 & 1 \\ 1 & -r & 1 \\ 1 & 1 & -r \end{bmatrix}$
$= -r(r^2 - 1) - (-r - 1) + (1 + r) = (r + 1)(2 + r - r^2)$
$= (1 + r)^2(2 - r)$ to obtain eigenvalues $r = -1, -1, 2$.

- For $r = 2$, we have

$$\mathbf{A} - r\mathbf{I} = \begin{bmatrix} -2 & 1 & 1 \\ 1 & -2 & 1 \\ 1 & 1 & -2 \end{bmatrix} \overset{\text{rref}}{\rightarrow} \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \\ 0 & 0 & 0 \end{bmatrix},$$

whence $2 \leftrightarrow \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$ is an eigenpair.

- For $r = -1$, we have

$$\mathbf{A} - r\mathbf{I} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \overset{\text{rref}}{\rightarrow} \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

so $-1 \leftrightarrow \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix}$ and $-1 \leftrightarrow \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$ are eigenpairs.

- We once again have a full set of linearly independent eigenvectors.

541/10

Find the eigenvalues and eigenvectors of $\mathbf{A} = \begin{bmatrix} 1 & 2 & -1 \\ 0 & 1 & 1 \\ 0 & -1 & 1 \end{bmatrix}$.

Solution

- Solve $0 = p(r) = \det(\mathbf{A} - r\mathbf{I}) = \det \begin{bmatrix} 1-r & 2 & -1 \\ 0 & 1-r & 1 \\ 0 & -1 & 1-r \end{bmatrix}$ $= (1-r)\left((1-r)^2 + 1\right)$ to obtain $r = 1, -1+i, -1-i$.

- For $r = 1$, $\begin{bmatrix} 0 & 2 & -1 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \overset{\text{rref}}{\rightarrow} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$, yielding eigenpair $1 \leftrightarrow \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$.

- For $r = -1+i$, $\begin{bmatrix} -i & 2 & -1 \\ 0 & -i & 1 \\ 0 & -1 & -i \end{bmatrix} \overset{\text{rref}}{\rightarrow} \begin{bmatrix} 1 & 0 & 2-i \\ 0 & 1 & i \\ 0 & 0 & 0 \end{bmatrix}$, which gives eigenpair $-1+i \leftrightarrow \begin{bmatrix} -1-2i \\ 1 \\ i \end{bmatrix}$.

- Similarly, $-1-i \leftrightarrow \begin{bmatrix} -1+2i \\ 1 \\ -i \end{bmatrix}$ is an eigenpair.

- Here we have a full set of linearly independent eigenvectors, although some are complex.

541/12

Determine a general solution of the system $\mathbf{x}' = \mathbf{A}\mathbf{x}$, where $\mathbf{A} = \begin{bmatrix} 1 & 3 \\ 12 & 1 \end{bmatrix}$.

Solution

- The eigenvalues of $\mathbf{A}$ are -5 and 7.

$$0 = \det \begin{bmatrix} 1-r & 3 \\ 12 & 1-r \end{bmatrix} = r^2 - 2r - 35 = (r+5)(r-7)$$

- $\begin{bmatrix} 6 & 3 \\ 12 & 6 \end{bmatrix} \overset{\text{rref}}{\rightarrow} \begin{bmatrix} 1 & \frac{1}{2} \\ 0 & 0 \end{bmatrix}$, whence $-5 \leftrightarrow \begin{bmatrix} 1 \\ -2 \end{bmatrix}$ is an eigenpair.

- $\begin{bmatrix} -6 & 3 \\ 12 & -6 \end{bmatrix} \overset{\text{rref}}{\rightarrow} \begin{bmatrix} 1 & -\frac{1}{2} \\ 0 & 0 \end{bmatrix}$, whence $7 \leftrightarrow \begin{bmatrix} 1 \\ 2 \end{bmatrix}$ is an eigenpair.

- A general solution is $\mathbf{x} = c_1 e^{-5t} \begin{bmatrix} 1 \\ -2 \end{bmatrix} + c_2 e^{7t} \begin{bmatrix} 1 \\ 2 \end{bmatrix}$ or $\mathbf{x} = \begin{bmatrix} c_1 e^{-5t} + c_2 e^{7t} \\ -2c_1 e^{-5t} + 2c_2 e^{7t} \end{bmatrix}$.

MATLAB Examples

As you can see from some of the hand examples, computations can get rather involved in problems involving eigenvalues and eigenvectors. MATLAB is of great assistance in this regard.

541/2 [revisited]

Find the eigenvalues and eigenvectors of $\mathbf{A} = \begin{bmatrix} 6 & -3 \\ 2 & 1 \end{bmatrix}$.

Solution

For this simple problem, we'll show how to compute eigenvalues and eigenvectors three different ways.

```
%
% NSS4-541/2: WHEREIN EIGENVALUES AND
% EIGENVECTORS ARE COMPUTED VARIOUS WAYS.
%
% Full-auto: Give me eigenvalues & eigenvectors.
A = sym([6 -3; 2 1])

A =

[  6, -3]
[  2,  1]

[V,D] = eig(A)

V =
```

```
[ 3/2,   1]
[   1,   1]


D =

[ 4, 0]
[ 0, 3]

% EIG returns a diagonal matrix D of eigenvalues
% and a full matrix V (WHEN IT EXISTS!) whose
% columns are the corresponding eigenvectors
% so that A*V = V*D; i.e., A*v = r*v for each
% eigenpair.
%
% PRELIMINARIES FOR SEMI-AUTO AND HAND WORK
I = eye(2)  % This is the identity matrix.
I =
     1     0
     0     1
r = diag(D) % Extract the eigenvalues of A.

r =

[ 4]
[ 3]

%
% Semi-auto: Give me a basis for an eigenspace.
M1 = A - r(1)*I

M1 =

[  2, -3]
[  2, -3]

null(M1)

ans =

[ 3/2]
[   1]

M2 = A - r(2)*I

M2 =

[  3, -3]
[  2, -2]

null(M2)

ans =

[ 1]
[ 1]

%
% Hand: I'll construct my own eigenspaces!
M1 = A - r(1)*I

M1 =

[  2, -3]
[  2, -3]

MR1 = rref(M1) % Finish rest with pencil.

MR1 =

[    1, -3/2]
[    0,    0]

M2 = A - r(2)*I

M2 =

[  3, -3]
[  2, -2]

MR2 = rref(M2) % Finish rest with pencil.

MR2 =

[  1, -1]
[  0,  0]

%
echo off; diary off
```

541/6 [revisited]

Find the eigenvalues and eigenvectors of $\mathbf{A} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$.

Solution

This time we'll dispense with the hand method and just look at the fully and semi-automatic methods.

```
%
% NSS4-541/6
%
A = sym([0 1 1; 1 0 1; 1 1 0])

A =

[ 0, 1, 1]
[ 1, 0, 1]
[ 1, 1, 0]

[V,D] = eig(A) % full auto

V =

[  1, -1, -1]
[  1,  1,  0]
[  1,  0,  1]


D =

[  2,  0,  0]
[  0, -1,  0]
[  0,  0, -1]

I = eye(3);
%
% Semi-auto: Give me bases for eigenspaces.
M1 = A - (-1)*I

M1 =

[ 1, 1, 1]
[ 1, 1, 1]
[ 1, 1, 1]

null(M1)

ans =

[  0,  1]
[  1,  0]
[ -1, -1]

M2 = A - 2*I

M2 =

[ -2,  1,  1]
[  1, -2,  1]
```

```
[  1,  1, -2]

null(M2)

ans =

[ 1]
[ 1]
[ 1]

%
echo off; diary off
```

543/34

Solve the IVP $\mathbf{x}' = \mathbf{A}\mathbf{x}$, where $\mathbf{A} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$ and $\mathbf{x}(0) = \mathbf{x}_0 = \begin{bmatrix} -1 \\ 4 \\ 0 \end{bmatrix}$.

Solution

Since we have additional work to do after finding eigenpairs, we'll resort to full auto mode. Note the use of matrix-vector methods to resolve the unknown constants. This is your Chapter 3 lab manual training being put to good effect!

```
%
% NSS4-543/34
%
syms t
A = sym([0 1 1; 1 0 1; 1 1 0])

A =

[ 0, 1, 1]
[ 1, 0, 1]
[ 1, 1, 0]

[V,D] = eig(A)

V =

[  1, -1, -1]
[  1,  1,  0]
[  1,  0,  1]


D =

[  2,  0,  0]
[  0, -1,  0]
[  0,  0, -1]

r = diag(D)

r =

[  2]
[ -1]
[ -1]

%
x1 = exp(r(1)*t) * V(:,1)

x1 =

[ exp(2*t)]
[ exp(2*t)]
[ exp(2*t)]

x2 = exp(r(2)*t) * V(:,2)

x2 =

[ -exp(-t)]
[  exp(-t)]
[        0]

x3 = exp(r(3)*t) * V(:,3)

x3 =

[ -exp(-t)]
[        0]
[  exp(-t)]

X = [x1 x2 x3]

X =

[ exp(2*t), -exp(-t), -exp(-t)]
[ exp(2*t),  exp(-t),        0]
[ exp(2*t),        0,  exp(-t)]

%
M = simple(subs(X, t, sym(0)))

M =

[  1, -1, -1]
[  1,  1,  0]
[  1,  0,  1]

x0 = [-1; 4; 0]
x0 =
    -1
     4
     0
c = M\x0

c =

[  1]
[  3]
[ -1]

x = X*c; pretty(x)

                            [exp(2 t) - 2 exp(-t)]
                            [                    ]
                            [exp(2 t) + 3 exp(-t)]
                            [                    ]
                            [ exp(2 t) - exp(-t) ]
%
echo off; diary off
```

541/8

Find the eigenvalues and eigenvectors of $\mathbf{A} = \begin{bmatrix} -3 & 1 & 0 \\ 0 & -3 & 1 \\ 4 & -8 & 2 \end{bmatrix}$.

Solution

When they ask you where the party went out of bounds, you can point to HERE. This is the first instance we've encountered where we do *not* have a full set of linearly independent eigenvectors. Your textbook would have you resort to all sorts of ad hoc techniques to resolve this, none of which is universal.

Instead, we'll introduce the **Jordan canonical form** (JCF), a tasty morsel from my grad school differential equations class. More specifically, we'll employ MATLAB's **jordan** command. This will always produce a full set of linearly independent *generalized* eigenvectors and their associated eigenvalues. We'll have more to said about the JCF and generalized eigenvectors in Section 9.8 as well as supplemental handouts. For now, rest assured that they constitute a "miracle weapon."

Here is the blurb from the MATLAB online help that describes the **jordan** command.

```
%
JORDAN Jordan Canonical Form.
%
JORDAN(A) computes the Jordan Canonical/Normal Form of the
matrix A. The matrix must be known exactly, so its elements
must be integers or ratios of small integers. Any errors in
the input matrix may completely change its JCF.

[P,J] = JORDAN(A) also computes the similarity
transformation, P, so that P\A*P = J.  The columns
of V are the generalized eigenvectors.

Example:
   [P,J] = jordan(A)

See also EIG, POLY.
Overloaded methods

   help sym/jordan.m
```

As stated in the help, $\mathbf{J} = \mathbf{P}^{-1}\mathbf{A}\mathbf{P}$ or $\mathbf{P}\mathbf{J} = \mathbf{A}\mathbf{P}$. A quick way to convince yourself that $\mathbf{J}$ is correct is to verify that $\mathbf{P}\mathbf{J} - \mathbf{A}\mathbf{P} = \mathbf{0}$, the zero matrix.

```
%
% NSS4-541/8
%
A = sym([-3 1 0; 0 -3 1; 4 -8 2])

A =

[ -3,  1,  0]
[  0, -3,  1]
[  4, -8,  2]

% "Brick shy of a load, Johnny?"
[V,D] = eig(A)

V =

[ 1, 1]
[ 1, 2]
[ 1, 4]


D =

[ -2,  0,  0]
[  0, -1,  0]
[  0,  0, -1]

r = diag(D)

r =

[ -2]
[ -1]
[ -1]

% The Jordan Canonical Form (JCF) to the rescue!
% Your first step down the Road to Perdition and
% exposure to generalized eigenvectors, about which
% more later...
[P,J] = jordan(A)

P =

[  4,  2, -3]
[  4,  4, -4]
[  4,  8, -4]


J =

[ -2,  0,  0]
[  0, -1,  1]
[  0,  0, -1]

checkJ = P*J - A*P

checkJ =

[ 0, 0, 0]
[ 0, 0, 0]
[ 0, 0, 0]

%
echo off; diary off
```

543/38

Determine a general solution of the system $\mathbf{x}' = \mathbf{A}\mathbf{x}$, where
$\mathbf{A} = \begin{bmatrix} 3 & -2 & 1 \\ 2 & -1 & 1 \\ -4 & 4 & 1 \end{bmatrix}$.

Solution

Once again, we see that we do not have a full set of linearly independent eigenvectors. Accordingly, we resort to the Jordan canonical form. Precisely *how* the solution is constructed from the JCF and its similarity transformation matrix P will have to wait for a weekend handout. For now in lecture, I'll try to mumble something about chains of generalized eigenvectors and Jordan blocks. Stay tuned. . .

```
%
% NSS4-543/38
%
syms c1 c2 c3 t
A = sym([3 -2 1; 2 -1 1; -4 4 1])

A =

[  3, -2,  1]
[  2, -1,  1]
[ -4,  4,  1]

[V,D] = eig(A)

V =

[ 1]
[ 1]
[ 0]


D =

[ 1, 0, 0]
[ 0, 1, 0]
```

```
[ 0, 0, 1]

r = diag(D)

r =

[ 1]
[ 1]
[ 1]

[P,J] = jordan(A)

P =

[ -4,  2,  1]
[ -4,  2,  0]
[  0, -4,  0]


J =

[ 1, 1, 0]
[ 0, 1, 1]
[ 0, 0, 1]

checkJ = P*J - A*P

checkJ =

[ 0, 0, 0]
[ 0, 0, 0]
[ 0, 0, 0]

%
eJt = exp(t) * [1 t t^2/2; 0 1 t; 0 0 1]

eJt =

[         exp(t),        exp(t)*t, 1/2*exp(t)*t^2]
[              0,          exp(t),       exp(t)*t]
[              0,               0,         exp(t)]

X = P * eJt / P; % (triple-wide trailer)
% Display it column by column.
Xcol1 = X(:,1); pretty(Xcol1)

                     [           2                  ]
                     [-2 exp(t) t  + 2 exp(t) t + exp(t)]
                     [                              ]
                     [               2              ]
                     [    -2 exp(t) t  + 2 exp(t) t     ]
                     [                              ]
                     [            -4 exp(t) t           ]
Xcol2 = X(:,2); pretty(Xcol2)

                      [              2                ]
                      [    2 exp(t) t  - 2 exp(t) t     ]
                      [                               ]
                      [          2                    ]
                      [2 exp(t) t  + exp(t) - 2 exp(t) t]
                      [                               ]
                      [           4 exp(t) t            ]
Xcol3 = X(:,3); pretty(Xcol3)

                                   [exp(t) t]
                                   [        ]
                                   [exp(t) t]
                                   [        ]
                                   [ exp(t) ]
%
check1 = diff(X,t) - A*X % The solutions are

check1 =

[ 0, 0, 0]
[ 0, 0, 0]
[ 0, 0, 0]

check2 = simple(det(X))  % linearly independent.

check2 =

exp(t)^3

%
c = [c1; c2; c3]

c =

[ c1]
[ c2]
[ c3]

x = X*c; % Long elements...
x1 = x(1); x2 = x(2); x3 = x(3);
% Display components of x separately.
pretty(x1)
%
%
              2                                          2
(-2 exp(t) t  + 2 exp(t) t + exp(t)) c1 + (2 exp(t) t  - 2 exp(t) t) c2
         + exp(t) t c3
pretty(x2)

              2                                2
(-2 exp(t) t  + 2 exp(t) t) c1 + (2 exp(t) t  + exp(t) - 2 exp(t) t) c2
         + exp(t) t c3
pretty(x3)

                  -4 exp(t) t c1 + 4 exp(t) t c2 + exp(t) c3
%

echo off; diary off
```

544/40

Determine a general solution of the system $\mathbf{x}' = \mathbf{A}\mathbf{x}$, where $\mathbf{A} = \begin{bmatrix} 1 & 3 & -2 \\ 0 & 7 & -4 \\ 0 & 9 & -5 \end{bmatrix}$.

Solution

Another day, another defective set of eigenvectors. Call in the JCF to render the needful. By looking at the chains of generalized eigenvectors in this problem and the preceding one, you may well be able to see the pattern involved in the construction of the solution. Hang in there. . .

```
%
% NSS4-544/40
%
syms c1 c2 c3 t
A = sym([1 3 -2; 0 7 -4; 0 9 -5])

A =

[  1,  3, -2]
[  0,  7, -4]
[  0,  9, -5]

[V,D] = eig(A)

V =

[   1,   0]
[   0,   1]
[   0, 3/2]


D =

[ 1, 0, 0]
[ 0, 1, 0]
[ 0, 0, 1]
```

```
r = diag(D)

r =

[ 1]
[ 1]
[ 1]

%
[P,J] = jordan(A)

P =

[   3,   0,   0]
[   6, 5/3, 2/3]
[   9,   1,   1]


J =

[ 1, 1, 0]
[ 0, 1, 0]
[ 0, 0, 1]

checkJ = P*J - A*P

checkJ =

[ 0, 0, 0]
[ 0, 0, 0]
[ 0, 0, 0]

%
eJt = [exp(t) t*exp(t) 0; ...
        0 exp(t) 0; ...
        0 0 exp(t)]

eJt =

[   exp(t), t*exp(t),        0]
[        0,   exp(t),        0]
[        0,        0,   exp(t)]

X = P * eJt / P

X =

[           exp(t),         3*t*exp(t),        -2*t*exp(t)]
[                0,  exp(t)+6*t*exp(t),        -4*t*exp(t)]
[                0,         9*t*exp(t), -6*t*exp(t)+exp(t)]

%
check1 = diff(X,t) - A*X % The solutions are

check1 =

[ 0, 0, 0]
[ 0, 0, 0]
[ 0, 0, 0]

check2 = simple(det(X))  % linearly independent.

check2 =

exp(t)^3

%
c = [c1; c2; c3]

c =

[ c1]
[ c2]
[ c3]

x = X*c; pretty(x)

               [exp(t) c1 + 3 t exp(t) c2 - 2 t exp(t) c3]
               [                                         ]
               [(exp(t) + 6 t exp(t)) c2 - 4 t exp(t) c3 ]
               [                                         ]
               [9 t exp(t) c2 + (-6 t exp(t) + exp(t)) c3]
%
echo off; diary off
```

544/44

Find a general solution of the Cauchy-Euler system

$$t\mathbf{x}'(t) = \begin{bmatrix} -4 & 2 \\ 2 & -1 \end{bmatrix}\mathbf{x}(t), \quad t > 0.$$

Solution

Via 544/42, a homework problem, $\mathbf{x}(t) = t^r\mathbf{v}$ is a solution of the Cauchy-Euler system if and only if r is an eigenvalue of $\mathbf{A}$ and $\mathbf{v}$ is an associated eigenvector.

```
%
% NSS4-544/44: Cauchy-Euler system
%
syms c1 c2 t
A = sym([-4 2; 2 -1])

A =

[ -4,  2]
[  2, -1]

[V,D] = eig(A)

V =

[  1, -2]
[  2,  1]


D =

[  0,  0]
[  0, -5]

r = diag(D)

r =

[  0]
[ -5]

%
x1 = t^r(1) * V(:,1)

x1 =

[ 1]
[ 2]

x2 = t^r(2) * V(:,2)

x2 =

[ -2/t^5]
[  1/t^5]

X = [x1 x2]

X =

[      1, -2/t^5]
[      2,  1/t^5]

%
check1 = t*diff(X,t) - A*X % The solutions are
```

```
check1 =

[ 0, 0]
[ 0, 0]

check2 = simple(det(X))  % linearly independent.

check2 =

5/t^5

%
c = [c1; c2];
x = X*c; pretty(x)
%
%
                                 [        c2 ]
                                 [c1 - 2 ----]
                                 [         5 ]
                                 [        t  ]
                                 [           ]
                                 [        c2 ]
                                 [2 c1 + ----]
                                 [         5 ]
                                 [        t  ]
%
echo off; diary off
```

542/18

Consider the system $\mathbf{x}' = \mathbf{A}\mathbf{x}$, $t \geq 0$, with $\mathbf{A} = \begin{bmatrix} -2 & 1 \\ 1 & -2 \end{bmatrix}$.

(a) Show that the matrix has eigenpairs $-1 \leftrightarrow \mathbf{u}_1 = [1; 1]$ and $-3 \leftrightarrow \mathbf{u}_2 = [1; -1]$.

(b) Sketch the trajectory of the solution having initial vector $\mathbf{x}(0) = \mathbf{u}_1$.

(c) Sketch the trajectory of the solution having initial vector $\mathbf{x}(0) = -\mathbf{u}_2$.

(d) Sketch the trajectory of the solution having initial vector $\mathbf{x}(0) = \mathbf{u}_1 - \mathbf{u}_2$.

Solution

Here are MATLAB diary files and plots.

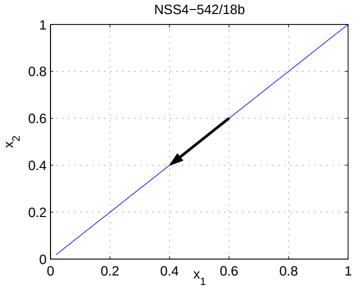
(a)

```
%
% NSS4-542/18a
%
syms c1 c2 c3 t
A = sym([-2 1; 1 -2])

A =

[ -2,  1]
[  1, -2]

[V,D] = eig(A)

V =

[  1, -1]
[  1,  1]


D =

[ -1,  0]
[  0, -3]

r = diag(D)

r =

[ -1]
[ -3]

%
echo off; diary off
```

(b)

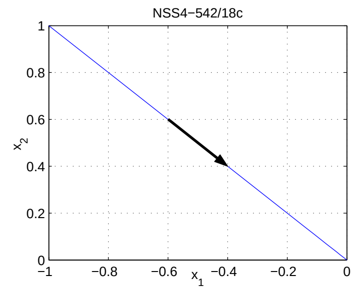
```
%
% NSS4-542/18b
%
syms t
A = sym([-2 1; 1 -2])

A =

[ -2,  1]
[  1, -2]

[V,D] = eig(A)

V =

[ -1,  1]
[  1,  1]


D =

[ -3,  0]
[  0, -1]

r = diag(D)

r =

[ -3]
[ -1]

%
u1 = [1;1]; u2 = [1;-1];
x1 = exp(-t) * u1

x1 =

[ exp(-t)]
[ exp(-t)]

x2 = exp(-3*t) * u2

x2 =

[  exp(-3*t)]
[ -exp(-3*t)]

X = [x1 x2]

X =

[    exp(-t),  exp(-3*t)]
[    exp(-t), -exp(-3*t)]

%
M = subs(X, t, 0)
```

```
M =
     1     1
     1    -1
x0 = u1
x0 =
     1
     1
c = M\x0
c =
     1
     0
x = X*c; pretty(x)

                                   [exp(-t)]
                                   [       ]
                                   [exp(-t)]
%
x1 = x(1); x2 = x(2);
t = linspace(0, 4, 50);
x1 = eval(vectorize(x1));
x2 = eval(vectorize(x2));
plot(x1,x2); grid on
%
echo off; diary off
```



(c)

```
%
% NSS4-542/18c
%
syms t
A = sym([-2 1; 1 -2])

A =

[ -2,  1]
[  1, -2]

[V,D] = eig(A)

V =

[ -1,  1]
[  1,  1]


D =

[ -3,  0]
[  0, -1]

r = diag(D)

r =

[ -3]
[ -1]

%
u1 = [1;1]; u2 = [1;-1];
x1 = exp(-t) * u1

x1 =

[ exp(-t)]
[ exp(-t)]

x2 = exp(-3*t) * u2

x2 =

[  exp(-3*t)]
[ -exp(-3*t)]

X = [x1 x2]

X =

[    exp(-t),  exp(-3*t)]
[    exp(-t), -exp(-3*t)]

%
M = subs(X, t, 0)
M =
     1     1
     1    -1
x0 = -u2
x0 =
    -1
     1
c = M\x0
c =
     0
    -1
x = X*c; pretty(x)

                                  [-exp(-3 t)]
                                  [          ]
                                  [exp(-3 t) ]
%
x1 = x(1); x2 = x(2);
t = linspace(0, 4, 50);
x1 = eval(vectorize(x1));
x2 = eval(vectorize(x2));
plot(x1,x2); grid on
%
echo off; diary off
```



(d)

```
%
% NSS4-542/18d
%
syms t
A = sym([-2 1; 1 -2])

A =

[ -2,  1]
[  1, -2]

[V,D] = eig(A)

V =

[ -1,  1]
[  1,  1]
```

```
D =

[ -3,  0]
[  0, -1]

r = diag(D)

r =

[ -3]
[ -1]

%
u1 = [1;1]; u2 = [1;-1];
x1 = exp(-t) * u1

x1 =

[ exp(-t)]
[ exp(-t)]

x2 = exp(-3*t) * u2

x2 =

[  exp(-3*t)]
[ -exp(-3*t)]

X = [x1 x2]

X =

[    exp(-t),  exp(-3*t)]
[    exp(-t), -exp(-3*t)]

%
M = subs(X, t, 0)
M =
     1     1
     1    -1
x0 = u1-u2
x0 =
     0
     2
c = M\x0
c =
     1
    -1
x = X*c; pretty(x)

                             [exp(-t) - exp(-3 t)]
                             [                   ]
                             [exp(-t) + exp(-3 t)]
%
x1 = x(1); x2 = x(2);
t = linspace(0, 4, 300);
x1 = eval(vectorize(x1));
x2 = eval(vectorize(x2));
plot(x1,x2); grid on
%
echo off; diary off
```