# MATLAB support

Here we briefly describe how to use each of the provided MATLAB functions. They are easy to use; indeed, all that is required of the user is an understanding of how to input matrices and vectors into MATLAB. Examples are included after each of the programs is briefly described. The provided examples generally correspond to an example in the book.

In order to use the below linear algebra programs it is only necessary that that the user know how to enter a matrix and vector in MATLAB. The entries in each row are separated by a comma, and each row is separated by a semi-colon. For example, if we have

$$A = \begin{pmatrix} 1 & 2 & -3 \\ -3 & -6 & 7 \end{pmatrix}, \quad b = \begin{pmatrix} 3 \\ 2 \\ 4 \end{pmatrix}$$

then they are entered in MATLAB via the command

```
» A=[1,2,-3;-3,-6,7];
» b=[3;2;4];
```

The semi-colon at the end of the line suppresses the output. If it is not present, then the entered matrix is mirrored. The supplied individual functions are:

- `cola`: find a basis for the column space of a matrix $A$ by using the command `cola(A)`
- `deta`: find the determinant of a matrix $A$ by using the command `deta(A)`
- `evalsa`: find the eigenvalues and associated eigenvectors of a matrix $A$ by using the command `evalsa(A)`
- `gaussa`: find the row reduced echelon form (RREF) of a matrix $A$ by using the command `gaussa(A)`
- `gaussab`: find the RREF of the augmented matrix $(A|b)$ by using the command `gaussab(A,b)`
- `inva`: find the inverse of a matrix $A$ by using the command `inva(A)`
- `nulla`: find a basis for the null space of a matrix $A$ by using the command `nulla(A)`
- `zerop`: find the zeros of a polynomial. The input is a row vector which contains the coefficients of the given polynomial. For example, if the polynomial is

  $$p(x) = 5x^4 - 3x^2 + 6x + 8,$$

  then the associated MATLAB commands are

```
» coef=[5,0,-3,6,8];
» zerop(coef)
```

The MATLAB code for `dfield8b` and `pplane8b` was developed by John Polking of Rice University. This code is also available off Polking's web site[1], and is provided here solely for the sake of convenience.

- `dfield8b`: numerically solves scalar ODEs $x' = f(t,x)$ for initial conditions which can be either inputted via a keyboard, or via the click of a mouse
- `pplane8b`: numerically solves autonomous planar ODEs $x' = f(x,y)$, $y' = g(x,y)$ for initial conditions which can be either inputted via a keyboard, or via the click of a mouse.

**Example 1.10.** Consider a linear system for which the coefficient matrix and nonhomogeneous term are

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 2 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} -1 \\ 4 \\ -7 \end{pmatrix}.$$

We will use the provided MATLAB function `gaussab` to put the augmented matrix into RREF. We have the sequence of commands:

```
» A=[1,2,3;4,5,6;7,8,2];
» b=[-1;4;-7];
» gaussab(A,b)
RREF of (A|b):
     1.0000          0          0     6.6190
          0     1.0000          0    -7.2381
          0          0     1.0000     2.2857
```

Ending each of the first two lines with a semi-colon suppresses the output; otherwise, the matrix and vector will be mirrored. Note that in the last command line there is no semi-colon at the end. We interpret the output to say that the linear system is consistent, and the unique solution is approximately

$$\boldsymbol{x} \sim \begin{pmatrix} 6.62 \\ -7.24 \\ 2.29 \end{pmatrix}.$$

**Example 1.20.** We consider an example for which the inverse is computed numerically. Here $A \in \mathbb{R}^{3\times 3}$, and the calculation will be done using the provided MATLAB function `inva`. Using this function for the matrix

$$A = \begin{pmatrix} 1 & 2 & 3 \\ -1 & 2 & -3 \\ 5 & 6 & 7 \end{pmatrix} \quad \rightsquigarrow \quad \mathtt{A} = [1,2,3;-1,2,-3;5,6,7];$$

generates the following output:

[1] `http://math.rice.edu/~polking/`

```
» inva(A)
Inverse of A:
  -1.0000       -0.1250       0.3750
   0.2500        0.2500            0
   0.5000       -0.1250      -0.1250
```

In other words, upon using the exact expressions for the numerical approximations we have

$$A^{-1} = \frac{1}{8}\begin{pmatrix} -8 & -1 & 3 \\ 2 & 2 & 0 \\ 4 & -1 & -1 \end{pmatrix}.$$

It is straightforward to check that $AA^{-1} = I_3$.

**Example 1.26.** Consider the homogeneous linear system $Ax = 0$, where

$$A = \begin{pmatrix} 3 & 4 & 7 & -1 \\ 2 & 6 & 8 & -4 \\ -5 & 3 & -2 & -8 \\ 7 & -2 & 5 & 9 \end{pmatrix}.$$

We will use the provided MATLAB command `nulla` to find a spanning set for Null($A$):

```
» A=[3,4,7,-1;2,6,8,-4;-5,3,-2,-8;7,-2,5,9];
» nulla(A)
Basis for Null(A):
        -1        -1
        -1         1
         1         0
         0         1
```

so that

$$\mathrm{Null}(A) = \mathrm{Span}\left\{\begin{pmatrix} -1 \\ -1 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} -1 \\ 1 \\ 0 \\ 1 \end{pmatrix}\right\}.$$

**Example 1.29.** Consider the linear system $Ax = b$, where

$$A = \begin{pmatrix} 3 & 4 & -7 & 2 \\ 2 & 6 & 9 & -2 \\ -5 & 3 & 2 & -13 \\ 7 & -2 & 5 & 16 \end{pmatrix}, \quad b = \begin{pmatrix} 5 \\ 27 \\ 11 \\ -1 \end{pmatrix}.$$

We will use the provided MATLAB command `nulla` to find the homogeneous solution, and the provided MATLAB command `gaussab` to find a particular solution. Regarding the null space we have

```
» A=[3,4,-7,2;2,6,9,-2;-5,3,2,-13;7,-2,5,16];
» b=[5;27;11;-1];
» nulla(A)
Basis for Null(A):
        -2
         1
         0
         1
```

so

$$\mathrm{Null}(A) = \mathrm{Span}\left\{ \begin{pmatrix} -2 \\ 1 \\ 0 \\ 1 \end{pmatrix} \right\}.$$

The homogeneous solution is then

$$\boldsymbol{x}_{\mathrm{h}} = t \begin{pmatrix} -2 \\ 1 \\ 0 \\ 1 \end{pmatrix}, \quad t \in \mathbb{R}.$$

In order to find the particular solution we do not need to reenter the MATLAB expressions for $\boldsymbol{A}$ and $\boldsymbol{b}$, so we simply type

```
» gaussab(A,b)
RREF of (A|b):
         1    0    0     2    0
         0    1    0    -1    3
         0    0    1     0    1
         0    0    0     0    0
```

The particular solution is the last column of the RREF of $(\boldsymbol{A}|\boldsymbol{b})$,

$$\boldsymbol{x}_{\mathrm{p}} = \begin{pmatrix} 0 \\ 3 \\ 1 \\ 0 \end{pmatrix}.$$

**Example 1.66.** We now consider an example for which a basis for $\mathrm{Col}(\boldsymbol{A})$ and $\mathrm{Null}(\boldsymbol{A})$ are computed numerically. Here $\boldsymbol{A} \in \mathbb{R}^{4\times 4}$ is given by

$$\boldsymbol{A} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ -1 & 2 & 1 & 0 \\ 5 & 6 & 11 & 16 \\ 2 & 4 & 6 & 8 \end{pmatrix} \quad \rightsquigarrow \quad \mathtt{A} = [1,2,3,4;-1,2,1,0;5,6,11,16;2,4,6,8];$$

Using the provided MATLAB function `gaussa` generates the output:

```
» gaussa(A)
RREF of A:
         1    0    1    2
         0    1    1    1
         0    0    0    0
         0    0    0    0
```

From this RREF we see that $\mathrm{rank}(\boldsymbol{A}) = 2$, and $\dim[\mathrm{Null}(\boldsymbol{A})] = 2$. In order to numerically generate a basis for the column space and null space we use the provided MATLAB functions `cola` and `nulla`, respectively, to get:

```
» cola(A)
Basis for Col(A):
          1          2
         -1          2
          5          6
          2          4
» nulla(A)
Basis for Null(A):
         -1         -2
         -1         -1
          1          0
          0          1
```

As expected, we get as a basis for Col($A$) the first two columns of $A$. As for the null space,

$$\mathrm{Null}(A) = \mathrm{Span}\left\{\begin{pmatrix} -1 \\ -1 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} -2 \\ -1 \\ 0 \\ 1 \end{pmatrix}\right\}.$$

**Section 1.10.** We now calculate the determinant of a matrix using the provided MATLAB function `deta`. Using this function for the matrix

$$A = \begin{pmatrix} 1 & 2 & 3 \\ -1 & 2 & -3 \\ 5 & 6 & 7 \end{pmatrix} \quad \rightsquigarrow \quad \texttt{A = [1,2,3;-1,2,-3;5,6,7];}$$

we have the following output:

```
» deta(A)
det(A): -32
```

In other words, $\det(A) = -32$.

**Example 1.97.** We consider an example for which the eigenvalues and eigenvectors must be computed numerically. Here $A \in \mathbb{R}^{3\times 3}$, which means that $p_A(\lambda)$ is a third-order polynomial. Unless the problem is very special, it is generally the case that it is not possible to (easily) find the three roots. This calculation will be done using the provided MATLAB function `evalsa`. Using this function for the matrix

$$A = \begin{pmatrix} 1 & 2 & 3 \\ -1 & 2 & -3 \\ 5 & 6 & 7 \end{pmatrix} \quad \rightsquigarrow \quad \texttt{A = [1,2,3;-1,2,-3;5,6,7];}$$

we have the following output:

```
» evalsa(A)
Eigenvalues:
  -0.8576 + 0.0000i    5.4288 + 2.8000i    5.4288 - 2.8000i
Eigenvectors:
  -0.8830 + 0.0000i   -0.2981 + 0.0209i   -0.2981 - 0.0209i
   0.1559 + 0.0000i    0.4467 - 0.3709i    0.4467 + 0.3709i
   0.4428 + 0.0000i   -0.7574 + 0.0000i   -0.7574 + 0.0000i
```

In other words,

$$\lambda_1 \sim -0.86,\ v_1 \sim \begin{pmatrix} -0.88 \\ 0.16 \\ 0.44 \end{pmatrix};\ \lambda_2 \sim 5.43 + \mathrm{i}2.80,\ v_2 \sim \begin{pmatrix} -0.30 \\ 0.45 \\ -0.76 \end{pmatrix} + \mathrm{i} \begin{pmatrix} 0.02 \\ -0.37 \\ 0.00 \end{pmatrix}.$$

The third eigenvalue is the complex-conjugate conjugate of the second, i.e., $\lambda_3 = \overline{\lambda_2}$, and the associated eigenvector is the complex-conjugate of $v_2$, i.e., $v_3 = \overline{v_2}$.

**Example 2.15.** Let us find the general solution to the scalar ODE,

$$x' = -x + t - 4t^3,$$

using the method of undetermined coefficients. Since $f(t) = t - 4t^3$, which is a third-order polynomial, we will guess the particular solution to be

$$x_{\mathrm{p}}(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3.$$

As seen in the text, this leads to the linear system,

$$a_0 + a_1 = 0,\ a_1 + 2a_2 - 1 = 0,\ a_2 + 3a_3 = 0,\ a_3 + 4 = 0,$$

which can be rewritten in matrix/vector form as

$$\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 2 & 0 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{pmatrix} a = \begin{pmatrix} 0 \\ 1 \\ 0 \\ -4 \end{pmatrix}, \quad a = \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix}.$$

We use the provided MATLAB function `gaussab` to solve the linear system via the sequence of commands

```
» A=[1,1,0,0;0,1,2,0;0,0,1,3;0,0,0,1];
» b=[0,1,0,-4];
» gaussab(A,b)
RREF of (A|b):
          1    0    0    0     23
          0    1    0    0    -23
          0    0    1    0     12
          0    0    0    1     -4
```

We conclude

$$a = \begin{pmatrix} 23 \\ -23 \\ 12 \\ -4 \end{pmatrix},$$

so the particular solution is

$$x_{\mathrm{p}}(t) = 23 - 23t + 12t^2 - 4t^3.$$

The general solution is

$$x(t) = c_1 \mathrm{e}^{-t} + 23 - 23t + 12t^2 - 4t^3.$$

**Example 4.3.** Consider the third-order ODE

$$y''' - 3y'' + 6y' + 9y = 0.$$

The associated characteristic polynomial is

$$p(\lambda) = \lambda^3 - 3\lambda^2 + 6\lambda + 9.$$

The roots cannot easily be found analytically. However, they can be found numerically using the provided MATLAB function `zerop`. In general, if the characteristic polynomial is

$$p(\lambda) = \lambda^n + a_{n-1}\lambda^{n-1} + \cdots + a_1\lambda + a_0,$$

the roots of the polynomial are found by first putting the coefficients into a row vector, and then calling the command. For our particular example, the sequence of commands

```
» coef=[1,-3,6,9];
» zerop(coef)
Roots:
  -4.6890 + 0.0000i    0.8445 + 1.0983i    0.8445 - 1.0983i
```

reveals the three roots to be

$$\lambda_1 \sim -4.69, \quad \lambda_2 \sim 0.84 + \mathrm{i}1.10, \quad \lambda_3 \sim 0.84 - \mathrm{i}1.10.$$

Since the roots must come in complex-conjugate pairs, we know the first root is purely real. The general solution is

$$y(t) = c_1 \mathrm{e}^{-4.69t} + c_2 \mathrm{e}^{0.84t}\cos(1.10t) + c_3 \mathrm{e}^{0.84t}\sin(1.10t).$$