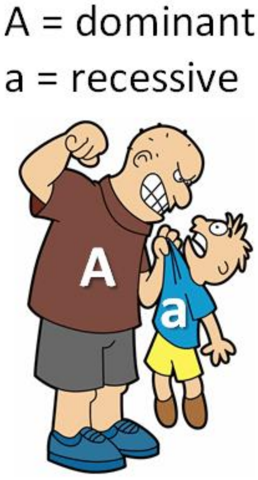
7. NUMERICAL LINEAR ALGEBRA

§7.1. The Power Method for Finding Dominant Eigenvalues

Finding eigenvalues, by first finding the characteristic polynomial, can be quite difficult for large matrices. Often we only want to find the dominant eigenvalue, that is, the one with largest absolute value.

The **dominant eigenvalue** of a matrix is a real eigenvalue that occurs with multiplicity 1 and whose absolute value exceeds the modulus of any other eigenvalue. A dominant eigenvector is an eigenvector for the **dominant eigenvalue** (if it exists). Clearly not all square matrices have a dominant eigenvalue.



Example 1: The matrix $A = \begin{pmatrix} 10 & 7 & -2 \\ 0 & 9 & -31 \\ 0 & 1 & -2 \end{pmatrix}$ has eigenvalues 10, $\dfrac{7 \pm \sqrt{3}i}{2}$. The non-real eigenvalues have modulus $\sqrt{13}$, so 10 is the dominant eigenvalue.

Example 2: If A is as above, then $A - 9I = \begin{pmatrix} 1 & 7 & -2 \\ 0 & 0 & -31 \\ 0 & 1 & -11 \end{pmatrix}$ has eigenvalues 1, $\dfrac{-11 \pm \sqrt{3}i}{2}$. (We just subtract 9 from the eigenvalues of A.) The non-real eigenvalues have modulus $\sqrt{31}$, so there is no dominant eigenvalue.

Theorem 1: Suppose A is a non-zero $n \times n$ diagonalisable real matrix with eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_k$ and suppose that λ_1 is the dominant eigenvalue. Let $\mathbf{e}_1, \ldots, \mathbf{e}_k$ be a corresponding basis of eigenvectors. Let $\mathbf{v}_0$ be any vector in $\mathbb{R}^n$ such that $\mathbf{v}_0 \notin \langle \mathbf{e}_2, \ldots, \mathbf{e}_n \rangle$.

For all n, let $\mathbf{v}_n = A^n \mathbf{v}_0$. Then $\dfrac{\langle A\mathbf{v}_n \mid \mathbf{v}_n \rangle}{|\mathbf{v}_n|^2} \to \lambda_1$ as $n \to \infty$.

Proof: Since A is a non-zero diagonalisable matrix it must have a non-zero eigenvalue and so $\lambda_1 \neq 0$.

Let $\langle \mathbf{e}_i \mid \mathbf{e}_j \rangle = e_{ij}$.

Since each $\mathbf{e}_i \neq 0$, $e_{ii} = |\mathbf{e}_i|^2 > 0$ for all i.
Now $\mathbf{v}_0 = x_1\mathbf{e}_1 + x_2\mathbf{e}_2 + ... + x_k\mathbf{e}_k$ for some real numbers x_1, x_2, ..., x_k where $x_1 \neq 0$.

Then $\mathbf{v}_n = \mathrm{A}^n\mathbf{v}_0 = x_1\,\lambda_1{}^n\,\mathbf{e}_1 + x_2\,\lambda_2{}^n\,\mathbf{e}_2 + ... + x_k\,\lambda_k{}^n\,\mathbf{e}_k$.
$\mathrm{A}\mathbf{v}_n = \mathrm{A}^{n+1}\mathbf{v}_0 = x_1\,\lambda_1{}^{n+1}\,\mathbf{e}_1 + x_2\,\lambda_2{}^{n+1}\,\mathbf{e}_2 + ... + x_k\,\lambda_k{}^{n+1}\,\mathbf{e}_k$.

So $$\frac{\langle \mathrm{A}\mathbf{v}_n \mid \mathbf{v}_n\rangle}{|\mathbf{v}_n|^2} = \frac{\sum_{i,j=1}^{k} x_ix_j\,\lambda_i{}^{n+1}\lambda_j{}^n\,e_{ij}}{\sum_{i,j=1}^{k} x_ix_j\,\lambda_i{}^n\lambda_j{}^n e_{ij}}$$

Here the sums are over all combinations of i, j.

$$= \frac{x_1{}^2\lambda_1{}^{2n+1}|\mathbf{e}_1|^2 + \sum_{ij\neq 1}^{k} x_ix_j\,\lambda_i{}^{n+1}\lambda_j{}^n\,e_{ij}}{x_1{}^2\lambda_1{}^{2n}|\mathbf{e}_1|^2 + \sum_{ij\neq 1}^{k} x_ix_j\,\lambda_i{}^n\lambda_j{}^n e_{ij}}$$

Here the sums are over all combinations of i, j *except* when $i = j = 1$. This has been separated out.

$$= \frac{\lambda_1 + \sum_{ij\neq 1}^{k} \left(\frac{x_i x_j}{x_i^2}\right)\lambda_i \left(\frac{\lambda_i \lambda_j}{\lambda_1^2}\right)^n \left(\frac{e_{ij}}{|\mathbf{e}_1|^2}\right)}{1 + \sum_{ij\neq 1}^{k} \left(\frac{x_i x_j}{x_1^2}\right)\left(\frac{\lambda_i \lambda_j}{\lambda_1^2}\right)^n \left(\frac{e_{ij}}{|\mathbf{e}_1|^2}\right)}$$

$$\to \lambda_1 \text{ as } n \to \infty.$$

This is because $\left|\frac{\lambda_i \lambda_j}{\lambda_1^2}\right| < 1$, and so $\left(\frac{\lambda_i \lambda_j}{\lambda_1^2}\right)^n \to 0$, except when $i = j = 1$. ✋☺

This theorem is the basis for the **Power Method**. We start by multiplying an initial vector **v** by A and compute ⟨A**v**|**v**⟩, $|\mathbf{v}|^2$ and their quotient, $q = \frac{\langle A\mathbf{v}|\mathbf{v}\rangle}{|\mathbf{v}|^2}$.

We’re going to repeatedly multiply **v** by *A*. Since the components of $A^n\mathbf{v}$ could become large, and since any non-zero scalar multiple of an eigenvector is an eigenvector, we scale A**v** by dividing it by the largest absolute value of its components. Having scaled A**v** we use this as the next **v** and continue.

If all goes well A**v** will converge to a dominant eigenvector and λ will converge to the dominant eigenvalue. For simplicity we set up a **Power Method Worksheet**.

POWER METHOD WORKSHEET Finds the dominant eigenvalue λ_Δ and eigenvector $\mathbf{v}_\Delta$					
A	**v**	**Av**	$\langle \mathbf{Av},\mathbf{v}\rangle$	$\lvert\mathbf{v}\rvert^2$	**q**
A	$\begin{pmatrix}1\\ \ldots\\ 1\end{pmatrix}$			n	$\frac{\langle A\mathbf{v}\vert\mathbf{v}\rangle}{\lvert\mathbf{v}\rvert^2}$
A	divide A**v** by largest component	→ $\mathbf{v}_\Delta$			→ λ_Δ
……	…………..	………	……..	….	………..

Example 3: Find the dominant eigenvalue of A = $\begin{pmatrix}3 & 1\\ 4 & 5\end{pmatrix}$

Solution:

A	**v**	**Av**	$\langle \mathbf{Av}\vert\mathbf{v}\rangle$	$\lvert\mathbf{v}\rvert^2$	**q**
$\begin{pmatrix}3 & 1\\ 4 & 5\end{pmatrix}$	$\begin{pmatrix}1\\ 1\end{pmatrix}$	$\begin{pmatrix}4\\ 9\end{pmatrix}$	13	2	6.5
$\begin{pmatrix}3 & 1\\ 4 & 5\end{pmatrix}$	$\begin{pmatrix}0.4444\\ 1\end{pmatrix}$	$\begin{pmatrix}2.3332\\ 6.7776\end{pmatrix}$	7.8144	1.1975	6.5256
$\begin{pmatrix}3 & 1\\ 4 & 5\end{pmatrix}$	$\begin{pmatrix}0.3443\\ 1\end{pmatrix}$	$\begin{pmatrix}2.0329\\ 6.3772\end{pmatrix}$	7.0771	1.1185	6.3273

$\begin{pmatrix}3 & 1\\4 & 5\end{pmatrix}$	$\begin{pmatrix}0.3178\\1\end{pmatrix}$	$\begin{pmatrix}1.9534\\6.2712\end{pmatrix}$	6.8920	1.1010	6.2598
$\begin{pmatrix}3 & 1\\4 & 5\end{pmatrix}$	$\begin{pmatrix}0.3115\\1\end{pmatrix}$	$\begin{pmatrix}1.9345\\6.246\end{pmatrix}$	6.8486	1.0970	6.2430
$\begin{pmatrix}3 & 1\\4 & 5\end{pmatrix}$	$\begin{pmatrix}0.3097\\1\end{pmatrix}$	$\begin{pmatrix}1.9291\\6.2388\end{pmatrix}$	6.8362	1.0959	6.2380
$\begin{pmatrix}3 & 1\\4 & 5\end{pmatrix}$	$\begin{pmatrix}0.3092\\1\end{pmatrix}$	$\begin{pmatrix}1.9276\\6.2368\end{pmatrix}$	6.8328	1.0956	6.2366
$\begin{pmatrix}3 & 1\\4 & 5\end{pmatrix}$	$\begin{pmatrix}0.3091\\1\end{pmatrix}$	$\begin{pmatrix}1.9273\\6.2364\end{pmatrix}$	6.8321	1.0955	6.2365
$\begin{pmatrix}3 & 1\\4 & 5\end{pmatrix}$	$\begin{pmatrix}0.3090\\1\end{pmatrix}$	$\begin{pmatrix}1.9270\\6.2360\end{pmatrix}$	6.8314	1.0955	6.2359

So our estimate of the dominant eigenvalue is 6.2359. Using the fact that the trace is 8 we can find the other eigenvalue to be 1.7641. Of course, we would never use the Power Method for a 2 × 2 matrix.

§7.2. Recessive Eigenvalues

Suppose we'd like to find other eigenvalues. The **recessive eigenvalue** (if it exists) of a matrix is a real eigenvalue that occurs with multiplicity 1 and whose absolute value is less than the modulus of any other eigenvalue. A **recessive eigenvector** is an eigenvector for the recessive eigenvalue (if it exists). Again, not all square matrices have a recessive eigenvalue.

Example 4: The matrix A = $\begin{pmatrix} 10 & 7 & -2 \\ 0 & 9 & -31 \\ 0 & 1 & -2 \end{pmatrix}$ has eigenvalues 10, $\dfrac{7 \pm \sqrt{3}i}{2}$.

The non-real eigenvalues have modulus $\sqrt{13}$, so this matrix does not have a recessive eigenvalue.

Example 5: The matrix $\begin{pmatrix} 1 & 7 & -2 \\ 0 & 0 & -31 \\ 0 & 1 & -11 \end{pmatrix}$ has eigenvalues 1, $\dfrac{-11 \pm \sqrt{3}i}{2}$. The non-real eigenvalues have modulus $\sqrt{31}$, so there is a recessive eigenvalue, namely 1.

Since the non-zero eigenvalues of A^{-1} are the reciprocals of the non-zero eigenvalues of A, the recessive eigenvalue of A is the reciprocal of the dominant eigenvalue of A^{-1}. So to find the recessive eigenvalue of A we proceed as follows.

TO FIND THE RECESSIVE EIGENVALUE OF A (Smallest in absolute value)

(1) Find A^{-1}. (If A is not invertible, the recessive eigenvalue is 0.)

(2) Use the Power Method to find the dominant eigenvalue, λ, of A^{-1}.

(3) Then $1/\lambda$ is the smallest eigenvalue of A.

Example 6: Find the recessive eigenvalue of

$$A = \begin{pmatrix} 212 & -172 & 39 \\ 462 & -372 & 84 \\ 882 & -702 & 159 \end{pmatrix}.$$

Solution: $A^{-1} = -\frac{1}{30}\begin{pmatrix} 6 & 1 & -2 \\ -21 & 23 & -7 \\ -126 & 76 & -20 \end{pmatrix}$.

So $B = 30A^{-1} = \begin{pmatrix} -6 & -1 & 2 \\ 21 & -23 & 7 \\ 126 & -76 & 20 \end{pmatrix}$.

This is the matrix to which we apply the Power Method.

B	**v**	**Bv**	**⟨Bv|v⟩**
$\begin{pmatrix}-6 & -1 & 2\\ 21 & -23 & 7\\ 126 & -96 & 20\end{pmatrix}$	$\begin{pmatrix}1\\1\\1\end{pmatrix}$	$\begin{pmatrix}-5\\5\\50\end{pmatrix}$	50
$\begin{pmatrix}-6 & -1 & 2\\ 21 & -23 & 7\\ 126 & -96 & 20\end{pmatrix}$	$\begin{pmatrix}-0.1\\0.1\\1\end{pmatrix}$	$\begin{pmatrix}2.5\\2.6\\-2.2\end{pmatrix}$	–2.19
$\begin{pmatrix}-6 & -1 & 2\\ 21 & -23 & 7\\ 126 & -96 & 20\end{pmatrix}$	$\begin{pmatrix}0.9615\\1\\-0.8462\end{pmatrix}$	$\begin{pmatrix}-8.4614\\-8.7319\\8.225\end{pmatrix}$	–23.8275

v	⟨B**v**|**v**⟩	$\lvert\mathbf{v}\rvert^2$	q
$\begin{pmatrix}1\\1\\1\end{pmatrix}$	50	3	16.6667
$\begin{pmatrix}-0.1\\0.1\\1\end{pmatrix}$	–2.19	1.02	–2.1471
$\begin{pmatrix}0.9615\\1\\-0.8462\end{pmatrix}$	–23.8275	2.6405	–9.0239

B	**v**	**Bv**	**⟨Bv|v⟩**
$\begin{pmatrix} -6 & -1 & 2 \\ 21 & -23 & 7 \\ 126 & -96 & 20 \end{pmatrix}$	$\begin{pmatrix} 0.9690 \\ 1 \\ -0.9419 \end{pmatrix}$	$\begin{pmatrix} -8.6978 \\ -9.2443 \\ 7.256 \end{pmatrix}$	–24.5069
$\begin{pmatrix} -6 & -1 & 2 \\ 21 & -23 & 7 \\ 126 & -96 & 20 \end{pmatrix}$	$\begin{pmatrix} 0.9409 \\ 1 \\ -0.7849 \end{pmatrix}$	$\begin{pmatrix} -8.2152 \\ -8.7354 \\ 6.8554 \end{pmatrix}$	–21.8459
$\begin{pmatrix} -6 & -1 & 2 \\ 21 & -23 & 7 \\ 126 & -96 & 20 \end{pmatrix}$	$\begin{pmatrix} 0.9404 \\ 1 \\ -0.7848 \end{pmatrix}$	$\begin{pmatrix} -8.2120 \\ -8.7452 \\ 6.7944 \end{pmatrix}$	–21.8000

v	**⟨Bv|v⟩**	$\|\mathbf{v}\|^2$	***q***
$\begin{pmatrix} 0.9690 \\ 1 \\ -0.9419 \end{pmatrix}$	–24.5069	2.8261	–8.6716
$\begin{pmatrix} 0.9409 \\ 1 \\ -0.7849 \end{pmatrix}$	–21.8459	2.5014	–8.7335
$\begin{pmatrix} 0.9404 \\ 1 \\ -0.7848 \end{pmatrix}$	–21.8000	2.5003	–8.7190

$\mathbf{B}$	$\mathbf{v}$	$\mathbf{Bv}$	$\langle\mathbf{Bv}\|\mathbf{v}\rangle$
$\begin{pmatrix} -6 & -1 & 2 \\ 21 & -23 & 7 \\ 126 & -96 & 20 \end{pmatrix}$	$\begin{pmatrix} 0.9390 \\ 1 \\ -0.7769 \end{pmatrix}$	$\begin{pmatrix} -8.1878 \\ -8.7193 \\ 6.7760 \end{pmatrix}$	–21.6719
$\begin{pmatrix} -6 & -1 & 2 \\ 21 & -23 & 7 \\ 126 & -96 & 20 \end{pmatrix}$	$\begin{pmatrix} 0.9390 \\ 1 \\ -0.7771 \end{pmatrix}$	$\begin{pmatrix} -8.1882 \\ -8.7207 \\ 6.7720 \end{pmatrix}$	–21.6719
$\begin{pmatrix} -6 & -1 & 2 \\ 21 & -23 & 7 \\ 126 & -96 & 20 \end{pmatrix}$	$\begin{pmatrix} 0.9389 \\ 1 \\ -0.7765 \end{pmatrix}$	$\begin{pmatrix} -8.1864 \\ -8.7186 \\ 6.7714 \end{pmatrix}$	–21.6628

$\mathbf{v}$	$\langle\mathbf{Bv}\|\mathbf{v}\rangle$	$\|\mathbf{v}\|^2$	$\boldsymbol{q}$
$\begin{pmatrix} 0.9390 \\ 1 \\ -0.7769 \end{pmatrix}$	–21.6719	2.4853	–8.7200
$\begin{pmatrix} 0.9390 \\ 1 \\ -0.7771 \end{pmatrix}$	–21.6719	2.4856	–8.7190
$\begin{pmatrix} 0.9389 \\ 1 \\ -0.7765 \end{pmatrix}$	–21.6628	2.4845	–8.7192

So we have found that the dominant eigenvalue of $B = 30A^{-1}$ is approximately –8.7192.

Hence the dominant eigenvalue of A^{-1} is approximately $-\frac{8.7192}{30} = -0.29064$.

Therefore the recessive eigenvalue of A is approximately $-\frac{1}{0.29064} = -3.4407$.

The vector $\begin{pmatrix} -8.1864 \\ -8.7186 \\ 6.7714 \end{pmatrix}$ is an eigenvector for A, as well as A^{-1} and $30A^{-1}$ for this eigenvalue.

§7.3. Finding Nearest Eigenvalues

We can adapt the above methods to the problem of finding other eigenvalues. Since the eigenvalues of A – kI are the eigenvalues of A, minus k, if we want the eigenvalue nearest to the real number k we find the recessive eigenvalue of A – kI.

TO FIND THE EIGENVALUE CLOSEST TO *k*

(1) Write down B = A – *k*I.

(2) Compute B^{-1}. (If A – *k*I is not invertible then *k* is itself an eigenvalue.)

(3) Use the Power Method to find the dominant eigenvalue, λ, of B^{-1}.

(4) Then λ^{-1} is the recessive eigenvalue for B.

(5) Then $\lambda^{-1} + k$ is the nearest eigenvalue to *k* for A.

Note that this method will not work for repeated eigenvalues, or conjugate pairs of eigenvalues.

Example 7: Find the eigenvalue of

$$A = \begin{pmatrix} 16 & 3 & 2 & 13 \\ 5 & 10 & 11 & 8 \\ 9 & 6 & 7 & 12 \\ 4 & 15 & 14 & 1 \end{pmatrix}$$

that is closest to 10, and find a corresponding eigenvector.

Solution: Let B = A – 10I =. $\begin{pmatrix} 10 & 3 & 2 & 13 \\ 5 & 0 & 11 & 8 \\ 9 & 6 & -3 & 12 \\ 4 & 15 & 14 & -9 \end{pmatrix}$

$$B^{-1} = \frac{1}{1440}\begin{pmatrix} -601 & 287 & 423 & -49 \\ 303 & -201 & =129 & 87 \\ 47 & 71 & -81 & 23 \\ 311 & -97 & -153 & -1 \end{pmatrix}.$$

$$\text{Let } C = 1440B^{-1} = \begin{pmatrix} -601 & 287 & 423 & -49 \\ 303 & -201 & =129 & 87 \\ 47 & 71 & -81 & 23 \\ 311 & -97 & -153 & -1 \end{pmatrix}.$$

Using the Power Method on C, starting with $\mathbf{v} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$ we get –720 as the dominant eigenvalue for C.

$\therefore\ -\dfrac{720}{1440} = -0.5$ is the dominant eigenvalue for B^{-1}.

$\therefore$ –2 is the recessive eigenvalue for B = A – 10I.

$\therefore$ 8 is the nearest eigenvalue to 10, for A.

For a corresponding eigenvector we may take a dominant eigenvector for C, namely $\begin{pmatrix} 1 \\ -0.5 \\ 0 \\ -0.5 \end{pmatrix}$, or more simply, $\begin{pmatrix} 2 \\ -1 \\ 0 \\ -1 \end{pmatrix}$

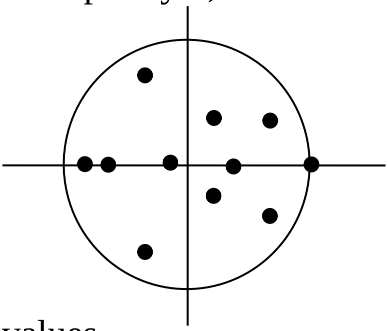
§7.4. Perron Matrices

If the Power Method converges it will usually give the dominant eigenvalue and eigenvector. The exception is when you pick a starting vector that just happens to be in the subspace spanned by the eigenvectors for the other eigenvalues. This is quite unlikely but you must be aware that it can happen. You could try a second starting vector if you were really worried.

But the method may not even converge. It may not converge for a couple of reasons. The *q* values estimate may go to infinity. This will happen if the eigenvalue with largest modulus is repeated. Or it may jump around. This will happen when there is more than one eigenvalue with largest modulus, or one repeated value.

It would be useful to know in advance whether the method should work. Such matrices will have one positive real eigenvalue, with multiplicity 1, that exceeds the modulus of all other eigenvalues.

A **Perron matrix** is a real square matrix that has a positive real eigenvalue R, with multiplicity 1, such that the modulus of all other eigenvalues is strictly less than R.



This is precisely the type of matrix for which the Power Method works.

The **spectrum** of a matrix A is the set of its eigenvalues. It's denoted by $\boldsymbol{\sigma(A)}$. The **spectral radius** of *A* is the largest modulus of the eigenvalues of A. So a Perron matrix has a spectral radius R where R is the dominant eigenvalue.

Example 8: Find the spectrum and spectral radius of

$$A = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \\ 3 & 4 & 1 & 2 \\ 4 & 1 & 2 & 3 \end{pmatrix}.$$

Solution: From example 4,
$\sigma(A) = \{10, -2 + 2i, -2 - 2i, 2\}$.
The spectral radius is 10. Note that $|-2 \pm 2i| = \sqrt{8} < 10$.

A real matrix A = (a_{ij}) is called a **non-negative** matrix if each $a_{ij} \geq 0$. We denote this by writing $\mathbf{A \geq 0}$. We call A a **positive** matrix if each $a_{ij} > 0$. This we denote by $\mathbf{A > 0}$.

Theorem 2 (PERRON): Positive matrices are Perron matrices.
Proof: We omit the proof here. ✋

Often a matrix is not a positive matrix but has a power that is. In most cases this shows that the original matrix is a Perron matrix.

Theorem 3: If $A^N > 0$ for some odd positive integer N then A is a Perron matrix.

Proof: Suppose $\sigma(A) = \{\lambda_1, \lambda_2, ..., \lambda_r\}$.

Then $\sigma(A^N) = \{\lambda_1^N, \lambda_2^N, ..., \lambda_r^N\}$.

Since A^N is positive, one of these eigenvalues, say λ_1^N, is real and for all $i \geq 2$, $|\lambda_i|^N < \lambda_1^N$.

Then for all $i \geq 2$, $|\lambda_i| < |\lambda_1|$.

Clearly λ_1 must have multiplicity 1.

Suppose λ_1 is non-real. Then its conjugate, say λ_2 would be also an eigenvalue.

But that would give $|\lambda_2|^N = |\lambda_1|^N$, a contradiction.

Hence λ_1 is real.

If $\lambda_1 < 0$ then $\lambda_1^N < 0$, a contradiction (remember that N is odd). ✋☺

The requirement that N be odd is necessary, as the following example shows.

Example 9: If A = $\begin{pmatrix} -1 & -\sqrt{2} \\ -\sqrt{2} & 0 \end{pmatrix}$ then

$A^2 = \begin{pmatrix} 3 & \sqrt{2} \\ \sqrt{2} & 2 \end{pmatrix} > 0.$

$\chi_{A}{}^{2}(\lambda) = \lambda^2 - 5\lambda + 4$

$\qquad = (\lambda - 1)(\lambda - 4).$

Hence $\sigma(A^2) = \{1, 4\}$ and so A^2 is a Perron matrix.

However $\chi_A(\lambda) = \lambda^2 + \lambda - 2$

$\qquad\qquad = (\lambda - 1)(\lambda + 2).$

So $\sigma(A) = \{1, -2\}$ and so A is not a Perron matrix.

But, for non-negative matrices, any power that is positive is sufficient to ensure that the original matrix is a Perron matrix.

Theorem 4: If $A \geq 0$ and $A^N > 0$ then $A^M > 0$ for all $M \geq N$.

Proof: Let $B = A^N$. Then the *i-j* component of $A^{N+1} = \sum_k a_{ik}b_{kj}$. Each term is non-negative, and can only be zero if $a_{ik} = 0$ for all k. But then the i'th row of every power of A would be zero, a contradiction. ✋☺

Corollary: If $A \geq 0$ and $A^N > 0$ for some N then A is a Perron matrix.

§7.5. Computational Complexity

An **algorithm** is a computational procedure for finding the answer to some mathematical (or other) problem. Mostly they are implemented on a computer, but simple algorithms are still carried out using just pencil and paper. There are still some people alive who don't need to reach for their calculators to check the addition on a restaurant bill! Addition and multiplication of integers can be carried out by hand using algorithms that are still taught in schools.

The two most important requirements are:
(1) it should terminate in finite time;
(2) it should always give the correct answer.

Beyond this are considerations of storage space in the computer and computational time. We have seen two methods for evaluating a determinant, the first order expansion and the second order expansion. Which is the best one to use in practice? We have seen two methods for finding the inverse of a matrix, using cofactors and reducing (A | I) to (I | A^{-1}) by elementary row operations. Which one is computationally more efficient?

First there is the consideration of storage space. The very early computers had very little storage space and programmers bent over backwards to use every single byte efficiently. A bit is the amount of storage space required to store a 0 or a 1. A byte is 8 bits and is capable of storing a number from 0 to 255. Bytes can be used together to store larger integers or real numbers (up to a certain number of significant figures). The first computer I ever programmed was at a time when there were only 6 computers in the whole of Australia. This one had 16K of memory and took up a whole room! Even a simple program could easily use this up very quickly.

The first personal computer I ever owned had only 1K of memory. It used a TV screen for its display and about a quarter of that precious memory was reserved for storing the screen display. Clever programmers on these machines soon learnt how to get access to the display memory and frequently used it to do intermediate computations before displaying the final answers. You could see this happening as the screen flashed all sorts of nonsense on the screen during the calculation before finally displaying the answers at the end.

These days are long gone. Memory is virtually unlimited. It’s not unusual to have, on an ordinary personal computer, 128 gigabytes of memory. That is about 128 thousand million bytes. Programs frequently use the hard disk as virtual memory, and here several terabytes are not uncommon. That’s about a million million bytes.

Apart from the ready availability of memory most alternative algorithms for performing the same calculation use roughly the same amount of memory. In comparing algorithms you can forget the amount of memory used.

This leaves speed. We want to choose an algorithm that performs faster than all its competitors. Now speed depends on hardware. Some computers are much faster than others, but no computer is as fast as its programmers would like. Speeds have increased over the years, but nowhere to the extent that memory has.

In comparing alternative algorithms we don't need to bother about the speed of the hardware. Whichever algorithm we use it will be used on a specific piece of hardware. What is important is the number of steps. We want to make this as little as possible.

Moving data around, such as when we swap two rows in a matrix, is almost instantaneous. Addition and subtraction take about the same amount of time. Multiplication and division often take somewhat longer, though on some computers they take about the same time.

More complicated mathematical functions, such as square roots or trigonometric functions, use power series expansions and need to be analysed separately. We won't be using any of these here, so we'll just count the total number of multiplications and divisions. For simplicity we'll assume that both of these take the same amount of time. In a test on a certain laptop using a certain programming language, each took about 10 microseconds.

The input to an algorithm will vary, and the number of steps (by which we mean just multiplication and division) will vary with the size of the input. Evaluating a 100 × 100 determinant will take many more steps than for a 3 × 3 determinant, no matter which algorithm we use. So we need to measure the size of the input and work out the number of steps as a function of this size. This function we call the **complexity function**. If we denote it by Γ then $\Gamma(n)$ is the number of steps the algorithm takes with input of size n.

The size of an input can be defined in many different ways and hence the complexity function depends on such a definition. For example we might define the size of an $n \times n$ matrix to be just n. Or we might define it to be n^2, the number of components. Perhaps we might define it to be the number of bytes required to store the input, which will be a constant multiple of n^2.

If the input is a positive integer N, we might define N to be the size of the input. Or we might define it to be the number of digits, which will be approximately $\log_{10}N$.

One of the problems in determining the complexity function of an algorithm is that for many algorithms inputs of similar sizes might take dramatically different times to perform. For example, factorising a positive integer N by the most obvious method of testing for divisibility by all the numbers from 2 up to $\sqrt{N}$ would take about $\sqrt{N}$ divisions if N is prime and we had to go all the way up to $\sqrt{N}$.

On the other hand if $N = 2^n$ for some n, we'd find factors of 2 at each stage and the algorithm would take about n steps, that is $\log_2 N$ steps which is very much better than √N. For such algorithms we'd need to estimate the average number of steps and this can get very complicated.

Now for factorisation Γ(n) = √N, if we take the worst case. This doesn't sound too bad for a complexity function. But that is because we're taking the number itself as a measure of size. A more appropriate measure would be the number of digits of N. This value will be about $n = \log_{10}\mathrm{N}$ and in terms of n, $\Gamma(n) = \sqrt{10^n}$ which is roughly $3^{n/2}$. Such a complexity function is called **exponential**. An algorithm with an exponential complexity function is considered to be of limited use.

There are improvements to the simple factorisation algorithm, but these only make minor reductions to the number of steps. Essentially factorisation of integers is considered to be an **intractable** problem, that is, algorithms *do* exist to solve the problem but the number of steps grows so fast with the input size that it is only feasible to use it on relatively small input. Factorising an arbitrary 200 digit number has been done but it has taken a large number of very fast computers many months to perform just a single factorisation.

Let's turn our attention to the problem of evaluating determinants, comparing the first order method with the second order method.

First Order Method: Let a_n be the number of steps in evaluating an $n \times n$ determinant by the first order method, running along the first row. We have to evaluate n determinants of size $(n-1) \times (n-1)$. Each involves a_{n-1} steps so that means na_{n-1} steps in all. Then each determinant is multiplied by plus or minus the entry in the first row, giving an extra n steps. In all we have $na_{n-1} + n$ steps.

So we get the recurrence equation for a_n, the number of steps in evaluating an $n \times n$ determinant by the First Order Method:

$$a_1 = 0;$$
$$a_n = n(a_{n-1} + 1).$$

Second Order Method: Let b_n be the number of steps in evaluating an $n \times n$ determinant by this method. Clearly $b_1 = 0$ and $b_2 = 3$. We take each of the $\binom{n}{2}$ pairs of columns. For each one we must compute an $(n-2) \times (n-2)$ determinant, taking b_{n-2} steps. We then multiply this by a certain 2×2 determinant, taking an extra 4 steps – three to calculate the 2 × 2 determinant and one to multiply the two determinants which, in all takes $b_{n-2} + 4$ steps.

This has to be done $\binom{n}{2}$ times before these values are combined to give the answer.
So we get the recurrence equation:

$$b_1 = 0;$$
$$b_2 = 2;$$
$$b_n = \binom{n}{2}(b_{n-2} + 4)\ .$$

For the Echelon Form Method (reduce to echelon form and multiply by the diagonal elements) the approximate number of multiplications and divisions is given by:

$$c_n = \frac{1}{6}(4n^3 - 3n^2 + 5n - 6).$$

n	**1st Order Expansion**	**2nd Order Expansion**	**Echelon Form**
1	0	0	0
2	2	2	4
3	9	9	15
4	40	30	37
5	205	120	74
6	1236	495	130

7	8,659	2,583	209
8	69,280	13,940	315

So the second order expansion is substantially quicker than the first order expansion. But the Echelon Form Method is enormously faster. I doubt if there’s a computer on earth that could compute a 1000 × 1000 determinant by any other method than the Echelon Form one.

Remember that I introduced the second order expansion method not because it’s quicker but as an aid to proving theorems about determinants.

Good algorithms are considered to be those where $\Gamma(n)$ is a polynomial. These are said to operate in **polynomial time**. Bad algorithms are those where $\Gamma(n)$ is **exponential**, or worse. Of course for some problems there are algorithms but no ‘good’ ones. Fortunately most problems in linear algebra can be solved by polynomial time algorithms.

EXERCISES FOR CHAPTER 7

Exercise 1: Use the Power Method to find the dominant eigenvalue, if one exists, for the following matrices:
(i) $\begin{pmatrix} 5 & 1 \\ 3 & 2 \end{pmatrix}$; (ii) $\begin{pmatrix} 2 & -1 \\ 4 & 1 \end{pmatrix}$; (iii) $\begin{pmatrix} 8 & 2 \\ 6 & 3 \end{pmatrix}$; (iv) $\begin{pmatrix} 5 & -2 \\ 0 & 3 \end{pmatrix}$.

Exercise 2: Use the Power Method to find the dominant eigenvalue, if one exists, for the following matrices:
(i) $\begin{pmatrix} 5 & 1 & 4 \\ 6 & 2 & -7 \\ -1 & 0 & 3 \end{pmatrix}$; (ii) $\begin{pmatrix} 6 & 1 & -1 \\ 0 & 8 & 13 \\ 4 & -6 & 5 \end{pmatrix}$; (iii) $\begin{pmatrix} 2 & 3 & -4 \\ -6 & 1 & 2 \\ 0 & 8 & 7 \end{pmatrix}$.

Exercise 3: Use the Power Method to find the dominant eigenvalue, if one exists, for the following matrices:
(i) $\begin{pmatrix} 1 & 1 & 1 & 5 \\ 2 & 1 & 4 & 0 \\ 3 & 2 & 5 & 1 \\ 0 & 1 & 4 & 2 \end{pmatrix}$; (ii) $\begin{pmatrix} 1 & 4 & 6 & 0 \\ 3 & 1 & 7 & 2 \\ 4 & 3 & 0 & 2 \\ 1 & 4 & 5 & 3 \end{pmatrix}$.

Exercise 4: Find the recessive eigenvalue, if one exists, of the following matrices:

(i) $\begin{pmatrix} 5 & 1 \\ 3 & 2 \end{pmatrix}$; (ii) $\begin{pmatrix} 5 & 1 & 4 \\ 6 & 2 & -7 \\ -1 & 0 & 3 \end{pmatrix}$; (iii) $\begin{pmatrix} 1 & 1 & 1 & 5 \\ 2 & 1 & 4 & 0 \\ 3 & 2 & 5 & 1 \\ 0 & 1 & 4 & 2 \end{pmatrix}$.

Exercise 5: Find the nearest eigenvalue to 5 for the matrix

$$A = \begin{pmatrix} 5 & -2 & 1 \\ -2 & 7 & 0 \\ 1 & 0 & 6 \end{pmatrix}.$$

SOLUTIONS FOR CHAPTER 7

Exercise 1: (i) 5.7913; (ii) no dominant eigenvalue; (iii) 9.7720; (iv) 5 (BEWARE: if you start with $\mathbf{v} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ the method will give the value 3. This is because this is an inappropriate starting vector.)

Exercise 2: (i) 5.8063; (ii) no dominant eigenvalue; (iii) 10.3768.

Exercise 3: (i) 8.9165; (ii) 11.1062.

Exercise 4: (i) The inverse is B = $\frac{1}{7}\begin{pmatrix} 2 & -1 \\ -3 & 5 \end{pmatrix}$.

Applying the Power Method to $\begin{pmatrix} 2 & -1 \\ -3 & 5 \end{pmatrix}$ we get the dominant eigenvalue of 5.7913.

The dominant eigenvalue of B is $\frac{5.7913}{7}$ = 0.8273.

Hence the recessive eigenvalue of the original matrix is $\frac{1}{0.8273}$ = 1.2088.

Note that in exercise 1 we found the dominant eigenvalue to be 5.7913.

The sum of these eigenvalues is 7.0001 which is very close to the trace.

(ii) The inverse is B = $\frac{1}{27}\begin{pmatrix} 6 & -3 & -15 \\ -11 & 19 & 59 \\ 2 & -1 & 4 \end{pmatrix}$.

Applying the Power Method to $\begin{pmatrix} 6 & -3 & -15 \\ -11 & 19 & 59 \\ 2 & -1 & 4 \end{pmatrix}$ we find that it doesn't have a dominant eigenvalue so the original matrix doesn't have a smallest. However all is not lost. We know from exercise 1 that the dominant eigenvalue is 5.7913 and the trace is 10, so the sum of the other two eigenvalues is 10 – 5.7913 = 4.1937. If the two remaining are real they must be both equal to $\frac{4.1937}{2}$ = 2.09685. The product of these eigenvalues is 25.4630. Clearly this can't be the case since the determinant must be an integer. In fact the determinant is 27.

So the remaining two eigenvalues must be $a \pm b$i where a, b are real. Clearly $a = 2.09685$.

Since the product of the eigenvalues must be 27 we have

$a^2 + b^2 = \frac{27}{5.7913} = 4.6622$.

So $b^2 = 4.6622 - 2.09685^2 = 0.2654$.

This gives $b = \pm 0.5152$. So in fact we have found all three eigenvalues.

(iii) The inverse is B = $\frac{1}{12}\begin{pmatrix} 3 & 9 & -3 & -6 \\ -6 & -30 & 22 & 4 \\ 0 & 6 & -4 & 2 \\ 3 & 3 & -3 & 0 \end{pmatrix}$.

Applying the Power Method to $\begin{pmatrix} 3 & 9 & -3 & -6 \\ -6 & -30 & 22 & 4 \\ 0 & 6 & -4 & 2 \\ 3 & 3 & -3 & 0 \end{pmatrix}$ we get the dominant eigenvalue of –33.0680.

The dominant eigenvalue of *B* is $-\frac{33.0680}{12} = -2.7557$.

Hence the smallest eigenvalue of the original matrix is $-\frac{1}{2.7557} = -0.3629$.

Exercise 5: Let B = A – 5I = $\begin{pmatrix} 0 & -2 & 1 \\ -2 & 2 & 0 \\ 1 & 0 & 1 \end{pmatrix}$.

$B^{-1} = \frac{1}{6}\begin{pmatrix} -2 & -2 & 2 \\ -2 & 1 & 2 \\ 2 & 2 & 4 \end{pmatrix}$.

The dominant eigenvalue of $6B = \begin{pmatrix} -2 & -2 & 2 \\ -2 & 1 & 2 \\ 2 & 2 & 4 \end{pmatrix}$

is 5.1394.

The dominant eigenvalue of B^{-1} is $\frac{5.1394}{6}$ = 0.8566.

The recessive eigenvalue of A – 5I is therefore

$\frac{1}{0.8566}$ = 1.1674.

The closest eigenvalue of A to 5 is 6.1674.

coopersnotes.net
LIST OF TITLES

GENERAL

- The Mathematics At The Edge Of The Rational Universe

ELEMENTARY

- Basic Mathematics
- Concepts of Algebra
- Concepts of Calculus
- Elementary Algebra
- Elementary Calculus

1st YEAR UNI

- Techniques of Algebra
- Techniques of Calculus
- Matrices

2nd YEAR UNI

- Linear Algebra
- Languages & Machines
- Discrete Mathematics

3rd YEAR UNI

- Group Theory volume 1
- Group Theory volume 2
- Galois Theory
- Graph Theory
- Number Theory
- Geometry
- Topology
- Set Theory

POSTGRADUATE

- Ring Theory
- Representation Theory
- Quadratic Forms
- Group Tables vol 1
- Group Tables vol 2