Application of Control Tec niques to Solving Linear Systems of Equations

Mofozul Ali

September 1994

Submitted to :

The University of Reading

Department of Mathematics

in partial fulfilment of the requirements for the

Degree of Master of Science

Abstract

Standard iteration techniques for solving linear systems are discussed. We then analyse what effect the eigenvalues and the eigenvectors of the iteration matrix have on the convergence of the iteration process. Finally, we propose an application of Control Theory to determine the eigenvalues and the eigenvectors of the iteration matrix in order to ensure rapid convergence.

Ch pter 1

Introduction

In this dissertation we study numerical methods for solving linear systems of the form :

$$A\mathbf{x} = \mathbf{b} \tag{1.1}$$

where A is a given real $n \times n$ matrix and b is a given real column vector of order n.

For large sparse systems iterative techniques are very efficient in terms of computer storage and time requirements. Systems of this type arise in the numerical solution of boundary-value problems and partial-differential equations.

We discuss standard iteration procedures and analyse the effect on convergence of assigning :

a) eigenvalues

b) eigenvectors

to the iteration matrix.

The analysis, then enables us to use an application of Control Theory to assign

In this chapter we are going to study numerical methods for solving linear systems of the form :

$$A\mathbf{x} = \mathbf{b} \tag{2.1}$$

where A is a given real $n \times n$ matrix and b is a given real column vector of order n. The solution vector exists and is unique if and only if A is nonsingular. The solution is given by :

$$= A^{-1} \quad . \tag{2.2}$$

quation (2.1) can be solved using such as L-U decomposition or Gaussian elimination. Methods of this type require A to be factorized and are impractical if A is large and sparse. are an alternative to direct methods and are ideally suited to inverting large sparse matrices. These methods involve an initial approximation ^{0} to the solution , and generate a

This defines the Jacobi iteration for the case when $n = 3$. For general n we have

$$x_i^{(k+1)} = [b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}]/a_{ii} \qquad for \quad i = 1, 2, ..., n. \qquad (2.3)$$

Alternatively, we can obtain the matrix iteration form by splitting A into its diagonal and off-diagonal parts. If we let be the diagonal of A, and $-L$ and $-U$ be the strictly lower and upper triangular parts of A, then $A =$ $- L - U$.

quation (2.1) now becomes :

$$(\quad - L - U)\mathbf{x} = \mathbf{b}$$

$$\mathbf{x} = (L + U)\mathbf{x} + \mathbf{x}$$

$$\mathbf{x} = \quad ^{-1}(L + U)\mathbf{x} + \quad ^{-1}\mathbf{b}$$

Therefore, the matrix form of the iteration is :

$$\mathbf{x}^{(k)} = \quad ^{-1}(L + U)\mathbf{x}^{(k-1)} + \quad ^{-1}\mathbf{b}, \qquad k = 1, 2, ... \qquad (2.4)$$

Generally, quation (2.3) is used in computation.

Example 1: Solve the following linear system:

$$\begin{pmatrix} 10 & -1 & 2 & 0 \\ -1 & 11 & -1 & 3 \\ 2 & -1 & 10 & -1 \\ 0 & 3 & -1 & 8 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 6 \\ 25 \\ -11 \\ 15 \end{pmatrix}$$

A simple program was written in Fortran to carry out the Jacobi iteration given by quation (2.3), with the initial vector $^0 =$. The results obtained are shown in Table 1.

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$x_4^{(k)}$
0	0.00000000	0.00000000	0.00000000	0.00000000
1	0.60000000	2.27272727	−1.10000000	1.87500000
2	1.04727273	1.71590909	−0.80522727	0.88522727
3	0.93263636	2.05330579	−1.04934091	1.13088068
4	1.01519876	1.95369576	−0.96810863	0.97384272
5	0.98899130	2.01141473	−1.01028590	1.02135051
6	1.00319865	1.99224126	−0.99452174	0.99443374
7	0.99812847	2.00230688	−1.00197223	1.00359431
8	1.00062513	1.99867030	−0.99903558	0.99888839
9	0.99967415	2.00044767	−1.00036916	1.00061919
10	1.00011860	1.99976795	−0.99982814	0.99978598

The decision to stop after ten iterations is based on

$$\frac{^{(10)} \quad ^{(9)}}{^{(10)}} \quad < \quad 7 \quad 10^{-4}$$

The exact solution is, in fact, $= (1x2 \quad 1 \; 1)^T$. $s \;\; m \;\; m \qquad X \quad C \;\; C \qquad X \qquad C \qquad$

$(k+1)$ (k) $(k+1)$
i 1 2

$(k+1)$
1

$i-1$ n
$(k+1)$ $(k+1)$ (k)
i i ij j ij j ii
j=1 j=i+1

(k) -1 $(k-1)$ -1

ii

1

2

3

4

0

be overcome by . The Preconditioned Conjugate Gradient method uses a preconditioner to accelerate convergence [Golub and Van Loan].

Input $_0$, $_0$. Set $_0 =$ $_0$ $_{-1} = _0$.

Input , the preconditioner, choose $=$ for Jacobi preconditioning.

For $= 0\ 1\ 2$ until convergence, do

$_k = _k$

$_k = {}^T_k\ _k\ {}^T_{k-1}\ _{k-1}$

$_k = _k + _k\ _{k-1}$

$_k = {}^T_k\ _k\ {}^T_k\ _k$

$_{k+1} = _k + _k\ _k$

$_{k+1} =$ $_k$ $_k$ $_k$ $_k$

Use Jacobi-Preconditioned Conjugate Gradient method to solve the following linear system :

$$\begin{bmatrix} 10 & 1 & 2 & 0 \\ 1 & 11 & 1 & 3 \\ 2 & 1 & 10 & 1 \\ 0 & 3 & 1 & 8 \end{bmatrix} \begin{bmatrix} _1 \\ _2 \\ _3 \\ _4 \end{bmatrix} = \begin{bmatrix} 6 \\ 25 \\ 11 \\ 15 \end{bmatrix}$$

Table 3: Preconditioned C.G. method

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$x_4^{(k)}$
0	0.00000000	0.00000000	0.00000000	0.00000000
1	0.46461746	1.75991463	−0.85179867	1.45192957
2	1.04214099	1.94075677	−0.91679619	1.12828630
3	1.00488813	1.98938063	−1.01181525	1.00690572
4	1.00000000	2.00000000	−1.00000000	1.0000000

The above method converges after only four iterations and attains the precise solution to machine accuracy.

The Preconditioned Conjugate Gradient Method is much more efficient than either of the two methods introduced before it, yet I have discussed the former two in more detail. This is because the those methods have a very similar form to the control problem we shall encounter in Chapter 3. The following two theorems give the conditions for convergence of the three methods,

Theorem :

If A is strictly diagonally dominant, then both the Jacobi and the Gauss-Siedel methods converge for any choice of $\mathbf{x}^0$.

Theorem : Covergence of the Preconditioned Conjugate Gradient Method is guaranteed if A is symmetric, positive definite and M is also positive definite.

A symmetric matrix A is called **positive definite** if $\mathbf{x}^T A\mathbf{x} > 0$ for every $\mathbf{x} \neq \mathbf{0}$.

An $n \times n$ matrix A is said to be **strictly diagonally dominant** if :

$$|a_{ii}| > \sum_{j=1}^{n} |a_{ij}| \qquad (j \neq i)$$

holds for each $i = 1, 2, ..., n$.

The proof of these theorems are omitted but can be found in Young (1971).

2.4 General Iteration Method

If we multiply the equation $A\mathbf{x} = \mathbf{b}$ by a matrix G, then $GA\mathbf{x} = G\mathbf{b}$

and we may write $\mathbf{x} = (I - GA)\mathbf{x} + G\mathbf{b}$

So a general iteration process is :

$$\mathbf{x}^{n+1} = (I - GA)\mathbf{x}^n + G\mathbf{b} \tag{2.7}$$

where G is to be chosen.

It can easily be seen that :

$G =$ $^{-1}$ gives Jacobi Iteration method

$G = ($ $- L)^{-1}$ gives Gauss-Siedel Iteration method

The convergence of this method is dependent on the eigenvalues of $(I - GA)$.

Theorem :

The iterative method, quation (2.7), is convergent if and only if :

$$\rho\,(I - GA) \quad < \quad 1$$

The proof of this theorem is omitted, but can be found in e.g. Young (1971).

Rates of Convergence : (Young (1971)

The *average rate of convergence* is defined by

$$R_n(I - GA) = -(\frac{1}{n})log\,||(I - GA)^n||$$

The is defined by

$$R(I \quad GA) = \lim_{n\to\infty} R_n(I \quad GA) = \quad log \ \rho\,(I \quad GA)$$

We shall refer to $R_n(I \quad GA)$ as the , and see that, eventually, it tends to the asymptotic rate.

The convergence rate of the iteration process depends on the spectrum (i.e. the eigenvalues) of $(I \quad GA)$. Our aim is to choose G such that the eigenvalues of $(I \quad GA)$ are near the origin, so that we have a better iteration method than the three presented above.

Since the matrices and have full rank, there are well known theorems in Control Theory, which state that the pair ( ) is always

. This means that there always exists a matrix , which assigns ( ) any specified eigenvalues and eigenvectors, i.e. we can find a matrix such that

$$(\qquad) = \quad ^{-1}\Lambda \qquad (2\,8)$$

where $\Lambda =$ ($_1$ $_2$ $_n$) are the eigenvalues to be assigned, and is any nonsingular matrix representing the eigenvectors.

In Chapter 3, we will discuss this result in more detail and present an algorithm for finding . But before that we are going to look at how affects convergence. We have already stated that the rate of covergence tends to the asymptotic rate which is determined by ( ). An efficient iteration method will achieve rapid early convergence and so we are going to study how , which determines the eigenstructure of ( ), affects covergence in the early stages of the iteration process.

In order to see what effect the range of eigenvalues, or the condition number of

the eigenvector matrix has on convergence, we re-arrange equation (2.8) to get :

$$= (\quad {}^{-1}\Lambda \quad)^{-1} \qquad (2\,9)$$

Note that the calculation of requires inversion of , and if we are able to perform that operation accurately then the solution to our main problem would be easily found from equation (2.2). But, in our inverse problem, equation (2.9) gives us an expression to find , which we can then use in our General Iteration method, given by equation (2.7), to examine convergence.

First of all we study how the spectrum of ( ) affects convergence. Since we are able to assign any eigenvectors to the matrix, in this case we choose , so that the condition number of , ( ) = 1. equation (2.9) now becomes :

$$= (\quad \Lambda)^{-1} \qquad (2\,10)$$

This equation allows us to find , for given eigenvalues. A program has been written in MATLAB, which possesses internal functions to perform matrix multiplications and inversions, to calculate and then carry out the general iteration method [equation (3.7)].

$\Lambda =$ (0 1 0 05 0 05 0 1)

k	*convergence ra e*
1	2.9957
2	2.3026
3	2.3026
4	2.3026

Notice that (0 1) = 2 3026, so after just two iterations the asymptotic rate, to machine accuracy, is reached. Naturally, we would expect fast conver-

1

2

3

4

5

T

to the following linear system :

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 4 & 0 & 0 & 1 & 0 & 0 \\ 0 & 3 & 0 & 1 & 0 & 0 & 1 & 0 & 2 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 5 & 0 & 0 & 2 \\ 1 & 0 & 0 & 2 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 3 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 3 & 0 & 0 & 0 & 2 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 4 & 0 & 0 & 2 & 0 & 1 & 1 \end{pmatrix}_{1} = \begin{pmatrix} 1 \\ 10 \\ 0 \\ 11 \\ 24 \\ 3 \\ 8 \\ 2 \\ 9 \\ 2 \end{pmatrix}$$

:

T

$-1 < \rho\,(I - GA) < 1$. If any of the eigenvalues have modulus greater than one then, of course, the iteration process does not converge, and if we choose one of the eigenvalues to be equal to one then it converges to an incorrect solution. This is because $\lambda_i = 1$, for some i, implies that $(I - \Lambda)$ has a zero row and column and therefore one of the solutions x_i remains unchanged at each iteration.

Case 1a) shows that if all the eigenvalues are close to zero then the initial and asymptotic rates of convergence are high and as such the iteration process terminates very quickly.

In cases b)–d) three of the eigenvalues are kept constant with the other increasing in magnitude resulting in a slight decrease in the convergence rates and a small increase in the total number of iterations. Cases e)–h) have the eigenvalue with the largest modulus constant while the other three are varied. These variations do not affect the convergence rates or the number of iterations.

Cases j)–o) test whether the distribution of the eigenvalues affects convergence. In each case the smallest and largest eigenvalues are kept constant with the other two spaced out in between, non-uniformly. This results in a small decrease in the initial rate, but the asymptotic rate and the number of iterations are unchanged.

Cases p)–t) show that if all the eigenvalues are close to one, then the initial and asymptotic rates are lower, but where at least half the eigenvalues are closer to zero than one, the asymptotic rate is attained after two iterations like all previous cases.

Repeated eigenvalues [i) and o)], do not slow convergence themselves, it is only their magnitude that is the determinant factor.

In almost all cases, we attain the asymptotic rate in two iterations, so we do have

rapid early convergence, with the actual rate being greater for a set of eigenvalues with lower magnitude.

Case 2

Now we are going to see how the condition number of $\nu(X)$, of X, affects convergence. Since we are able to find a G to assign any eigenvectors to the matrix $(I - GA)$, we can choose X to be any nonsingular 4×4 matrix. By keeping Λ to be constant we can see what effect X, i.e. $\nu(X)$, has.

Let $\Lambda = diag(-0.5, -0.1, 0.3, 0.7)$. Choose nonsingular matrices X which have different condition numbers, calculate G from quation (2.9), and apply the iteration process.

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1.1 \end{pmatrix} \qquad \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix} \qquad \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}$$

a) $\nu(X) = 1$ b) $\nu(X) = 1.1$ c) $\nu(X) = 2.0$ d) $\nu(X) = 4$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1.5 & 0 & 1.5 \end{pmatrix} \qquad \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix} \qquad \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 2 & 1 & 1 \end{pmatrix} \qquad \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 2 & 0 \\ 2 & 1 & 1 & 2 \end{pmatrix}$$

e) $\nu(X) = 5$ f) $\nu(X) = 8$ g) $\nu(X) = 10.67$ h) $\nu(X) = 12$

2 0 0 0 1 $\frac{1}{2}$ $\frac{1}{3}$ $\frac{1}{4}$ 1 0 0 0

0 1 0 0 $\frac{1}{2}$ $\frac{1}{3}$ $\frac{1}{4}$ $\frac{1}{5}$ 2 1 0 0

2 0 1 0 $\frac{1}{3}$ $\frac{1}{4}$ $\frac{1}{5}$ $\frac{1}{6}$ 1 2 1 0

1 0 3 2 $\frac{1}{4}$ $\frac{1}{5}$ $\frac{1}{6}$ $\frac{1}{7}$ 2 1 2 1

i) ( ) = 19 50 j 2 N - c 7 e J 6 5 - 3 c e F p g , P 3 6 J j F a , P K 5 N 0 J 5 F

T g

Choice of X	*ini ial ra e of convergence*	*asy p o ic ra e of convergence*	*No. of i er. for asy p. ra e*	*o al no. of i era ions*
a)$\nu(X) = 1.00$	1.04982	0.35667	2	17
b)$\nu(X) = 1.10$	1.04982	0.35667	2	17
c)$\nu(X) = 2.00$	1.04982	0.35667	2	17
d)$\nu(X) = 4.00$	1.04982	0.35667	2	17
e)$\nu(X) = 5.00$	1.04982	0.35667	7	20
f)$\nu(X) = 8.00$	1.04982	0.35667	8	20
g)$\nu(X) = 10.67$	0.75463	0.35667	14	20
h)$\nu(X) = 12.00$	0.62549	* * *	*	19
i)$\nu(X) = 19.50$	0.50931	* * *	*	19
j)$\nu(X) = 50.92$	0.16252	* * *	*	20
k)$\nu(X) = 72.00$	0.00157	* * *	*	20

rate. Previously, we began with high convergence rates which decreased until reaching the asymptotic rate and then assumed that constant value. When the condition number of X is significantly high, the convergence rates continue to fall even below the asymptotic rate without achieving a constant, uniform value. The * notation represents a number significantly lower than the asymptotic value. The iteration does converge, but is not as fast.

If we now assign eigenvalues of lower magnitude and follow the method above, with the same matrices X, then we get a similar result. Let $\Lambda = diag(-0.10, -0.05, 0.05, 0.1)$.

Table 7

Choice of X	*initial rate of convergence*	*asymptotic rate of convergence*	*No. of iter. for asymp. rate*	*total no. of iterations*
$a)\nu(X) = 1.00$	2.99573	2.30259	2	4
$b)\nu(x) = 1.10$	2.99573	2.30259	2	4
$c)\nu(X) = 2.00$	2.99573	2.30259	2	4
$d)\nu(X) = 4.00$	2.72747	***	*	5
$e)\nu(X) = 5.00$	2.72747	***	*	5
$f)\nu(X) = 8.00$	2.72747	***	*	5
$g)\nu(X) = 10.67$	2.07944	***	*	5
$h)\nu(X) = 12.00$	2.07944	***	*	5
$i)\nu(X) = 19.50$	1.89712	***	*	5
$j)\nu(X) = 50.92$	1.81708	***	*	5
$k)\nu(X) = 72.00$	1.79263	***	*	5

Again we find that for low condition numbers convergence is unaffected, but if

the condition number of X is high, then even a set of eigenvalues close to zero do not produce rapid convergence, the * represents a figure significantly lower than the asymptotic rate.

In Case 1, we concluded that the majority of the eigenvalues needed to be close to zero for rapid early covergence. Case 2 demonstrates that, while the above condition may be necessary, it is not sufficient. Therefore, we require both eigenvalues of low magnitude and X, the matrix of the corresponding eigenvectors, to be **well conditioned**, too.

So in Chapter 3, when we apply control techniques to assign eigenvalues and some or all of the eigenvectors, we must aim to assign eigenvalues of as low magnitude as possible which yield a matrix X that is reasonably well conditioned.

In this chapter we discuss , i.e. a way of assigning eigenvalues and eigenvectors by state feedback in the linear time invariant system. This procedure is generally used in Control Theory to stabilise an unstable system. We are going to apply this procedure to try to ensure rapid convergence of the iteration process. We shall do this by using the pole assignment technique to control the eigenstructure, i.e. the eigenvalues and the eigenvectors, of the iteration matrix so that it is convergent.

The state feedback pole assignment problem in control system design is essentially an inverse eigenvalue problem. A desirable property of any system design is that the poles should be insensitive to perturbations in the coefficient matrices ofP2jJ2F0GJ63Fba

We now consider the time invariant, linear, multivariable system :

$$\dot{} = \quad + \qquad (3\ 1)$$

where , are and dimensional vectors, respectively, and , are real, constant matrices. Matrix is assumed to have full rank. The behaviour of system (3.1) is determined by its , i.e. the eigenvalues of . In order to modify the poles and make it stable, we choose a state-feedback control, , given by :

$$= \quad + \qquad (3\ 2)$$

where the matrix , the or , is chosen such that the modified dynamic system :

$$\dot{}$$

1 2 n

j

Theorem :

A solution F to Problem 1 exists for **every** set Δ of self conjugate complex numbers if and only if the pair (A, B) is **completely controllable,** that is , if and only if:

$$\mathbf{s}^T A = \mu \mathbf{s}^T \quad and \quad \mathbf{s}^T B = \mathbf{0} \quad \Longleftrightarrow \quad \mathbf{s}^T = \mathbf{0}.$$

If the pair (A, B) is not controllable, i.e. there exists $\mathbf{s}^T \neq \mathbf{0}$ such that $\mathbf{s}^T A = \mu \mathbf{s}^T$ and $\mathbf{s}^T B = \mathbf{0}$, then

$$\mathbf{s}^T (A + BF) = \mu \mathbf{s}^T \quad for \quad all \quad F.$$

Then μ is an eigenvalue of $A + BF$ for all F and must belong to any set Δ of poles to be assigned. The pole μ is said to be **uncontrollable**, and it cannot be modified by any feedback control.

In the single-input case ($m = 1$), the solution to Problem 1, when it exists, is unique. For proof see Mayne and Murdock (1970). If $1 < m < n$, then various solutions exist. Finally, if $m = n$, then a solution always exists, since $rank(B) = n$ implies that the left null space of B contains only the trivial solution and the pair (A, B) is always **completely controllable**; therefore any feedback matrix can always be achieved by feedback, i.e. in this case we can assign the matrix $(A+BF)$ to have any desired eigenstructure.

The main aim in Control Theory is to develop methods for finding F, so that the system is i.e. insensitive to pertubations. Let $_j$ and $_j$, $j = 1, 2, ..., n$, be the right and left eigenvectors of the closed loop system matrix $M \equiv A + BF$,

corresponding to eigenvalue λ_j :

$$M\mathbf{x}_j = \lambda_j, \quad \mathbf{y}_j^T M = \lambda_j \mathbf{y}_j^T \tag{3.4}$$

If M is *non-defective*, then M is diagonalizable and it can be shown [Wilkinson (1965)] that the sensitivity of the eigenvalue λ_j to pertubations in A,B and F depends upon the magnitude of the **condition number** c_j, where

$$c_j = 1/s_j = \frac{||\mathbf{y}_j||_2 \, ||\mathbf{x}_j||_2}{\left|\mathbf{y}_j^T \mathbf{x}_j\right|} \geq 1 \tag{3.5}$$

Wilkinson (1965) gives a bound on the sensitivities of the eigenvalues,

$$\max_j c_j \leq \nu(X) \equiv ||X||_2 \left|\left|X^{-1}\right|\right|_2 \tag{3.6}$$

where $\nu(X)$ is the **condition number** of X. Note that the condition numbers take a minimum value $c_j = 1$, for all $j = 1, 2, ..., n$, if and only if M is a normal matrix, that is $M^*M = MM^*$. In this case the eigenvectors of M may be chosen to be an orthonormal basis, and therefore X is perfectly conditioned with $\nu(X) = 1$.

Problem 1 can now be re-written to allow for **robustness** :

Problem 1′

Given (A, B) and Δ (as in Problem 1), find a real matrix F and non-singular matrix X satisfying

$$(A + BF)X = X\Lambda \tag{3.7}$$

where $\Lambda = diag(\lambda_1, \lambda_2, ..., \lambda_n)$ such that some measure ν of the conditioning, or robustness, of the eigenproblem is optimized.

Note that, in the case where $m = 1$, if F exists then X is uniquely determined

(up to scaling), and therefore the condition numbers, $_j$, cannot be controlled. When = , we can choose to be orthogonal, (e.g.), and hence $_j = 1$. The more interesting cases in Control Theory, which attract more attention, are for the general multi-input system where 1 . In these cases, it is possible to control the sensitivities of assigned poles to a restricted extent by an appropriate choice of eigenvectors.

1 2 n

1^T

0 1

0 1

-1 T -1
0

-1

and pre-multiplication by ^{T} then gives the two equations

$$= \ {}^T_0(\quad \Lambda \quad ^{-1} \qquad)$$

$$0 = \ {}^T_1(\quad \Lambda \quad ^{-1} \qquad)$$

from which (3.8) and (3.10) follow directly, since is invertible from the condition that is nonsingular.

T

1 2 n

0 1

j j j $^T_1$ j

j j

$\mathbf{x_j} \in S_j$ and

$$X_j = [\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_{j-1}}, \mathbf{x_{j+1}}, .., \mathbf{x_n}]$$

At each sweep of the iteration, the vector $\mathbf{x_j}$ is replaced by a new vector $\mathbf{x_j} \in S_j$, selected to improve the conditioning of X. A complete sweep has been made when j has run from 1 to n. The iteration is then continued with the new matrix X until it becomes well conditioned in some sense (i.e. the condition number of X becomes unchanged for some tolerance).

Step 3:

Find the matrix $M = A + BF$ by solving $MX = X\Lambda$ and compute F explicitly from $F = Z^{-1}U_0^T(M - A)$.

The matrix $M = X\Lambda X^{-1}$ is constructed by solving the following equation $X^T M^T = (X\Lambda)^T$ for M^T using direct LU decomposition or Gaussian limination methods.

A program has been written in MATLAB to carry out the above procedures and assign required eingenvalues to the matrix $(A + BF)$.

3.3 Applications

Before we begin to apply the pole assignment procedure to our problem, there are two important issues which need attention.

The first thing to note is that the numerical algorithm, described above, is used to determine the eigenvalues of the matrix $(A + BF)$, for given matrices A and B. But our problem requires us to assign eigenvalues to the iteration matrix $(I - GA)$, for given matrices I and A. This problem is in fact, in the form of :

Problem 2

Given real matrices (A, C) of orders $(n \times n, m \times n)$ respectively, and a set of n complex numbers , Δ=$(\lambda_1, \lambda_2, ..., \lambda_n)$, closed under complex conjugation, find a real $n \times m$ matrix G such that the eigenvalues of $A - GC$ are $\lambda_j, j = 1, 2, ...n$.

Problem 2 is in fact the dual of Problem 1. Whereas Problem 1 was a controller problem, this is its dual known as an observer problem. This makes very few changes to the numerical algorithm, presented above. quation (3.7) now becomes :

$$(A - GC)X = X\Lambda \tag{3.14}$$

where G is to be found. A detailed algorithm for eigenvalue assignment to observer systems can be found in the thesis by S. Stringer(1993). But for our problem we can solve the dual problem and obtain the required matrix G. We need to substitute :

$$A = A^T \qquad B = C^T \qquad and \qquad F = -G^T$$

into the original (cotroller) system and solve it using the numerical algorithm. We can then obtain G from it. So to assign eigenvalues to $(I - GA)$, we put $A = I^T = I$, $B = A^T$ and calculate F. Hence :

$$G = -F^T$$

The other note of importance is that due to the formulation of our iteration process, the matrices A and B are both $n \times n$ matrices. This means that $m = n$ and therefore, as discussed in Section (3.1) we can assign any eigenvalues to $(A + BF)$

and choose X such that $\nu(X) = 1$. Because of this, this case does not attract much attention and therefore it is difficult to find test cases for it.

Example 1: [Cavin and Bhattacharyya(1983)]

$$n = 1, \quad m = 2, \quad \Lambda = (-1, -2)$$

$$A = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \qquad B = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

This gives

$$X = \begin{bmatrix} -0.635 & -0.591 \\ -0.317 & -0.394 \end{bmatrix}$$

and

$$F = [2.00 - 6.00]$$

These results obtained by pole assignment agree with Cavin and Bhattacharyya, which they obtained using an application of Sylvester's equation, since $m = 1$ and therefore F is unique. $\nu(X) = 15.98$ and cannot be controlled.

Example 2: [Barnett(1975)]

$$n = 3, \quad m = 2, \quad \Lambda = (1, 1, 3)$$

$$A = \begin{matrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 6 & -11 & 6 \end{matrix} \qquad B = \begin{matrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{matrix}$$

This gives

$$= \begin{matrix} 1\ 6053 & 3\ 0941 & 1\ 4887 \\ 2\ 0941 & 4\ 2907 & 2\ 1966 \end{matrix}$$

This result is obtained after two iterations and the condition number, $(\quad) = 7\ 81$ is the best that can be achieved since the problem is not completely controllable.

$$\tilde{} = \begin{matrix} 10 & 1 & 2 & 0 \\ 1 & 11 & 1 & 3 \\ 2 & 1 & 10 & 1 \\ 0 & 3 & 1 & 8 \end{matrix} \qquad = \begin{matrix} 6 \\ 25 \\ 11 \\ 15 \end{matrix}$$

We wish to assign $\Lambda = \quad (\ 0\ 1 \quad 0\ 05\ 0\ 05\ 0\ 1)$ to the matrix $(\quad\ \tilde{})$ and then apply the iteration process with the computed $\quad$.

If we substitute $\quad = {}^{T} = \quad$ and $\quad = {}^{\tilde{}T}$ we can find $\quad$ easily from quation (3.10), viz.,

$$= {}^{-1}\ {}^{T}_{0}(\quad \Lambda\ {}^{-1} \quad)$$

by taking $\quad = \quad$.

Having obtained $\quad$, $\quad$ is found from $\quad = {}^{T}$:

$$= \begin{matrix} 1\ 1557809 & 01 & 1\ 0263692 & 02 & 2\ 2758621 & 02 & 6\ 6937120 & 03 \\ 9\ 7971602 & 03 & 1\ 0762677 & 01 & 4\ 8275862 & 03 & 3\ 975659937120 & 02 \end{matrix} \quad 118$$

This value of G will assign the required eigenvalues to $(I - G\tilde{A})$. If we now use this G in our iteration process :

$$\mathbf{x}^{n+1} = (I - G\tilde{A})\mathbf{x}^n + G\mathbf{b}$$

then the method converges to the exact solution in four iterations. From xamples $1-3$ of Chapter 2 we realised that Jacobi converged in 10 iterations, Gauss-Siedel in 5 and the Preconditioned Conjugate Gradient method in 4 iterations. So for this particular example, we have improved on Jacobi and Gauss-Siedel. Since we can assign any eigenvalues to $(I - G\tilde{A})$ and still achieve $\nu(X) = 1$, we can assign eigenvalues of a smaller magnitude than above and see what the outcome is. If we assign the following eigenvalues :

$$\Lambda = diag(-0.001, -0.0005, 0.0005, 0.001)$$

then we obtain :

$$G = \begin{pmatrix} 1.0507204e-01 & 9.3307221e-03 & -2.0689862e-02 & -6.0852535e-03 \\ 9.3306755e-03 & 1.0250220e-01 & 4.5977241e-03 & -3.7863611e-02 \\ -2.0689552e-02 & 4.5976782e-03 & 1.0574660e-01 & 1.1494195e-02 \\ -6.0851318e-03 & -3.7863043e-02 & 1.1494138e-02 & 1.4063416e-01 \end{pmatrix}$$

and the iteration process with this given G converges after just three iterations. If we now assign $\Lambda = diag(-0.00001, -0.000005, 0.000005, 0.00001)$. Then the computed matrix G enables our iteration to find the exact solution in just two iterations. The values at each iteration are shown below :

Table 8:

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$x_4^{(k)}$
0	0.00000000	0.00000000	0.00000000	0.00000000
1	0.99999997	1.99999999	−1.00000004	0.9999999
2	1.00000000	2.00000000	−1.00000000	1.0000000

It can easily be seen that this value of G ensures **very** rapid convergence. The values after just one iteration are very close to the exact solution itself. So for this linear system, the pole assignment procedure is much more efficient than the standard iteration methods discussed in Chapter 2. We cannot improve on this by proceeding with this idea, and assigning eigenvalues of even smaller magnitude, although the the values after one iteration are very, very close to the exact solution.

We now we consider another example, where A is a sparse, unsymmetric 10×10 matrix.

Example 4:

$$\tilde{A} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 1 & 0 \\ 0 & 1 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 1 \\ -1 & 0 & 1 & 0 & 4 & 0 & 0 & -1 & 0 & 0 \\ 0 & 3 & 0 & 1 & 0 & 0 & -1 & 0 & 2 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 & 5 & 0 & 0 & -2 \\ -1 & 0 & 0 & 2 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 & 1 & -3 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 & 1 & 1 & 0 \\ 0 & 3 & 0 & 0 & 0 & -2 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 4 & 0 & 0 & -2 & 0 & 1 & 1 \end{pmatrix} \qquad \mathbf{b} = \begin{pmatrix} 1 \\ 10 \\ 0 \\ -11 \\ 24 \\ 3 \\ -8 \\ 2 \\ 9 \\ 2 \end{pmatrix}$$

If we choose to assign $\Lambda = diag(-0.1, -0.08, -0.06, -0.04, -0.02, 0.00, 0.02, 0.04, 0.06, 0.08)$, to (A + BF) then the resulting matix G ensures convergence in 4 iterations. If we continue to assign eigenvalues of smaller magnitude then, as in the previous case, we can attain the exact solution in just two iterations.

We have studied three standard iteration methods for solving the linear system $=$, and found, unsurprisingly, that the Preconditioned Conjugate Gradient Method was most efficient.

$$^{n+1} = (\quad) \; ^{n} + \qquad (4\ 1)$$

By analysing the matrix form of the iteration process given by quation (4.1), we showed that convergence was not only dependent on the eigenvalues of the matrix ( ), but also on the corresponding eigenvectors. Therefore an efficient iteration process must combine eigenvalues of low magnitude with eigenvectors

demonstrated to have very rapid convergence, even in the case of a fairly sparse matrix.

The procedure can be used to solve large linear systems, very speedily. The only conditions necessary for the application of this method are that :

1) the matrix A must be of full rank, and

2) the pair (I, A) must be completely controllable.

The pair (I, A) is **completely controllable** if and only if:

$$\mathbf{s}^T I = \mu \mathbf{s}^T \quad and \quad \mathbf{s}^T A = \mathbf{0} \quad \Longleftrightarrow \quad \mathbf{s}^T = \mathbf{0}.$$

If the problem is not completely controllable then we may still assign the eigenvalues but the choice of corresponding eigenvectors will be restricted, and therefore X may not be well-conditioned. Nevertheless we can still apply the numerical algorithm. Step 2 of the algorithm selects the eigenvectors from the required subspace, in order to minimise $\nu(X)$ – see xample 2 in Chapter 3. This procedure may help improve the rate of convergence, depending on the condition number of X.

Bibliography

[1] Barnett S. *Introduction to Mathematical Control Theory.* Oxford University Press, Oxford (1975).

[2] Burden R.L. and Faires J.D. *Numerical Analysis 3rd Edition.* Prindle, Weber and Schmidt, Boston (1985).

[3] Cavin R.K. and Bhattacharyya S.P. *Robust and Well-conditioned Eigenstucture Assignment Via Sylvester's Equation.* Optimal Control Applics. and Methods, Vol.4, 205-212 (1983).

[4] Golub G.H. and Van Loan C.F. *Matrix Computations.* The Johns Hopkins University Press (1983).

[5] Kautsky J. and Nichols N. *Robust Eigenstructure Assignment.* Numerical Analysis Report (1983).

[6] Kautsky J., Nichols N. and Van Dooren P. *Robust Pole Assignment in Linear Feedback.* Int. J. Control, Vol.41, No.5, 1129-1155 (1985).

[7] Ogata K. *State Space Analysis of Control Systems.* Prentice-Hall (1969).

[8] Stringer S.M. *The Use of Robust Observers in the Simulation of as Supply Networks.* Reading University Thesis (1993).

[9] Varga R.S. *Matrix Iterative Analysis.* Prentice-Hall (1962).

[10] Wilkinson J.H. *The Algebraic Eigenvalue Problem.* Oxford University Press, Oxford (1965).

[11] Wonham W.M. *On Pole Assignment in Multi-input Controllable Systems.* I Trans. Auto. Control AC-12, 660-665 (1967).

[12] Young D.M. *Iterative Solution of Large Linear Systems.* Academic Press, New York (1971).