Chapter 11
Linear Algebra

11.1 Constructing Matrices

A matrix in *Mathematica* is a list of lists; all the lists—rows of the matrix—must have the same length.

In[1] := M = {{a,b},{c,d}}

Out[1] = {{a,b},{c,d}}

MatrixForm is used to print a matrix nicely.

In[2] := MatrixForm[M]

Out[2]//MatrixForm =

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

If the (i,j)th element of a matrix is given by an expression depending on i and j, this matrix can be constructed by the function Table.

In[3] := A = Table[a[i,j],{i,1,2},{j,1,2}]

Out[3] = {{a[1,1],a[1,2]},{a[2,1],a[2,2]}}

In[4] := B = Table[1/(i+j+1),{i,1,2},{j,1,2}]

Out[4] = $\left\{\left\{\frac{1}{3},\frac{1}{4}\right\},\left\{\frac{1}{4},\frac{1}{5}\right\}\right\}$

In[5] := MatrixForm[A]

Out[5]//MatrixForm =

$$\begin{pmatrix} a[1,1] & a[1,2] \\ a[2,1] & a[2,2] \end{pmatrix}$$

In[6] := MatrixForm[B]

Out[6]//MatrixForm =

$$\begin{pmatrix} \frac{1}{3} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{5} \end{pmatrix}$$

The function Array is similar, but its first argument is a function of two parameters, not an expression. It may be just a symbol or an anonymous function (Function).

In[7] := A = Array[a,{2,2}]

Out[7] = {{a[1,1],a[1,2]},{a[2,1],a[2,2]}}

A. Grozin, *Introduction to Mathematica® for Physicists*, Graduate Texts in Physics,
DOI 10.1007/978-3-319-00894-3_11, © Springer International Publishing Switzerland 2014

In[8] := B = Array[Function[{i, j}, 1/(i + j + 1)], {2, 2}]
Out[8] = $\left\{\left\{\frac{1}{3}, \frac{1}{4}\right\}, \left\{\frac{1}{4}, \frac{1}{5}\right\}\right\}$
Mathematica does not distinguish column vectors and row vectors; both are just lists.
In[9] := V = Table[v[i], {i, 1, 2}]
Out[9] = $\{v[1], v[2]\}$
In[10] := V = Array[v, 2]
Out[10] = $\{v[1], v[2]\}$
In[11] := U = Array[u, 2]
Out[11] = $\{u[1], u[2]\}$

11.2 Parts of a Matrix

A matrix element.
In[12] := A[[1, 2]]
Out[12] = $a[1, 2]$
A row.
In[13] := A[[1]]
Out[13] = $\{a[1, 1], a[1, 2]\}$
A column.
In[14] := A[[All, 2]]
Out[14] = $\{a[1, 2], a[2, 2]\}$
A new value can be assigned to a matrix element.
In[15] := M[[1, 2]] = 0
Out[15] = 0
In[16] := MatrixForm[M]
Out[16]//MatrixForm =
$$\begin{pmatrix} a & 0 \\ c & d \end{pmatrix}$$
Add 1 to the first row.
In[17] := M[[1]] ++
Out[17] = $\{a, 0\}$
In[18] := MatrixForm[M]
Out[18]//MatrixForm =
$$\begin{pmatrix} 1+a & 1 \\ c & d \end{pmatrix}$$
Add the second column to the first one.
In[19] := M[[All, 1]] += M[[All, 2]]
Out[19] = $\{2 + a, c + d\}$
In[20] := MatrixForm[M]
Out[20]//MatrixForm =
$$\begin{pmatrix} 2+a & 1 \\ c+d & d \end{pmatrix}$$

11.3 Queries

The function VectorQ checks if its argument is a vector, i.e., a list whose elements are not lists.

In[21] := {VectorQ[*V*], VectorQ[*M*]}

Out[21] = {True, False}

The function MatrixQ checks if its argument is a matrix, i.e., a list of same-length lists.

In[22] := {MatrixQ[*V*], MatrixQ[*M*], MatrixQ[{{*a*, *b*}, {*x*, *y*, *z*}}]}

Out[22] = {False, True, False}

The argument of the function Dimensions must be a matrix. This function returns a two-element list—the numbers of rows and columns of the matrix.

In[23] := Dimensions[*M*]

Out[23] = {2, 2}

It can be conveniently used for simultaneous assignment to two variables.

In[24] := {n1, n2} = Dimensions[{{*a*, *b*, *c*}, {*x*, *y*, *z*}}]

Out[24] = {2, 3}

In[25] := n1

Out[25] = 2

In[26] := n2

Out[26] = 3

In[27] := Clear[*M*, n1, n2]

11.4 Operations with Matrices and Vectors

Vectors can be added and multiplied by scalar expressions.

In[28] := 2 ∗ *V* + *U*

Out[28] = {*u*[1] + 2*v*[1], *u*[2] + 2*v*[2]}

Matrices can be added and multiplied by scalar expressions.

In[29] := MatrixForm[*A* + 2 ∗ *B*]

Out[29]//MatrixForm =

$$\begin{pmatrix} \frac{2}{3} + a[1,1] & \frac{1}{2} + a[1,2] \\ \frac{1}{2} + a[2,1] & \frac{2}{5} + a[2,2] \end{pmatrix}$$

The scalar product of two vectors.

In[30] := *V*.*U*

Out[30] = *u*[1]*v*[1] + *u*[2]*v*[2]

The product of a matrix and a column vector.

In[31] := *A*.*V*

Out[31] = {*a*[1, 1]*v*[1] + *a*[1, 2]*v*[2], *a*[2, 1]*v*[1] + *a*[2, 2]*v*[2]}

The product of a row vector and a matrix.

In[32] := *V*.*A*

Out[32] = {*a*[1, 1]*v*[1] + *a*[2, 1]*v*[2], *a*[1, 2]*v*[1] + *a*[2, 2]*v*[2]}

The product of two matrices.

In[33] := MatrixForm[*A*.*B*]

Out[33]//MatrixForm =

$$\begin{pmatrix} \frac{1}{3}a[1,1]+\frac{1}{4}a[1,2] & \frac{1}{4}a[1,1]+\frac{1}{5}a[1,2] \\ \frac{1}{3}a[2,1]+\frac{1}{4}a[2,2] & \frac{1}{4}a[2,1]+\frac{1}{5}a[2,2] \end{pmatrix}$$

It is not commutative.

In[34] := MatrixForm[*A*.*B* − *B*.*A*]

Out[34]//MatrixForm =

$$\begin{pmatrix} \frac{1}{4}a[1,2]-\frac{1}{4}a[2,1] & \frac{1}{4}a[1,1]-\frac{2}{15}a[1,2]-\frac{1}{4}a[2,2] \\ -\frac{1}{4}a[1,1]+\frac{2}{15}a[2,1]+\frac{1}{4}a[2,2] & -\frac{1}{4}a[1,2]+\frac{1}{4}a[2,1] \end{pmatrix}$$

Determinant (the matrix must be square).

In[35] := Det[*A*]

Out[35] = $-a[1,2]a[2,1]+a[1,1]a[2,2]$

Trace (the matrix must be square).

In[36] := Tr[*A*]

Out[36] = $a[1,1]+a[2,2]$

Transposing.

In[37] := MatrixForm[Transpose[*A*]]

Out[37]//MatrixForm =

$$\begin{pmatrix} a[1,1] & a[2,1] \\ a[1,2] & a[2,2] \end{pmatrix}$$

The inverse matrix.

In[38] := MatrixForm[Inverse[*A*]]

Out[38]//MatrixForm =

$$\begin{pmatrix} \frac{a[2,2]}{-a[1,2]a[2,1]+a[1,1]a[2,2]} & -\frac{a[1,2]}{-a[1,2]a[2,1]+a[1,1]a[2,2]} \\ -\frac{a[2,1]}{-a[1,2]a[2,1]+a[1,1]a[2,2]} & \frac{a[1,1]}{-a[1,2]a[2,1]+a[1,1]a[2,2]} \end{pmatrix}$$

A square matrix can be raised to an integer power.

In[39] := MatrixForm[MatrixPower[*B*, 3]]

Out[39]//MatrixForm =

$$\begin{pmatrix} \frac{197}{2160} & \frac{1009}{14400} \\ \frac{1009}{14400} & \frac{323}{6000} \end{pmatrix}$$

The power -1 is the inverse matrix.

In[40] := MatrixForm[MatrixPower[*B*, −1]]

Out[40]//MatrixForm =

$$\begin{pmatrix} 48 & -60 \\ -60 & 80 \end{pmatrix}$$

This is the solution of the linear system $A.X = V$.

In[41] := Together[Inverse[*A*].*V*]

Out[41] = $\left\{ \frac{a[2,2]v[1]-a[1,2]v[2]}{-a[1,2]a[2,1]+a[1,1]a[2,2]}, \frac{a[2,1]v[1]-a[1,1]v[2]}{a[1,2]a[2,1]-a[1,1]a[2,2]} \right\}$

The same can be done using LinearSolve.

In[42] := LinearSolve[*A*, *V*]

Out[42] = $\left\{ \frac{a[2,2]v[1]-a[1,2]v[2]}{-a[1,2]a[2,1]+a[1,1]a[2,2]}, \frac{a[2,1]v[1]-a[1,1]v[2]}{a[1,2]a[2,1]-a[1,1]a[2,2]} \right\}$

In[43] := Clear[*A*, *B*, *U*, *V*]

11.5 Eigenvalues and Eigenvectors

Here is some symbolic matrix.

In[44] := MatrixForm[*M* =
{{(1−*x*)^3∗(3+*x*), 4∗*x*∗(1−*x*^2), −2∗(1−*x*^2)∗(3−*x*)},
{4∗*x*∗(1−*x*^2), −(1+*x*)^3∗(3−*x*), 2∗(1−*x*^2)∗(3+*x*)},
{−2∗(1−*x*^2)∗(3−*x*), 2∗(1−*x*^2)∗(3+*x*), 16∗*x*}}]

Out[44]//MatrixForm =

$$\begin{pmatrix} (1-x)^3(3+x) & 4x\left(1-x^2\right) & -2(3-x)\left(1-x^2\right) \\ 4x\left(1-x^2\right) & -(3-x)(1+x)^3 & 2(3+x)\left(1-x^2\right) \\ -2(3-x)\left(1-x^2\right) & 2(3+x)\left(1-x^2\right) & 16x \end{pmatrix}$$

It is singular.

In[45] := Det[*M*]

Out[45] = 0

Its rank.

In[46] := MatrixRank[*M*]

Out[46] = 2

The function NullSpace returns a list of vectors forming a basis of the null space of the matrix, i.e., the subspace of vectors nullified by the matrix.

In[47] := *s* = NullSpace[*M*]

$$\text{Out}[47] = \left\{\left\{-\frac{2}{-1+x}, \frac{2}{1+x}, 1\right\}\right\}$$

In this case, the null space is one-dimensional—it has a single basis vector. Let's check it.

In[48] := Together[*M*.*s*[[1]]]

Out[48] = {0, 0, 0}

The function Eigenvalues returns a list of eigenvalues of a matrix.

In[49] := Simplify[Eigenvalues[*M*], Element[*x*, Reals]]

$$\text{Out}[49] = \left\{0, \left(3+x^2\right)^2, -\left(3+x^2\right)^2\right\}$$

We have added the second argument to Simplify which informs *Mathematica* that the variable x is real. The function Eigenvectors returns the list of the corresponding eigenvectors (in the same order).

In[50] := Simplify[Eigenvectors[*M*], Element[*x*, Reals]]

$$\text{Out}[50] = \left\{\left\{-\frac{2}{-1+x}, \frac{2}{1+x}, 1\right\}, \left\{\frac{-1+x}{1+x}, \frac{1-x}{2}, 1\right\}, \left\{\frac{1+x}{2}, \frac{1+x}{-1+x}, 1\right\}\right\}$$

The function Eigensystem returns both. It is convenient for simultaneous assignment to two variables.

In[51] := {val, vec} = Simplify[Eigensystem[*M*], Element[*x*, Reals]];

Let's check.

In[52] := Do[Print[Simplify[*M*.vec[[*i*]] − val[[*i*]] ∗ vec[[*i*]]]], {*i*, 1, 3}]

{0, 0, 0}
{0, 0, 0}
{0, 0, 0}

In[53] := Clear[*M*, *s*, val, vec]

11.6 Jordan Form

Here is a matrix of rational numbers.

In[54] := MatrixForm[*M* =
{{13/9, −2/9, 1/3, 4/9, 2/3},
{−2/9, 10/9, 2/15, −2/9, −11/15},
{1/5, −2/5, 41/25, −2/5, 12/25},
{4/9, −2/9, 14/15, 13/9, −2/15},
{−4/15, 8/15, 12/25, 8/15, 34/25}}]

Out[54]//MatrixForm =

$$\begin{pmatrix} \frac{13}{9} & -\frac{2}{9} & \frac{1}{3} & \frac{4}{9} & \frac{2}{3} \\ -\frac{2}{9} & \frac{10}{9} & \frac{2}{15} & -\frac{2}{9} & -\frac{11}{15} \\ \frac{1}{5} & -\frac{2}{5} & \frac{41}{25} & -\frac{2}{5} & \frac{12}{25} \\ \frac{4}{9} & -\frac{2}{9} & \frac{14}{15} & \frac{13}{9} & -\frac{2}{15} \\ -\frac{4}{15} & \frac{8}{15} & \frac{12}{25} & \frac{8}{15} & \frac{34}{25} \end{pmatrix}$$

The function JordanDecomposition returns a pair of matrices—the Jordan form J and the transformation matrix P which reduces our matrix to its Jordan form.

In[55] := {*P*,*J*} = JordanDecomposition[*M*];

In[56] := MatrixForm[*J*]

Out[56]//MatrixForm =

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 1-\mathrm{i} & 0 \\ 0 & 0 & 0 & 0 & 1+\mathrm{i} \end{pmatrix}$$

In[57] := MatrixForm[*P*]

Out[57]//MatrixForm =

$$\begin{pmatrix} -2 & 1 & 0 & \frac{5\mathrm{i}}{12} & -\frac{5\mathrm{i}}{12} \\ -2 & -\frac{1}{2} & 0 & -\frac{5\mathrm{i}}{6} & \frac{5\mathrm{i}}{6} \\ 0 & 0 & \frac{6}{5} & -\frac{3}{4} & -\frac{3}{4} \\ 1 & 1 & 0 & -\frac{5\mathrm{i}}{6} & \frac{5\mathrm{i}}{6} \\ 0 & 0 & \frac{9}{10} & 1 & 1 \end{pmatrix}$$

Let's check.

In[58] := MatrixForm[*P*.*J*.Inverse[*P*] − *M*]

Out[58]//MatrixForm =

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Here are the eigenvalues and the eigenvectors of our matrix. Note that only one eigenvector corresponds to the eigenvalue 2, because the corresponding Jordan block has the size 2×2. The eigenvectors are the columns of the transformation matrix P.

In[59] := {val, vec} = Eigensystem[*M*];
In[60] := val
Out[60] = $\{2, 2, 1+\mathrm{i}, 1-\mathrm{i}, 1\}$
In[61] := vec
Out[61] = $\left\{\left\{1, -\frac{1}{2}, 0, 1, 0\right\}, \{0,0,0,0,0\}, \left\{-\frac{5\mathrm{i}}{12}, \frac{5\mathrm{i}}{6}, -\frac{3}{4}, \frac{5\mathrm{i}}{6}, 1\right\}, \left\{\frac{5\mathrm{i}}{12}, -\frac{5\mathrm{i}}{6}, -\frac{3}{4}, -\frac{5\mathrm{i}}{6}, 1\right\}, \{-2,-2,0,1,0\}\right\}$
In[62] := Clear[*M*, *J*, *P*, val, vec]

11.7 Symbolic Vectors, Matrices, and Tensors

Let's inform *Mathematica* that u, v, and w are symbolic three-dimensional vectors with real components. The scalar product is $u.v$, and the vector product is Cross$[u, v]$. The function TensorReduce simplifies expressions with vectors.
In[63] := $Assumptions = Element[*u*|*v*|*w*, Vectors[3, Reals]]
Out[63] = $(u|v|w) \in$ Vectors$[3,$ Reals$]$
In[64] := TensorReduce[*u*.*v* − *v*.*u*]
Out[64] = 0
In[65] := TensorReduce[Cross[*u*, *v*] + Cross[*v*, *u*]]
Out[65] = 0
In[66] := TensorReduce[*u*.Cross[*v*, *w*] + *v*.Cross[*w*, *u*] + *w*.Cross[*u*, *v*]]
Out[66] = $3u \times v.w$
In[67] := TensorReduce[Cross[*u*, Cross[*v*, *w*]]]
Out[67] = $-wu.v + vu.w$
In[68] := TensorReduce[*u*.(2 ∗ *v* + 3 ∗ *w*)]
Out[68] = $2u.v + 3u.w$

Now let's say that u and v are d-dimensional vectors, and S and A are $d \times d$ matrices, S symmetric and A antisymmetric.
In[69] := $Assumptions = {Element[*u*|*v*, Vectors[*d*, Reals]],
Element[*S*, Matrices[{*d*, *d*}, Reals, Symmetric[{1, 2}]]],
Element[*A*, Matrices[{*d*, *d*}, Reals, Antisymmetric[{1, 2}]]]};
In[70] := TensorReduce[*v*.*A*.(*u* + *v*)]
Out[70] = $-u.A.v$
In[71] := TensorReduce[*u*.*S*.*v* + *v*.*S*.*u*]
Out[71] = $2u.S.v$
The tensor product $S_{ij}A_{kl}$ is contracted in j and k and in i and l.
In[72] := TensorReduce[TensorContract[TensorProduct[*S*, *A*], {{2, 3}, {1, 4}}]]
Out[72] = 0

The Riemann curvature tensor has the properties $R_{ijkl} = -R_{jikl}$ and $R_{ijkl} = R_{klij}$.

In[73] := $Assumptions = Element[*R*, Arrays[{4,4,4,4}, Reals, {{{2,1,3,4},−1},{{3,4,1,2},1}}]]

Out[73] = $R \in$ Arrays[{4,4,4,4}, Reals, {{Cycles[{{1,2}}],−1}, {Cycles[{{1,3},{2,4}}],1}, {Cycles[{{3,4}}],−1}}]

In[74] := TensorReduce[TensorContract[*R*, {{1,2}}]]

Out[74] = 0

The Ricci tensor.

In[75] := R2 = TensorContract[*R*, {{1,3}}]

Out[75] = TensorContract[R, {{1,3}}]

In[76] := {TensorRank[R2], TensorDimensions[R2], TensorSymmetry[R2]}

Out[76] = {2, {4,4}, Symmetric[{1,2}]}

$R_{ijkl}R_{ijkl} + R_{ijkl}R_{kl ji} + R_{ijkl}R_{ikjl}$.

In[77] := TensorReduce[
TensorContract[TensorProduct[*R*,*R*], {{1,5},{2,6},{3,7},{4,8}}]+
TensorContract[TensorProduct[*R*,*R*], {{1,7},{2,8},{3,6},{4,5}}]+
TensorContract[TensorProduct[*R*,*R*], {{1,5},{2,7},{3,6},{4,8}}]]

Out[77] = TensorContract[$R \otimes R$, {{1,5},{2,7},{3,6},{4,8}}]

Unfortunately, *Mathematica* cannot take $R_{ijkl} + R_{iklj} + R_{iljk} = 0$ into account.

In[78] := $Assumptions = True;