The Ninth International Conference on Computing in Civil and Building Engineering
April 3-5, 2002, Taipei, Taiwan

MATRIX ITERATION FOR LARGE SYMMETRIC EIGENVALUE PROBLEMS

Martin Ruess

Department of Civil Engineering, Technical University of Berlin
martin@ifb.bv.tu-berlin.de

Abstract

Eigenvalue problems are common in engineering tasks. In particular the prediction of structural stability and dynamic behavior leads to large symmetric real matrices with profile structure, for which a set of successive eigenvalues and the corresponding eigenvectors must be determined.

In this paper, a new method of solution for the eigenvalue problem for large real symmetric matrices with profile structure is presented. This method yields the eigenstates in the sequence of the absolute values of their eigenvalues. The profile structure is preserved during iteration, thus reducing the storage requirements and the computational effort. Deflation of the matrix in combination with spectral shifts and repeated preconditioning are used to accelerate the iteration. The method is capable of handling multiple eigenvalues and eigenvalues of equal magnitude but opposite sign. For large matrices, less than one decomposition of the matrix is required for each desired eigenvalue. The determination of the eigenvector corresponding to a given eigenvalue requires one decomposition of the matrix.

Solution strategy

The solution strategy extends the well-known QR-method. The special eigenvalue problem (1) is transformed to the diagonal form (2) by a sequence of similarity transformations. The diagonal coefficients of $\mathbf{D}$ are the eigenvalues of (1). The eigenvectors of equation (2) are the unit vectors $\mathbf{e}_i$. The columns of the transformation matrix $\mathbf{Q}$ therefore contain the eigenvectors of $\mathbf{A}$.

$$\mathbf{A}\,\mathbf{x} = \lambda\,\mathbf{x} \tag{1}$$

$$\mathbf{D}\,\mathbf{e}_i = \lambda_i\,\mathbf{e}_i \tag{2}$$

$$\mathbf{D} := \mathbf{Q}^T\,\mathbf{A}\,\mathbf{Q} \quad \text{Diagonalmatrix with the eigenvalues of } \mathbf{A} \tag{3}$$

$$\mathbf{x}_i := \mathbf{Q}^T\,\mathbf{e}_i \quad \text{Eigenvector of } \mathbf{A} \text{ for the eigenvalue } \lambda_i \tag{4}$$

The transformation matrix $\mathbf{Q}$ is determined stepwise by the decomposition of $\mathbf{A}$ into an orthonormal matrix $\mathbf{Q}_s$ and a right triangular matrix $\mathbf{R}_s$ in each step s. The iterated matrix $\mathbf{A}_{s+1}$ is calculated according to (6).

$$\mathbf{Q}_s\,\mathbf{R}_s := \mathbf{A}_s \quad \text{with} \quad \mathbf{Q}_s^T\,\mathbf{Q}_s = \mathbf{Q}_s\,\mathbf{Q}_s^T = \mathbf{I} \tag{5}$$

$$\mathbf{A}_{s+1} := \mathbf{R}_s\,\mathbf{Q}_s = \mathbf{Q}_s^T\,\mathbf{A}_s\,\mathbf{Q}_s \tag{6}$$

The iteration process with (5) and (6) lets the iterated matrix $\mathbf{A}_{s+1}$ converge to diagonal form [1]. The diagonal matrix $\mathbf{D}$ contains the eigenvalues of $\mathbf{A}$ in the sequence of their magnitudes in descending order :

$$|d_1| \geq |d_2| \geq \ldots \geq |d_{k-1}| \geq |d_k| \geq |d_{k+1}| \geq \ldots \geq |d_n| \tag{7}$$

It is shown in [1] that the convergence of the method depends on the convergence of a left triangular matrix $\mathbf{L}$ with diagonal coefficients 1 which in step s of the iteration has the form :

$\mathbf{L}_s$ =

1	0	0	...
$l_{21}\left(\frac{d_2}{d_1}\right)^s$	1	0	...
$l_{31}\left(\frac{d_3}{d_1}\right)^s$	$l_{32}\left(\frac{d_3}{d_2}\right)^s$	1	...
...	...	...	...

The property $d_m/d_i \leq 1$ $(m > i)$ in (7) lets $\mathbf{L}$ converge to $\mathbf{I}$ in the general case. Eigenvalues of equal magnitude but opposite sign will, however, lead to block matrices on the diagonal, which must be diagonalized by Jacobi rotations. Convergence is fastest in the last row and column, since d_n is the eigenvalue of smallest magnitude.

The left and upper profile is denoted by $up[i]$, the right and lower profile of $\mathbf{A}$ is $lp[i]$. The profile of $\mathbf{A}$ is called convex if $m \geq i$ implies $up[m] \geq up[i]$ and $lp[m] \geq lp[i]$, respectively. In the following it is assumed that the profile of $\mathbf{A}$ is convex.
The iteration procedure retains a convex profile structure of the coefficient matrix. This reduces the storage requirements and the computational effort.

QR-Decomposition and RQ-Recombination

In step s Matrix $\mathbf{A}_s$ is decomposed into the product of an orthonormal matrix $\mathbf{Q}_s$ and a right triangular matrix $\mathbf{R}_s$. The decomposition is performed by reducing matrix $\mathbf{A}_s$ stepwise to triangular form with plane rotation matrices $\mathbf{P}_{ik}$. Row k is used to drive the coefficients in rows $i = k + 1$ to $i = lp[k]$ to zero. The reduction is carried out column by column. Hence the generated zero elements in the predecessor columns are preserved. The process is continued until matrix $\mathbf{A}_s$ is an upper triangular matrix $\mathbf{R}_s$.

The multiplication of $\mathbf{A}$ with the rotation matrix $\mathbf{P}_{ik}^T$, which drives a_{ik} to zero, affects only the coefficients in rows k and i (Fig. 1). The transformations in column k extend from row k to row $lp[k]$. The right profile in row k is changed from $lp[k]$ to $lp[lp[k]]$. Due to the convexity of the matrix profile, the right profile in rows $k + 1, \ldots, lp[k]$ remains unchanged. The transformation of $\mathbf{A}_s$ into a right triangular matrix $\mathbf{R}_s$ is thus :

$$\mathbf{Q}_s = \mathbf{P}_{21} \ \ldots \ \mathbf{P}_{lp[1]1} \ \mathbf{P}_{32} \ \ldots \ \mathbf{P}_{lp[2]2} \ \ldots \tag{8}$$

$$\mathbf{R}_s = \ \ldots \ \mathbf{P}_{lp[2]2}^T \ \ldots \ \mathbf{P}_{32}^T \ \mathbf{P}_{lp[1]1}^T \ \ldots \ \mathbf{P}_{21}^T \ \mathbf{A}_s \tag{9}$$

The inverse multiplication of the decomposition product completes the transformation (6) :

$$\mathbf{A}_s = \mathbf{Q}_s \ \mathbf{R_s} \tag{10}$$

$$\mathbf{Q}_s^T \ \mathbf{A}_s \ \mathbf{Q}_s = \mathbf{R_s} \ \mathbf{Q}_s \tag{11}$$

$$\mathbf{A}_{s+1} = \mathbf{R}_s \ \mathbf{P}_{21} \ \ldots \ \mathbf{P}_{lp[1]1} \ \mathbf{P}_{32} \ \ldots \ \mathbf{P}_{lp[2]2} \ \ldots \tag{12}$$

The recombination for $\mathbf{A}_{s+1}$ starts with the right triangular matrix $\mathbf{R}_s$. Multiplication with $\mathbf{P}_{21}$ destroys only the zero in location $(2, 1)$ of $\mathbf{R}_s$. Generally, multiplication with $\mathbf{P}_{ik}$ destroys the zero in location (i, k). All other zeros are preserved. The sequence of multiplications in (12) destroys the zeros columnwise, starting with column 1. In each column k, the zeros are destroyed starting with row $k + 1$, ending with row $lp[k]$.

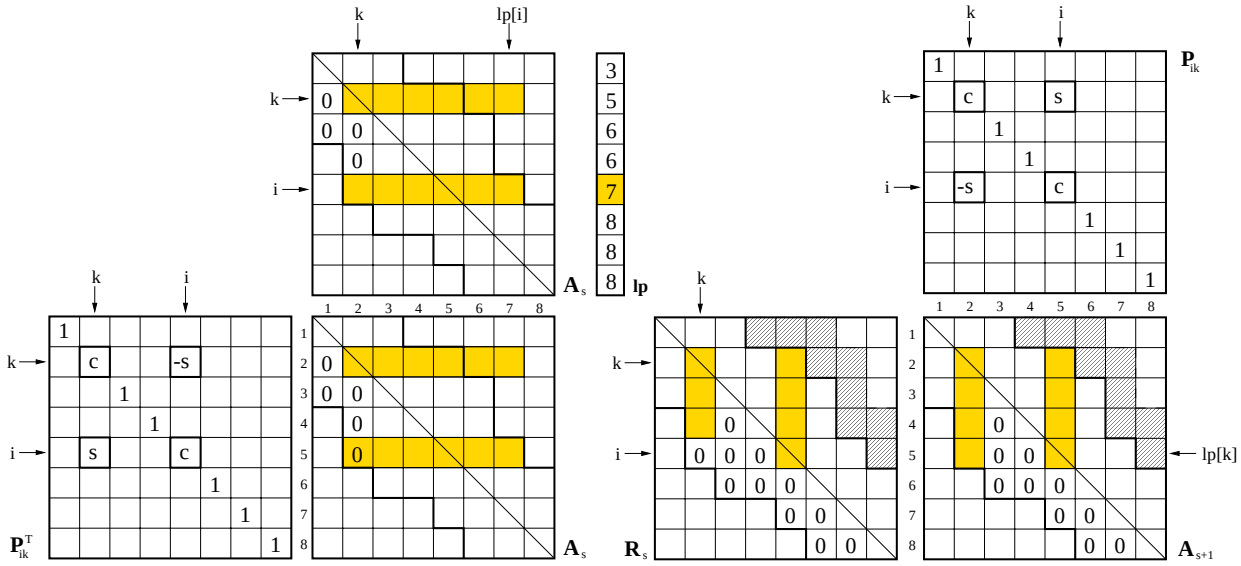


Fig. 1: Decomposition and recombination of $\mathbf{A}_s$

Due to symmetry of matrix $\mathbf{A}_{s+1}$ it is not necessary to compute the elements above the diagonal. The shaded area marks the additional, temporarily used storage per row during decomposition. Due to convexity of $\mathbf{A}$, these values are not required for the recombination of $\mathbf{RQ}$. The symmetry of the matrix $\mathbf{A}_{s+1}$ can be used to prove that $\mathbf{A}_s$ and $\mathbf{A}_{s+1}$ have the same profile. This discovery extends the QR-method to profile matrices. It is used to reduce the computational effort and is essential to the success of the Matrixiteration Method.

Since the matrix $\mathbf{A}$ is symmetric, the transformed matrix $\mathbf{Q}^T\mathbf{AQ}$ is also symmetric. The upper profile of $\mathbf{A}$ is preserved in the transformations (10) to (11), the lower profile is preserved by symmetry. It follows from (12) that the coefficients of $\mathbf{Q}^T\mathbf{AQ}$ on and below the diagonal do not depend on the temporarily stored coefficients of $\mathbf{R}$ (Fig.1). The coefficients above row k need not be computed in column k of $\mathbf{A}_{s+1}$. After all coefficients of $\mathbf{A}_{s+1}$ on and below the diagonal have been computed, the coefficients above the diagonal are determined with the symmetry condition $a_{im} = a_{mi}$.

Deflation, Preconditioning and Complexity

Deflation : Assume that in step s the off-diagonal elements in the last row and the last column of matrix $\mathbf{A}_s$ have converged to zero. Then the last diagonal element a_{nn} is a good approximation for eigenvalue λ_n. Matrix $\mathbf{A}_s$ can be shifted by a_{nn}. Since the last row and column then contain only zero elements, they can be discarded to deflate the matrix. In order to accelerate the convergence of the iteration, the spectrum of $\mathbf{A}$ is shifted before iteration in the last row and column has converged. Let the decomposition of the given matrix be $\mathbf{A} = \mathbf{QDQ}^T$ with $\mathbf{Q}$ and $\mathbf{D}$ defined in (1) and (2). Then a spectral shift c leads to the modified eigenvalue problem (13) with the decompostition (14).

$$(\mathbf{A} - c\,\mathbf{I})\,\mathbf{x} = \mu\,\mathbf{x} \quad \text{with} \quad \lambda = c + \mu \tag{13}$$

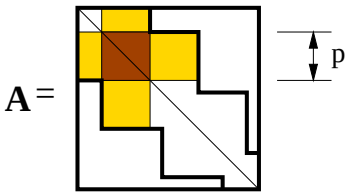
$$(\mathbf{Q}\,\mathbf{D}\,\mathbf{Q}^T - c\,\mathbf{Q}\,\mathbf{Q}^T)\,\mathbf{x} = \mathbf{Q}\,(\mathbf{D} - c\,\mathbf{I})\,\mathbf{Q}^T\,\mathbf{x} = \mu\,\mathbf{x} \tag{14}$$

A shift by c modifies the coefficients in matrix $\mathbf{L}$ and accelerates the iteration in the last rows and columns significantly :

$$l_s \quad = \quad l_{im} \left(\frac{d_i - c}{d_m - c} \right)^s \quad \ll \quad l_{im} \left(\frac{d_i}{d_m} \right)^s \tag{15}$$

It is a good strategy to shift as close as possible to the next eigenvalue. Therefore in the presence of multiple eigenvalues the smaller of the two last diagonal elements is chosen as shift parameter. To avoid overshifting the convergence of the diagonal element must be checked after each QR-step. If necessary, the shift must be modified.

Preconditioning : A preconditioning process is used to enhance diagonal dominance and a descending order of the approximate eigenvalues on the diagonal of $\mathbf{A}$ at an early stage of the iteration. Similarity transformations with orthonormal matrices $\mathbf{U}_k$ are stepwise performed to those parts of size p of $\mathbf{A}$ which have a mutually constant upper and lower profile.



For this purpose matrix $\mathbf{A}$ is subdivided into column ranges in such a way that the upper profile and lower profile is constant within each column range. Each column range leads to a block matrix on the diagonal of $\mathbf{A}$. The rows and columns associated with the block are transformed with an orthonormal matrix $\mathbf{U}_k$ which is selected so that the transformation of the block is a diagonal matrix $\mathbf{D}_k$.

$$\mathbf{A}_k^{p \times p} \quad = \quad \mathbf{U}_k \, \mathbf{D}_k \, \mathbf{U}_k^T \tag{16}$$

Complexity : Let matrix $\mathbf{A}$ have dimension n and an average bandwidth b. The triangular matrix $\mathbf{R}_s$ is computed with 4 multiplications in each of $1.5b$ columns for bn coefficients, thus a total of $6b^2n$ multiplications. The recombination $\hat{\mathbf{A}} = \mathbf{RQ}$ is computed with 4 multiplications in each of $0.5b$ columns for bn coefficients, thus a total of $2b^2n$ multiplications. The computation of q eigenvalues thus requires approximately $8b^2nq$ multiplications.
The complexity of the preconditioning is not significant since the order p of the submatrices is small. The similarity transformation for one block requires $2p^2b$ multiplications. The computational effort for $\mathbf{U}_k$ is of order O(p^3). The total effort for a complete preconditioning of $\mathbf{A}$ is approximately 1% of the effort of a QR-step.

Computation of the eigenvectors

The eigenmatrix $\mathbf{X}$ is the limit of $\mathbf{Q}_s$ in (8). If the eigenmatrix is formed during the iteration the effort is O$[s(bn^2 + nb^2)]$, the storage requirements is O$[n^2]$. The effort is reduced by computing the eigenvectors $\mathbf{x}_i$ corrsponding to a q-fold eigenvalue λ_i with the shifted matrix $\mathbf{C} = \mathbf{A} - \lambda_i \mathbf{I}$ which has a q-fold eigenvalue zero. Reducing $\mathbf{C}$ to an upper triangular matrix $\mathbf{R}$ according to (9) yields the information to build the q eigenvectors $\mathbf{x}_i$ corresponding to the q-fold eigenvalue λ_i.

$$\mathbf{X} \quad = \quad \mathbf{P}_{21} \; \ldots \; \mathbf{P}_{lp[1]1} \; \mathbf{P}_{32} \; \ldots \; \mathbf{P}_{lp[n]n-1} \; \mathbf{I}_q \tag{17}$$

By evaluating the product in (17) from right to left, only the last q columns of $\mathbf{X}$ need to be computed and stored. The computation of the eigenvectors is therefore independent of the number of

computed eigenvalues. The complexity for computing the eigenvectors for a q-fold eigenvalue λ is approximately $6\,b\,n\,q$.

Numerical Examples

The vibration of a thin square plate is characterized by a large number of multiple eigenvalues. Moreover large parts of the total eigenvalue spectrum are clustered. Thus, this problem is regarded to be a severe test case for the new method.

The convergence behavior of the method is demonstrated with a simply supported plate with a 32×32 mesh. This leads to 1089 nodes and 3015 degrees of freedom. The stiffness matrix of the plate has an average bandwidth of 95 and 752 eigenstates of the problem have multiplicity 2.

All 3015 eigenvalues are determined within 2496 cycles of iteration. Thus the average number of cycles per eigenvalue is 0.83 . The following diagram shows the progression of the iteration.

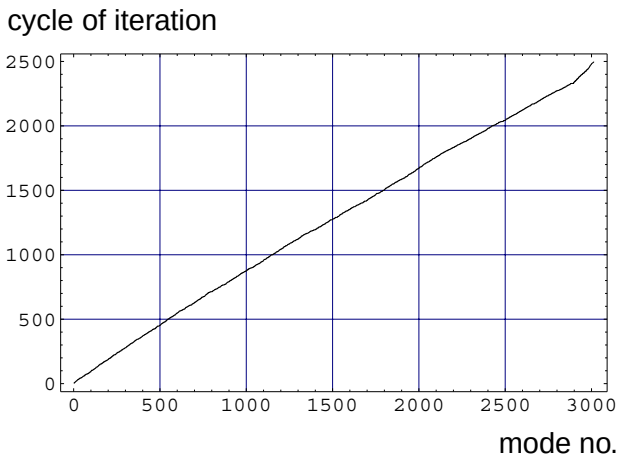


Fig. 2: Convergence of the iteration

The total storage requirement, the estimated numerical effort and the actual effort are shown in Tab.1. To account for deflation the values are estimated with $n = 0.5 \cdot 3015$.

dimension	:	3015
average bandwidth	:	95
storage requirement (double)	:	$576,713$
total no. of reduction steps (estimated)	:	$0.3575 \cdot 10^9$
total no. of reduction steps (measured)	:	$0.3549 \cdot 10^9$
total number of multiplications (estimated)	:	$0.2717 \cdot 10^{12}$
total number of multiplications (measured)	:	$0.2738 \cdot 10^{12}$

Table 1: Computational effort

In a second example only the smallest eigenvalues of a plate vibration are determined. The example is chosen to compare the computational results and the accuracy of the Inverse Matrixiteration with the conventional Lanczos subspace method. The example has 1415 degrees of freedom and a mean bandwidth of 65 elements.

The 10 smallest eigenvalues and error estimates are listed in Tab.2. The residual norm $\|\mathbf{r}\| = \|\mathbf{A}\mathbf{x}_i - \lambda_i \mathbf{x}_i\|$ in Columns 2 and 6 increases quite fast in the Lanczos computation but stays at the same level throughout the computation with Inverse Matrixiteration. Columns 3 and 7 show the difference between the Rayleigh quotient for the computed eigenvector and the computed eigenvalue. In most cases this difference is smaller than $\epsilon_M \|\mathbf{A}\|_2$ with $\epsilon_M =$ machine precision.

Therefore these quantities can be regarded as zero. Column 4 indicates the cycle in which the Inverse Matrixiteration converges to the eigenvalue.

1	*2*	*3*	*4*	*5*	*6*	*7*
$\lambda_{i(InvMIteration)}$	$\|r\|_{IM}$	$\frac{\mathbf{x}_i^T\mathbf{A}\mathbf{x}_i}{\mathbf{x}_i^T\mathbf{x}_i} - \lambda_i$	cycle	$\lambda_{i(Lanczos)}$	$\|r\|_{Lan}$	$\frac{\mathbf{x}_i^T\mathbf{A}\mathbf{x}_i}{\mathbf{x}_i^T\mathbf{x}_i} - \lambda_i$
1.26890544	3.43E-05	-1.41E-10	3	1.26890545	2.54E-10	-7.11E-13
3.16569289	4.03E-05	-3.06E-09	5	3.16569290	3.60E-11	-4.95E-13
3.16569289	2.89E-03	+6.12E-10	5	3.16569290	1.71E-10	+1.31E-13
5.03949639	8.71E-05	-1.28E-09	8	5.03949639	5.91E-06	-1.68E-13
6.28850889	5.95E-03	+3.54E-09	10	6.28850889	6.30E-02	+2.30E-09
6.29511868	6.91E-05	-1.08E-09	10	6.29511869	6.12E-09	-1.68E-13
8.13467966	9.62E-05	+2.57E-09	12	8.13467967	5.36E-05	+2.82E-12
8.13467966	4.79E-04	-4.90E-10	12	8.13467966	2.07E-02	+4.20E-12
10.65210300	8.55E-05	-2.15E-09	14	10.65210306	2.00E-02	+7.22E-09
10.65210299	2.71E-04	+2.09E-08	15	10.65359364	4.16E+01	+1.87E-12

Table 2: The 10 smallest eigenvalues and their errors determined with Inverse Matrixiteration(1-4) and with a restarted Lanczos implementation using exact shifts(5-7)

The Lanczos iteration was restarted 17 times. The implementation uses full reorthogonalization. To detect the last double eigenvalue $(\lambda_{10} = 10.65359364)$ it was necessary to increase the subspace to dimension 20. With smaller subspaces the iteration procedure was not able to determine this eigenstate within the first 100 restarts. Tests with a larger number of multiple eigenvalues failed in acceptable computation time. The influence of the largest eigenvalues could not be dampened satisfactorily. The sequence of the computed eigenvalues was not ordered. The accuracy of the eigenvalues decreased rapidly after the $11th$ eigenvalue.

Conclusions

Inverse Matrixiteration is a suitable method for the computation of any number of eigenstates of large profile matrices. The eigenvalues are determined in order of ascending magnitude. The combination of preconditioning, shifting and deflation leads to good convergence in the presence of multiple eigenvalues and of eigenvalues with equal magnitude but opposite sign.
The computational effort as well as the storage effort are reduced by the preservation of the convex profile. The method permits the solution of the eigenvalue problem for large profile matrices with acceptable numerical effort.

References

[1] P.J. Pahl, M. Ruess. Eigenstates of Profiled Matrices, Invited Lecture at SEMC Conference, Cape Town, 2001

[2] M. Ruess, P.J. Pahl. Die Bestimmung von Eigenzuständen mit dem Verfahren der Inversen Matrizeniteration, Vortrag IKM, Weimar, 2000

[3] J.H. Wilkinson. The Algebraic Eigenvalue Problem, Clarendon Press, Oxford, 1965