

Load the linear algebra package. Remember that Maple commands must end with either a colon (:) or a semicolon (;). Use a colon to suppress the output, and use a semicolon to display the output. If we want help on a command, all we need to do is type the command, highlight it with the mouse, and select Help from the drop down menu. Or, we can highlight one of the commands displayed as the output for with(linalg) and select Help.

```
> with(linalg);
```

```
Warning, the protected names norm and trace have been redefined and
unprotected
```

```
[BlockDiagonal, GramSchmidt, JordanBlock, LUdecomp, QRdecomp, Wronskian, addcol,
addrow, adj, adjoint, angle, augment, backsub, band, basis, bezout, blockmatrix,
charmat, charpoly, cholesky, col, coldim, colspace, colspan, companion, concat,
cond, copyinto, crossprod, curl, definite, delcols, delrows, det, diag, diverge,
dotprod, eigenvals, eigenvalues, eigenvectors, eigenvects, entermatrix, equal,
exponential, extend, ffgausselim, fibonacci, forwardsub, frobenius, gausselim,
gaussjord, geneqns, genmatrix, grad, hadamard, hermite, hessian, hilbert,
htranspose, ihermite, indexfunc, innerprod, intbasis, inverse, ismith, issimilar,
iszero, jacobian, jordan, kernel, laplacian, leastsqrs, linsolve, matadd, matrix,
minor, minpoly, mulcol, mulrow, multiply, norm, normalize, nullspace, orthog,
permanent, pivot, potential, randmatrix, randvector, rank, ratform, row, rowdim,
rowspan, rref, scalarmul, singularvals, smith, stackmatrix, submatrix,
subvector, sumbasis, swapcol, swaprow, sylvester, toeplitz, trace, transpose,
vandermonde, vecpotent, vectdim, vector, wronskian]
```

Example #1

Define a matrix with the matrix command. The first two arguments are the number of rows and number of columns, respectively. The third argument, enclosed in square brackets [] lists the entries of the matrix. This particular matrix P is a stochastic matrix (each row adds up to 1) and represents the transition matrix of some Markov chain.

```
> P:=matrix(3,3,[0.3, 0.4, 0.3, 0.7, 0.3, 0, 0.1, 0.6, 0.3]);
```

$$P := \begin{bmatrix} .3 & .4 & .3 \\ .7 & .3 & 0 \\ .1 & .6 & .3 \end{bmatrix}$$

We can compute powers of this matrix. Remember that P^2 is just shorthand for P * P (the matrix product of P and P). We need the command evalm to evaluate any matrix expression.

```
> evalm(P^(2));
```

$$\begin{bmatrix} .40 & .42 & .18 \\ .42 & .37 & .21 \\ .48 & .40 & .12 \end{bmatrix}$$

Note that just typing P^2 does not help us!

```
> P^2;
```

$$P^2$$

We compute a higher power. It is a fact that (under certain conditions) P^n converges as n -> infinity to some matrix whose rows are all identical. In this case, P satisfies those conditions, and so we can approxi-

mate this limit by looking at some higher powers of P.

```
> evalm(P^(10));
      [ .4224142708  .3965516274  .1810341018 ]
      [ .4224129042  .3965521333  .1810349625 ]
      [ .4224146256  .3965510536  .1810343208 ]
> evalm(P^(20));
      [ .4224137931  .3965517242  .1810344827 ]
      [ .4224137932  .3965517242  .1810344828 ]
      [ .4224137931  .3965517242  .1810344828 ]
> evalm(P^(30));
      [ .4224137930  .3965517242  .1810344826 ]
      [ .4224137931  .3965517242  .1810344828 ]
      [ .4224137931  .3965517242  .1810344827 ]
> evalm(P^(100));
      [ .4224137930  .3965517240  .1810344827 ]
      [ .4224137933  .3965517243  .1810344828 ]
      [ .4224137932  .3965517243  .1810344828 ]
```

It seems that we have some variability in the 10th decimal place. This is due to Maple rounding-off the displayed answer. We are pretty confident that we know what the limit of P^n is accurate to 9 decimal places.

Example #2

In order to illustrate diagonalizing matrices, we will work with a 2 x 2 matrix.

```
> A:=matrix(2,2,[.4,.6,.5,.5]);
```

$$A := \begin{bmatrix} .4 & .6 \\ .5 & .5 \end{bmatrix}$$

We find the eigenvalues of A...

```
> eigenvals(A);
      -1.000000000, 1.000000000
```

and the corresponding eigenvectors of A. (Technically, eigenvectors are not unique. Any scalar multiple of a eigenvector is also an eigenvector.)

```
> eigenvecs(A);
      [1.000000000, 1, {[-.7100226978, -.7100226979]}],
      [-.1000000000, 1, {[-.7682212797, .6401843996]}]
```

The output gives the eigenvalue, its multiplicity, and the corresponding eigenvector.

Recall that we can diagonalize A. That is, we can write $A=C*B*C^{-1}$ where B is a diagonal matrix with the eigenvalues on the diagonal, and C is the matrix whose columns correspond to the eigenvectors.

```
> B:=matrix(2,2,[1,0,0,-.1]);
```

$$B := \begin{bmatrix} 1 & 0 \\ 0 & -.1 \end{bmatrix}$$

It is a little irritating that Maple has no simple way to diagonalize a matrix. Instead, I create a matrix C2 which is the transpose of the matrix C. This matrix is easy to create by cutting-and-pasting the output of the

eigenvects(A) command into the following command. DO NOT TYPE THESE NUMBERS OUT BY HAND!

```
> C2:=matrix(2,2,[-.7100226978, -.7100226979,-.7682212797,
> .6401843996]);
```

$$C2 := \begin{bmatrix} -.7100226978 & -.7100226979 \\ -.7682212797 & .6401843996 \end{bmatrix}$$

The matrix C is the inverse of the matrix C2. Be sure that the ith column in C is the eigenvector corresponding to the ith entry in the diagonal matrix B.

```
> C:=transpose(C2);
```

$$C := \begin{bmatrix} -.7100226978 & -.7682212797 \\ -.7100226979 & .6401843996 \end{bmatrix}$$

We can determine the inverse of C using either of the following commands.

```
> evalm(C^(-1));
```

$$\begin{bmatrix} -.6401843996 & -.7682212797 \\ -.7100226979 & .7100226978 \end{bmatrix}$$

```
> inverse(C);
```

$$\begin{bmatrix} -.6401843996 & -.7682212797 \\ -.7100226979 & .7100226978 \end{bmatrix}$$

We check that the matrix product C*B*C^(-1) does, in fact, equal A. If we want to multiply matrices together, we need to use &* instead of just *.

```
> evalm(C&*B&*inverse(C));
```

$$\begin{bmatrix} .3999999999 & .6000000000 \\ .5000000001 & .5000000002 \end{bmatrix}$$

If we are interested in the matrix exponential exp(At), we proceed as follows.

```
> exponential(A,t);
```

$$\begin{bmatrix} .5454545454 \%2 + .4545454546 \%1 & .5454545455 \%1 - .5454545455 \%2 \\ .4545454546 \%1 - .4545454546 \%2 & .4545454545 \%2 + .5454545455 \%1 \end{bmatrix}$$

$$\%1 := e^{(1.000000000 t)}$$

$$\%2 := e^{(-.1000000000 t)}$$

Example #3

There are a couple of ways to define an identity matrix. Here we define Id to be the 3x3 identity matrix.

```
> Id:=Matrix(3,3,shape=identity);
```

$$Id := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

This is a second way to define the 3x3 identity matrix. Notice that to display the matrix Idd we need the second evalm(Idd) command.

```
> Idd:=array(1..3,1..3, identity);
```

$$Idd := \text{array}(\text{identity}, 1..3, 1..3, [])$$

```
> evalm(Idd);
```

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

Define a matrix Q as follows.

```
> Q:=matrix(3,3,[0,1/6,1/6,0,1/6,1/6,1/6,0,0]);
```

$$Q := \begin{bmatrix} 0 & \frac{1}{6} & \frac{1}{6} \\ 0 & \frac{1}{6} & \frac{1}{6} \\ \frac{1}{6} & 0 & 0 \end{bmatrix}$$

Determine the inverse of the identity minus Q (i.e., either Idd-Q or Id - Q)

```
> inverse(evalm(Idd-Q));
```

$$\begin{bmatrix} \frac{30}{29} & \frac{6}{29} & \frac{6}{29} \\ \frac{1}{29} & \frac{35}{29} & \frac{6}{29} \\ \frac{5}{29} & \frac{1}{29} & \frac{30}{29} \end{bmatrix}$$

```
> inverse(evalm(Id-Q));
```

$$\begin{bmatrix} \frac{30}{29} & \frac{6}{29} & \frac{6}{29} \\ \frac{1}{29} & \frac{35}{29} & \frac{6}{29} \\ \frac{5}{29} & \frac{1}{29} & \frac{30}{29} \end{bmatrix}$$

Add together the entries in the first row of Id-Q. DO NOT RECOPY THE NUMBERS — cut-and-paste-them instead.

```
> 30/29+6/29+6/29;
```

$$\frac{42}{29}$$

If we want a decimal approximation for an answer, we use the evalf (floating point evaluation) command. The % is shorthand for the previous answer.

```
> evalf(%);
```

$$1.448275862$$

Alternatively, we could evalf directly.

```
> evalf(30/29+6/29+6/29);
```

$$1.448275862$$