MAX-PLANCK-INSTITUT FÜR PLASMAPHYSIK

GARCHING BEI MÜNCHEN

COMPUTING COMPLEX EIGENVALUES OF LARGE NON-HERMITIAN MATRICES

W. KERNER, K. LERBINGER and J. STEUERWALD

IPP 6/236 December 1984

Die nachstehende Arbeit wurde im Rahmen des Vertrages zwischen dem Max-Planck-Institut für Plasmaphysik und der Europäischen Atomgemeinschaft über die Zusammenarbeit auf dem Gebiete der Plasmaphysik durchgeführt.

IPP 6/236 December 1984

COMPUTING COMPLEX EIGENVALUES OF LARGE NON-HERMITIAN MATRICES

W. KERNER, K. LERBINGER and J. STEUERWALD

Abstract

The generalized eigenvalue problem $\mathcal{A}\mathbf{x} = \lambda\mathcal{B}\mathbf{x}$ with a non-symmetric matrix $\mathcal{A}$ is solved by means of inverse vector iteration. The algorithm makes use of the band structure of the matrices thus allowing quite large dimensions ($d \leq 3742$). In the application all complex eigenvalues for the resistive Alfven modes are successively computed.

I. Introduction

Many problems in physics and engineering study the oscillations and the stability of a given system, thus requiring the evaluation of eigenvalues. In general, the physical model includes dissipation and overstable or damped normal modes evolve, which are described by complex eigenvalues.

Let us consider the generalized eigenvalue problem $\mathcal{A}\mathbf{x} = \lambda\mathcal{B}\mathbf{x}$, which is typical of a finite-element discretization, where $\mathcal{A}$ is a general, non-Hermitian matrix and $\mathcal{B}$ a Hermitian, positive definite matrix. The eigenvalue λ and the eigenvector $\mathbf{x}$ are complex. More specifically, the problem is posed to compute selected eigenvalues and eigenvectors when the dimension d of the matrices $\mathcal{A}$ and $\mathcal{B}$ is large, i.e. $d > 1000$. For systems of stiff equations such large dimensions become necessary even for one-dimensional problems. A discretization in terms of finite differences or finite elements usually leads to large but sparse matrices, often with band structure.

While for the treatment of the self-adjoint eigenvalue problem accurate and very efficient solvers exist - we refer to the excellent books of Wilkinson /1,2/ and Parlett /3/ and, in the special case of inverse vector iteration, to Refs. /4,5/ - the situation is much more difficult for non-Hermitian matrices. Let us assume that the chosen discretization does not lead to pathologic Jordan-type matrices. The QR algorithm can then be applied to compute all the eigenvalues of the system /1,2,6/. However, this algorithm destroys the band structure of the matrices and produces full matrices. The storage and CPU time requirements eventually put a limit on the dimension d of the matrices, e.g. on the CRAY-1 of IPP Computer Center d has to be less than 600. However, much finer grids are necessary in the resistive MHD model for finding the stability limits and scaling properties /7,8/.

The inverse vector iteration is a very efficient method to compute selected eigenvalues and eigenvectors of general matrices. It preserves the band structure and thus allows the treatment of very large matrices. The slow convergence, which is sometimes considered a severe drawback, is strongly improved by a suitable complex shift. Fast convergence is found by restarting the iteration with a new shift. In conjunction with a continuation procedure in a relevant parameter the result is found by performing only a few shifts. In the case of Hermitian matrices Sylvester's theorem (see, for example, Ref. /3/) yields the number of eigenvalues in a given real interval and every desired eigenvalue can be found by the bisection method. A generalization of this theorem for general matrices does not exist and therefore the inverse vector iteration cannot be used as a black box to compute all the eigenvalues in a given complex domain. But this also holds for the subspace iteration /9/ or for the Lanczos algorithm /10,11/. In practice, we did not find this drawback very restrictive; the results from a coarse mesh by means of the QR algorithm or from a fine mesh by continuation provides a good guess for a suitable shift. Inspection of details of the eigenfunctions, such as the number of radial oscillations, even makes it possible to compute all eigenvalues in a certain domain of the complex λ-plane successively.

In this paper the algorithm for inverse vector iteration is described and results are presented for the spectrum of the quite complicated resistive MHD operator. Large matrices with a dimension up to 3742 are treated and the extension to even larger matrices is discussed. Not many large-scale applications dealing with the complex eigenvalue problem have been presented in the literature so far. We, therefore, encourage its tackling by using inverse vector iteration because of its accuracy and simplicity. The algorithm and its implementation are described in Section II. The CPU time and storage requirements are discussed in Section III. The results are presented in Section IV, and Section V contains the discussion and conclusion.

II. Algorithm

The general eigenvalue problem

$$\mathcal{A}\mathbf{x} = \lambda \mathcal{B}\mathbf{x} \tag{1}$$

is to be solved, where $\mathcal{A}$ and $\mathcal{B}$ are complex matrices; $\mathcal{B}$ is Hermitian and positive definite and $\mathcal{A}$ is arbitrary. The eigenvalue λ and the eigenvector $\mathbf{x}$ are in general complex. In the algorithm presented the band structure of $\mathcal{A}$ and $\mathcal{B}$, which usually occurs in a finite difference or finite element discretization, is preserved and utilized.

The initial value λ_o is considered as an approximation to the eigenvalue of the system (1), i.e.

$$\lambda = \lambda_o + \Delta\lambda_o \qquad (|\Delta\lambda_o| << |\lambda_o|). \tag{2}$$

With the shift λ_o the eigenvalue equation reads

$$(\mathcal{A} - \lambda_o \mathcal{B})\mathbf{x} = \Delta\lambda_o \mathcal{B}\mathbf{x}. \tag{3}$$

With $\mathbf{x_o}$ as initial guess new vectors $\mathbf{x_i}$ are computed as solutions of the iterative system

$$(\mathcal{A} - \lambda_o \mathcal{B})\mathbf{x}_i = \Delta\lambda_{i-1} \mathcal{B}\mathbf{x}_{i-1} \qquad i = 1, 2, 3, \ldots , \tag{4}$$

and the iterated eigenvalues $\Delta\lambda_i$ by means of the Rayleigh-quotient

$$\Delta\lambda_i = \frac{\mathbf{x}_i^*(\mathcal{A} - \lambda_o \mathcal{B})\mathbf{x}_i}{\mathbf{x}_i^* \mathcal{B}\mathbf{x}_i} \qquad i = 1, 2, 3 \ldots \ . \tag{5}$$

Here the symbol * denotes the Hermitian conjugate. The vector $\mathbf{x}_o$ is usually initialized by random numbers. Special choices prompted by analytical solutions or previous numerical results are admissible and can speed up the convergence. However, this is only done for convenience.

An important condition for the convergence of the system is given by the following result:

Let $\lambda_1, \lambda_2, ..., \lambda_d \in C$ be the eigenvalues of $\mathcal{A}\mathbf{x} = \lambda\mathbf{x}$, $\lambda \in C$ an approximation to one λ_j, $1 \leq j \leq d$, and $\mathbf{t}_o, \mathbf{t}_1, ... \in C^d$ a sequence of vectors, $\mathbf{t}_o$ with random initialization. The inverse vector iteration is then defined by

$$(\mathcal{A} - \lambda I)\mathbf{t}_i = \mathbf{t}_{i-1} \qquad i = 1, 2, 3... \tag{6}$$

If λ is chosen sufficiently close to an eigenvalue

$$|\lambda_j - \lambda| << |\lambda_k - \lambda| \tag{7}$$

for all $k \neq j$, then the sequence $\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3, ...$ converges to the eigenvector corresponding to λ_j. For the proof we refer to textbooks, e.g. Ref./12/. With an appropriately chosen shift λ_o any eigenvalue of the system together with its eigenvector can be computed. Since the matrix $\mathcal{B}$ is positive definite the inverse $\mathcal{B}^{-1}$ exists and the general eigenvalue problem (1) can be reduced to its standard form $\tilde{\mathcal{A}}\mathbf{x} = \lambda\mathbf{x}$ with $\tilde{\mathcal{A}} = \mathcal{B}^{-1}\mathcal{A}$. We then apply the above theorem to this system.

Since the matrix $\mathcal{A}$ is in general non-Hermitian, a second sequence is defined by the left-hand eigenvectors, yielding the iteration

$$\mathbf{y}_o = \mathbf{x}_o \quad ,$$

$$(\mathcal{A} - \lambda_o \mathcal{B})\, \mathbf{x}_i = \Delta\lambda_{i-1} \mathcal{B}\mathbf{x}_{i-1}, \qquad i = 1, 2, 3, \ldots \quad , \tag{8a}$$

$$(\mathcal{A} - \lambda_o \mathcal{B})^* \mathbf{y}_i = \overline{\Delta\lambda}_{i-1}\, \mathcal{B}\mathbf{y}_{i-1} \quad , \tag{8b}$$

where the bar denotes the complex conjugate. In the nonsymmetric case an improved eigenvalue is determined from right-hand and left-hand eigenvectors $\mathbf{x}_i$ and $\mathbf{y}_i$ by using the generalized Rayleigh-quotient

$$\Delta\lambda_i = \frac{\mathbf{y}_i^*(\mathcal{A} - \lambda_o \mathcal{B})\mathbf{x}_i}{\mathbf{y}_i^* \mathcal{B}\mathbf{x}_i} \quad . \tag{9}$$

The system of linear equations is solved by factorization. The shifted matrix is written as a product of triangular matrices

$$\mathcal{A}' = \mathcal{A} - \lambda_o \mathcal{B} = \mathcal{L}\mathcal{U} \quad . \tag{10}$$

This $\mathcal{L}\mathcal{U}$ decomposition preserves the band structure and can be performed efficiently. With the definitions

$$\mathbf{r}_{i-1} = \Delta\lambda_{i-1} \mathcal{B}\mathbf{x}_{i-1} = \Delta\lambda_{i-1} \quad \tilde{\mathbf{x}}_{i-1}, \tag{11a}$$

$$\mathbf{s}_{i-1} = \overline{\Delta\lambda}_{i-1} \mathcal{B}\mathbf{y}_{i-1} = \overline{\Delta\lambda}_{i-1} \quad \tilde{\mathbf{y}}_{i-1}, \tag{11b}$$

the following equations are solved for $\mathbf{p}_i$ and $\mathbf{q}_i$:

$$\mathcal{L}\mathbf{p}_i = \mathbf{r}_{i-1}, \tag{12a}$$

$$\mathcal{U}^*\mathbf{q}_i = \mathbf{s}_{i-1}, \tag{12b}$$

and eventually new vectors $\mathbf{x}_i$ and $\mathbf{y}_i$ are evaluated by

$$\mathcal{U}\mathbf{x}_i = \mathbf{p}_i, \tag{13a}$$

$$L^*\mathbf{y}_i = \mathbf{q}_i. \tag{13b}$$

Utilizing the decomposition of $\mathcal{A}'$, eq. (10), it holds that

$$\mathbf{y}_i^* (\mathcal{A} - \lambda_o \mathcal{B})\mathbf{x}_i = \mathbf{y}_i^* \mathcal{L}\mathcal{U}\mathbf{x} = \mathbf{q}_i^* \mathbf{p}_i \quad ,$$

and the Rayleigh quotient assumes the form

$$\Delta\lambda_i = \frac{\mathbf{q}_i^*\mathbf{p}_i}{\mathbf{y}_i^*\mathcal{B}\mathbf{x}_i} = \frac{\mathbf{q}_i^*\mathbf{p}_i}{\mathbf{y}_i^*\tilde{\mathbf{x}}_i} \quad , \tag{14}$$

where $\tilde{\mathbf{x}}_i = \mathcal{B}\mathbf{x}_i$, defined in eq. (11a), is needed to compute the new vector $\mathbf{r}_i = \Delta\lambda_i \quad \tilde{\mathbf{x}}_i$. The iteration is terminated if the error is smaller than a defined tolerance ϵ, which has to be less than the machine accuracy,

$$|(\lambda_i - \lambda_{i-1})/\lambda_o| < \epsilon \tag{15}$$

for $\lambda_o \neq 0$. Convergence is then obtained in the step i = m. Termination is also forced if the eigenvalue is not monotonically approximated, in order to avoid pathologic iteration paths, i.e. if

$$|\Delta\lambda_i - \Delta\lambda_{i-1}| > |\Delta\lambda_{i-1} - \Delta\lambda_{i-2}| \tag{16}$$

for $i > 3$. The final eigenvalue is given by

$$\lambda = \lambda_o + \Delta\lambda_m \tag{17}$$

and the eigenvector by

$$\mathbf{x} = \mathbf{x}_m. \tag{18}$$

Since we routinely apply inverse vector iteration to analyze part of the complex spectrum of systems with a matrix size too large for the QR algorithm, we obviously have to define a continuation procedure. Knowledge of a relevant part of the spectrum in one point in parameter space is used to explore new regions. A continued vector iteration is then obtained by increasing the number of mesh points or changing the physical parameter α considered. It is explained in detail in the flow chart in Figure 2 subsequent to the description of the algorithm in Section III. In cases where eigenvalues lie close together good results were obtained with mesh refinement if the number of radial intervals was

successively doubled. The final mesh size n_{max} is given by the available core. A way to increase this maximum mesh size by utilizing external storage is discussed in the next section.

III. Implementation

This implementation of the algorithm makes use of routines from the LINPACK library /13/. If the matrices $\mathcal{A}$ and $\mathcal{B}$ are real, then the eigenvalues occur in complex conjugate pairs. The matrices $\mathcal{A}$ and $\mathcal{B}$ are given as INPUT stored in the usual band-matrix storage mode, so that the zero elements outside the bands do not occur at all. Next, the shifted matrix $\mathcal{A}'$ is computed and factored. Note that $\mathcal{A}'$ is now considered complex. In order to make full use of the fast execution on the CRAY-1 vector computer, the LINPACK routines CGBFA for factorization and CGBSL for successive solution of linear systems are used. The evaluation of left-hand and right-hand vectors is achieved by using the same decomposition. The flow chart of the algorithm, presented in Figure 1, displays the steps in the execution. The iteration is terminated if the error is smaller than the desired tolerance ϵ. If convergence is achieved, the eigenvalue λ is given by $\lambda = \lambda_o + \Delta\lambda_m$ and the eigenvector by $\mathbf{x} = \mathbf{x}_m$. Usually five to ten steps are needed for convergence. The maximum number of steps is chosen as $i_{max} = 20$. It is found that an eigenvalue is obtained more easily by choosing a new shift rather than by performing more iterations.

The minimum amount of storage required includes the matrices $\mathcal{A}$ and $\mathcal{B}$ and the decomposition of $\mathcal{A}' = \mathcal{L}\mathcal{U}$ together with the vectors $\mathbf{x}_i$ and $\mathbf{y}_i$. If the norm of the differences from two successive steps $||\mathbf{x}_i - \mathbf{x}_{i-1}||$ and $||\mathbf{y}_i - \mathbf{y}_{i-1}||$ is not used to monitor the convergence, it is possible to overwrite $\mathbf{x}_i$ and $\mathbf{y}_i$ by $\mathbf{r}_i$ and $\mathbf{s}_i$ defined in eq. (11). In addition, a work-space for the pivoting in the linear system has to be given. These storage requirements can easily be improved by keeping only the minimum data necessary

for the algorithm in the fast memory and by storing data on disk. The storage improved algorithm then works as follows:

1. Compute matrix $\mathcal{B}$ and store it on disk B.

2. Compute matrix $\mathcal{A}$ and perform the shift during computation $\mathcal{A}' = \mathcal{A} - \lambda_o \mathcal{B}$.

3. Factor $\mathcal{A}' = \mathcal{L}\mathcal{U}$ and store $\mathcal{L}$ and $\mathcal{U}$ on disk A.

4. Compute new vectors and keep $\mathbf{x}_i, \mathbf{y}_i, \mathbf{p}_i$ and $\mathbf{q}_i$ in the fast memory.

5. Read in $\mathcal{L}$, $\mathcal{U}$ or $\mathcal{B}$ separately, if needed.

This optimized version is a simple extension of the original one. Only one complex matrix in band-matrix storage mode is required in the fast memory at any step together with additional work-space for the factorization with the dimension of the upper band width. The flow chart of this program is displayed in Figure 1. The continued inverse vector iteration for the case of mesh refinement is explained in Figure 2. After initialization the continuation loop starts, where the inverse vector iteration is applied. The converged eigenvector is then transferred by interpolation to the new mesh.

Next we estimate the CPU time and the storage necessary for the algorithm. The number of operations to factor a band matrix with band width b and dimension d is

$$N_F \approx d \cdot b^2/3 \quad ,$$

and the number of operations to solve the linear system

$$N_L \approx d \cdot b \quad .$$

For m iterations there are then

$$N_I \;=\; N_F \;+\; m\, N_L \;\approx\; d \cdot b(b/3 \;+\; m) \tag{19}$$

operations required. Most of them are spent in the decomposition; this part contributes to

$$p \;=\; \frac{100}{1 \;+\; 3m/b} \quad \% \quad .$$

Example 1

In the case of d = 934, b = 47 and m = 6 there are $N_I \;\approx\; 0.95 \;\cdot\; 10^6$ operations necessary with 72 % spent in the factorization. The storage available at IPP, $7.3 \cdot 10^6$ Bytes, which corresponds to 365000 complex words, allows 270,000 matrix elements together with the necessary vectors. In the example given above two matrices could be kept in the core. Factorization and iteration required 3.2 s CPU time on a CRAY-1 with 4.6 s elapsed time.

In the case of continued vector iteration with mesh refinement the dimension d increases in each step $d_l \;\in\; \{\; d_1, d_2, ...d_L; d_1 < d_2 < .... < d_L \;\}$. The total number of operations is then

$$N_I \;=\; b \cdot \left(\frac{b}{3} \;\; \sum_{l=1}^{L} \;\; d_l \;+\; \sum_{l=1}^{L} \;\; m_l \cdot d_l \right), \tag{20}$$

and the fraction p for the decomposition

$$p = \frac{100}{1 + 3 \sum_{l=1}^{L} m_l \, d_l / (b \sum_{l=1}^{L} d_l)} \quad \% \quad .$$

Example 2

The continued vector iteration is performed for three steps with $d_1 = 934$, $d_2 = 1870$ and $d_3 = 3742$, $m_1 = m_2 = m_3 = 3$ and again b = 47. This gives $N_I = 5.8 \times 10^6$ operations. In this case only one complex matrix can be kept in the fast memory and I/O on disk is necessary. The CPU time is 10.1 s with 18.4 s elapsed time. Now 84 % of the CPU time is spent for the factorization.

It is our experience that two to three iterations are saved if the shift λ_0 is taken from a previous run and parameters are changed continuously. Since the I/O has to be paid for such an improvement is clearly visible in the execution of the algorithm. At present we are working on a version with arbitrary matrix size, where the required storage space is partitioned into tractable pieces which are successively worked up . While the basic algorithm is still the same, this implementation is quite different. It will be reported separately.

IV. Results

The algorithm presented is applied to a problem in plasma physics where the question of stability and heating prompts study of the entire spectrum of normal modes. The compressible, resistive MHD equations are linearized around an equilibrium in cylindrical geometry with an ansatz for the perturbed quantities $\mathbf{x}(r, \theta, z; t) = \mathbf{x}(r) e^{\lambda t + i(m\theta + kz)}$, where m and k are wave numbers and λ is the eigenvalue. The spectrum is evaluated by applying the Galerkin method in conjunction with finite elements, which leads to the

generalized eigenvalue problem eq. (1). The vector $\mathbf{x}$ contains the perturbed velocity, pressure and magnetic field $\mathbf{x}^t(r) = (\mathbf{v}^t(r), p(r), b_r(r), b_z(r))$ and has six components when $\nabla \cdot \mathbf{b} = 0$ is used. The matrix $\mathcal{A}$ contains details of the equilibrium and of the expansion functions, which are chosen as cubic Hermite and quadratic finite elements. It is a real, but non-symmetric matrix of dimension $d = 12(N_r - 1) + 10$, where $N_r + 1$ denotes the number of radial grid points. The matrix $\mathcal{B}$ represents the norm and is symmetric and positive definite; more details are given in Refs. /7,8/. Owing to the quite different length and time scales a fine grid is necessary for accurate representation of the eigenfunctions, i.e. $N_r > 100$ with $d > 1200$. The matrices $\mathcal{A}$ and $\mathcal{B}$ have block tridiagonal structure with blocks formed by 12×12 matrices. The matrices are sparse; outside a band of width $b = b_u = b_l = 23$ (for upper and lower band width) there are only zero elements. We have added zeroes outside the blocks but inside the band width.

We now present details of the results for the resistive Alfven branch of the spectrum. The equilibrium is given by

$$B_z(r) = B_o(1 + c\, r) \quad , c = 2.$$

$$B_\theta(r) = 0 \quad ,$$

$$p(r) = B_o^2 \left(c(1 - r) + \frac{c^2}{2} (1 - r^2) \right) \quad ,$$

$$\rho(r) = \rho_o = ct.$$

The Alfven modes are characterized by incompressibility and are easily distinguished from the fast and slow compressible modes. For finite resistivity all the modes are damped.

The QR algorithm can only be applied up to 51 radial points with a matrix dimension of $d \approx 600$. The eigenvalues for a case with $\eta = 1 \times 10^{-4}$ are displayed in Figure 3. In the case of zero resistivity the Alfven modes form a continuum, also indicated in Figure 3, with singular eigenfunctions. Especially interesting is the question what happens for small resistivity, i.e. in the limit $\eta \rightarrow 0$. The results for smaller resistivity are quite puzzling; Figure 4 shows the spectrum for the same mesh as in Figure 3 but with smaller eta. With decreasing resistivity the location of the eigenvalues in the complex λ plane drastically changes. On the contrary, analytical results suggest that in the limit of vanishing η the eigenvalues lie on prescribed curves. (Refs. /14,15/.) More mesh points can only be introduced by using inverse vector iteration. It has been confirmed in conjunction with careful convergence studies that the damping, i.e. - Re λ, is independent of the resistivity. With up to 313 radial points, this result could be verified for the resistivity as small as $\eta = 10^{-6}$. In Figure 5 the spectrum is displayed for $\eta = 2.0 \times 10^{-5}$, the same value as in Figure 4, but computed by applying inverse vector iteration using 313 mesh points. Comparison of the converged results of Figures 3 and 5 indeed reveals that the eigenvalues lie on identical curves. The eigenfunctions have an increasing number of radial nodes $\nu = 1,2,3,...$ with $\nu = 1$ closest to the ideal continuum. The upper (lower) line of the triangle corresponds to eigenfunctions with oscillations near the boundary (near the origin). The more oscillations, the higher the damping. At the branch point the oscillations occur at the centre and vanish at the end points, and further along the eigenvalue curve these oscillations extend over the entire radius. Diagram 7 shows three eigenfunctions for three different eigenvalues. Purely damped modes emerge from a second branch point on the negative real axis of the λ plane. The eigenfunctions are Bessel function-like with practically constant amplitude but an increasing number of radial modes away from this branch point towards the two accumulation points $\lambda = 0$ and $\lambda = \infty$. If the numerical resolution is not good enough, completely false results are obtained for normal modes with eigenvalues between the two branch points - as those shown in Figure 4. To represent

adequately both the radial oscillations and the amplitude modulation requires a much finer grid than to resolve the purely damped modes. Only with the reported fine grids of $N \simeq 300$ were we able to understand the numerical results near the branch points. By applying inverse vector iteration in conjunction with convergence studies we worked out the details of the entire resistive Alfven spectrum. The smaller the resistivity, the more eigenvalues lie on the curve, but the curve itself becomes independent of eta in the limit $\eta \to 0$. The ideal ($\eta = 0$) Alfven continuum is approximated only at the two end points in the limit of $\eta \to 0$ by modes where the eigenfunction is peaked in a small layer at r = 0 and r = 1.0. This layer width δ decreases with η as $\delta \sim \eta^{1/3}$. Diagram 6 displays the first twelve eigenvalues for $\eta = 2 \times 10^{-6}$. Figures 7a and 7d display the eigenfunctions of two cases with almost the same eigenvalue but for two different values of the resistivity. The structure of the eigenfunction is similar, but for smaller resistivity more radial oscillations occur in a finite radial domain.

Finally, we want to point out that by applying inverse vector iteration in the described continuation procedure the complete Alfven spectrum for even more complicated configurations with three and four branch points was mapped out. In addition, results were obtained for the resistive sound-mode spectrum with a similar pattern but much smaller eigenvalues. Details are reported in Refs. /8, 16/.

V. Discussion

Many applications in physics and engineering culminate in numerical solution of the complex eigenvalue problem $\mathcal{A}\mathbf{x} = \lambda\mathcal{B}\mathbf{x}$. If the dimension d of the matrices is small, i.e. $d < 100$, the QR algorithm, which solves for all eigenvalues and eigenvectors, is suitable.

In many cases the physical model includes dissipation with a small value of the corresponding parameter. The problem then has quite different spatial and temporal

scales which require a fine grid for accurate numerical approximation. The numerical properties and numerical accuracy can only be checked by elaborate convergence studies. In such sophisticated models, analysis of the complete spectrum of eigenvalues is useful for the development of suitable methods and for application. In the resistive MHD model eigenvalues with positive real part correspond to exponentially growing instabilities, which can terminate plasma discharges. The numerical search for stable equilibria is therefore very important. But also the stable part of the spectrum is relevant for plasma heating. Put briefly, there are many reasons for the solution of the complex eigenvalue problem for large systems. The QR then cannot be used any more because of its immense storage and CPU time requirements. For repeated study of a specific part of the spectrum, such as stability, it is not efficient in any case.

For really large matrices we propose using inverse vector iteration. Since it preserves the band structure of the matrices, it is a very efficient method of computing selected eigenvalues and eigenvectors. The algorithm contains factorization and, subsequently, solution of linear systems and is, basically, simple. Most of the computing time is spent in the $\mathcal{LU}$ decomposition of a band matrix. With a suitable shift λ_0 fast convergence is found. Its fast execution allows - in an interactive manner - efficient analysis of part of, or even of the entire, spectrum. There is no guarantee against one or several eigenvalues being missed somewhere in the complex plane. But this is a general problem of the non-Hermitian case. All efficient solvers, such as subspace iteration or Lanzcos algorithm, suffer from this drawback, which has to be lived with. On the other hand, if one eigenvalue of the branch of the spectrum of interest is found by means of random shifts or by a continuation procedure as done by the authors, one can indeed find the next one and eventually the entire branch. Careful inspection of the corresponding eigenfunctions, e.g. in terms of radial oscillations, allows one to decide whether the nearest eigenvalue has been found. The efficiency of the method makes up for the fact that one has to restart several times with slightly different

shifts. In the application of the algorithm, we do not just show with a few examples that we can indeed compute complex eigenvalues and eigenvectors; with the discussion of the entire Alfven spectrum we also want to point out the difficulties involved in a complex spectrum. Near the branch points of the Alfven spectrum a very fine grid is necessary. Only careful convergence studies performed by inverse vector iteration allow correct, i.e. converged, results to be separated from false ones. If the resistivity is chosen too small for a given grid, totally wrong eigenvalues appear as shown in Fig. 2. This means that with, for example, $\eta = 2 \times 10^{-5}$ the starting values from the QR algorithm are not of much use. Nevertheless, we are able to work out the details of the entire spectrum for values of the resistivity as small as $\eta = 10^{-5}$. It then becomes evident that the eigenvalues lie on specific curves which are independent of η in the limit of small resistivity. Part of the Alfven spectrum, such as the purely damped modes or the neighbouring ones to the ideal continuum, are easily computed for η as small as $\eta = 10^{-6}$. In conjunction with mesh accumulation we are able to analyse instabilities for η values of $\eta \simeq 10^{-10}$. In summary, it is emphasized that the complete spectrum can be found by inverse vector iteration with, admittedly, quite a lot- but very fast - computer runs.

Let us come back to the algorithm itself. Its successful application encourages us to extend the inverse vector iteration to such large systems that one matrix cannot be kept in memory, even in band-matrix storage mode. Such large systems naturally occur in the normal mode analysis of two-dimensional toroidal equilibria. In this case the algorithm has to be split up into tractable pieces. For the $\mathcal{LU}$ decomposition this is possible and thus such an extension is straightforward.

Another useful extension is the computation of an entire branch. Starting from one eigenvalue it should then be possible by suitable orthogonalization in the iteration to find the closest one and hence, eventually, the entire curve. This procedure generates a special subspace. In our opinion such a version, desirable as it is, will not replace the

algorithm based on inverse vector iteration, that is presented here.

References

/1 / J.H. Wilkinson, "The Algebraic Eigenvalue Problem", Clarendon Press, Oxford 1965

/2 / J.H. Wilkinson and C. Reinsch, "Linear Algebra", Springer-Verlag Berlin, Heidelberg, New York 1971

/3 / B.N. Parlett, "The Symmetric Eigenvalue Problem", Prentice-Hall, Inc., Englewood Cliffs, N.J. 07632, 1980

/4 / R. Gruber, Comput. Phys. Comm. 20, 421 (1980)

/5 / R. Gruber and D. Scott, Comput. Phys. Comm. 23, 115 (1981)

/6 / B.T. Smith et al., Matrix Eigensystem Routines - EISPACK Guide, 2nd Edition, Springer- Verlag Berlin, Heidelberg, New York 1976

/7 / W. Kerner, K. Lerbinger, R. Gruber and T. Tsunematsu to appear in Comput. Phys. Comm.

/8 / W. Kerner and K. Lerbinger, 1984 International Conference on Plasma Physics, Lausanne - Switzerland, June 27 - July 3, 1984, paper P16-8

/9 / Y. Saad, "Chebyshev acceleration techniques for solving nonsymmetric eigenvalue problem", Yale University, Dep. of Computer Science, Technical Report 255, December, 1982

/10/ B.N. Parlett, Computing Methods in Applied Sciences and Engineering, V, R. Glewinski, J.L. Lions (editors), North-Holland Publishing Company, INRIA, 1982

/11/ B.N. Parlett, D.R. Taylor and Z.A. Liu, to be published

/12/ J. Stoer and R. Bulirsch, "Einführung in die Numerische Mathematik II", Heidelberger Taschenbücher Band 114, Springer-Verlag Berlin, Heidelberg, New York 1973

/13/ J.J. Dongerra, C.B. Moler, J.R. Bunch and G.W. Stewart, "LINPACK USER'S GUIDE" SIAM, Philadelphia 1979

/14/ Y.P. Pao and W. Kerner to appear in Phys. Fluids

/15/ D. Lortz and G. Spies, Tenth International Conference on Plasma Physics and Controlled Nuclear Fusion Research, London UK, September 12 - 19, 1984, paper E-III-11

/16/ W. Kerner and K. Lerbinger, IPP-Report in preparation

Figure Captions:

Figure 1 Flow chart of the algorithm for the inverse vector iteration.

Figure 2 Flow chart of the algorithm for the continued inverse vector iteration.

Figure 3 The resistive Alfven spectrum for $\eta = 10^{-4}$ and $N_r = 40$ intervals computed by the QR algorithm. All eigenvalues are correct i.e. converged. The solid bar on the imaginary axis denotes the ideal Alfven continuum ($0.40 \leq Im(\lambda_{id}) \leq 2.80$).

Figure 4 The resistive Alfven spectrum for $\eta = 2 \times 10^{-5}$ and $N_r = 40$ intervals computed by the QR algorithm. A major part of the eigenvalues is false due to insufficient numerical resolution. The solid bar on the imaginary axis denotes the ideal Alfven continuum ($0.40 \leq Im(\lambda_{id}) \leq 2.80$).

Figure 5 The resistive Alfven spectrum for $\eta = 2 \times 10^{-5}$ and $N_r = 312$ successively computed by inverse vector iteration. All eigenvalues are correct i.e. converged. The solid bar on the imaginary axis denotes the ideal Alfven continuum ($0.40 \leq Im(\lambda_{id}) \leq 2.80$).

Figure 6 The first twelve eigenvalues of the upper branch of the resistive Alfven spectrum for $\eta = 2 \times 10^{-6}$ and $N_r = 312$ intervals successively computed by inverse vector iteration. The solid bar on the imaginary axis denotes the ideal Alfven continuum ($0.40 \leq Im(\lambda_{id}) \leq 2.80$).

Figure 7 Normal component of the velocity $v_1 = rv_r$ computed by inverse vector iteration corresponding to the eigenvalue

a) $\lambda = -0.27 + i \cdot 2.33$ for $\eta = 2 \times 10^{-5}$, which is the fourth mode of the upper branch.

b) $\lambda = -0.63 + i \cdot 1.44$ for $\eta = 2 \times 10^{-5}$ which is the tenth mode of the lower branch.

c) $\lambda = -1.57 + i \cdot 0.38$ for $\eta = 2 \times 10^{-5}$ which is the last mode with oscillation i.e. finite imaginary part.

d) $\lambda = 0.27 + i \cdot 2.32$ for $\eta = 2 \times 10^{-6}$ which is the twelvth mode of the upper branch.

Note that real and imaginary part of the eigenfunctions have similar structure and equal magnitude.

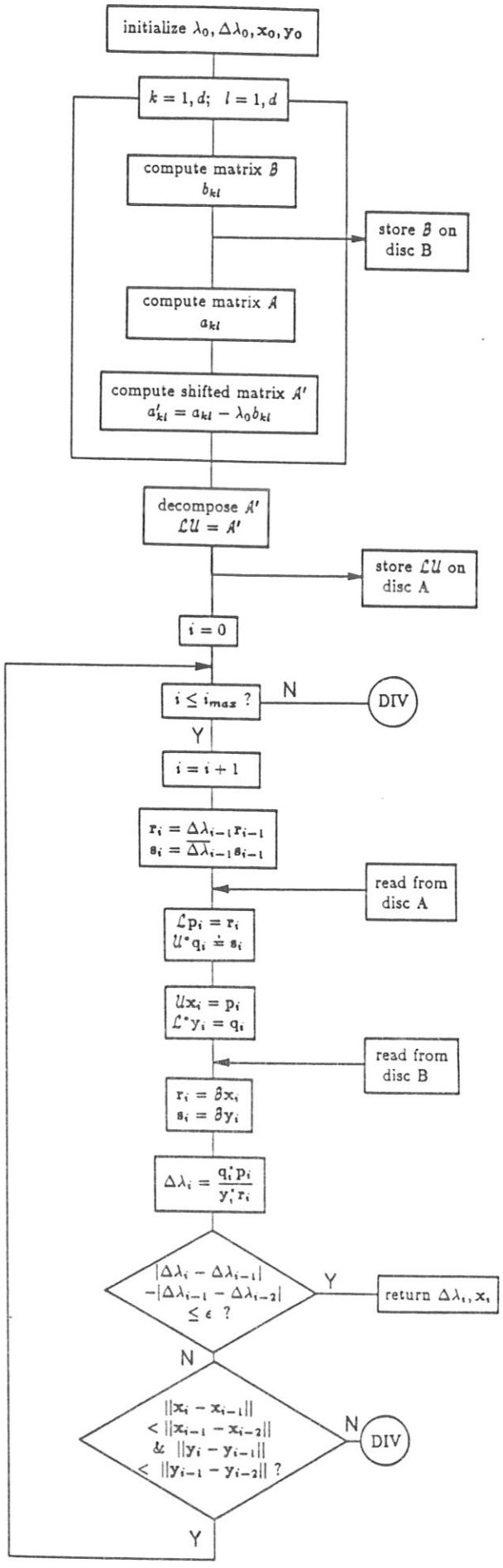
initialize $\lambda_0, \Delta\lambda_0, x_0, y_0$
$k = 1, d;\ \ l = 1, d$
compute matrix B b_{kl}
store B on disc B
compute matrix A a_{kl}
compute shifted matrix A' $a'_{kl} = a_{kl} - \lambda_0 b_{kl}$
decompose A' $LU = A'$
store LU on disc A
$i = 0$
$i \leq i_{max}$?
N
DIV
Y
$i = i + 1$
$r_i = \Delta\lambda_{i-1} r_{i-1}$
$s_i = \overline{\Delta\lambda}_{i-1} s_{i-1}$
read from disc A
$Lp_i = r_i$
$U^* q_i \doteq s_i$
$Ux_i = p_i$
$L^* y_i = q_i$
read from disc B
$r_i = Bx_i$
$s_i = By_i$
$\Delta\lambda_i = \frac{q_i^* p_i}{y_i^* r_i}$
$|\Delta\lambda_i - \Delta\lambda_{i-1}| - |\Delta\lambda_{i-1} - \Delta\lambda_{i-2}| \leq \epsilon$?
Y
return $\Delta\lambda_i, x_i$
N
$\|x_i - x_{i-1}\| < \|x_{i-1} - x_{i-2}\|$ & $\|y_i - y_{i-1}\| \leq \|y_{i-1} - y_{i-2}\|$?
N
DIV
Y

Fig. 1

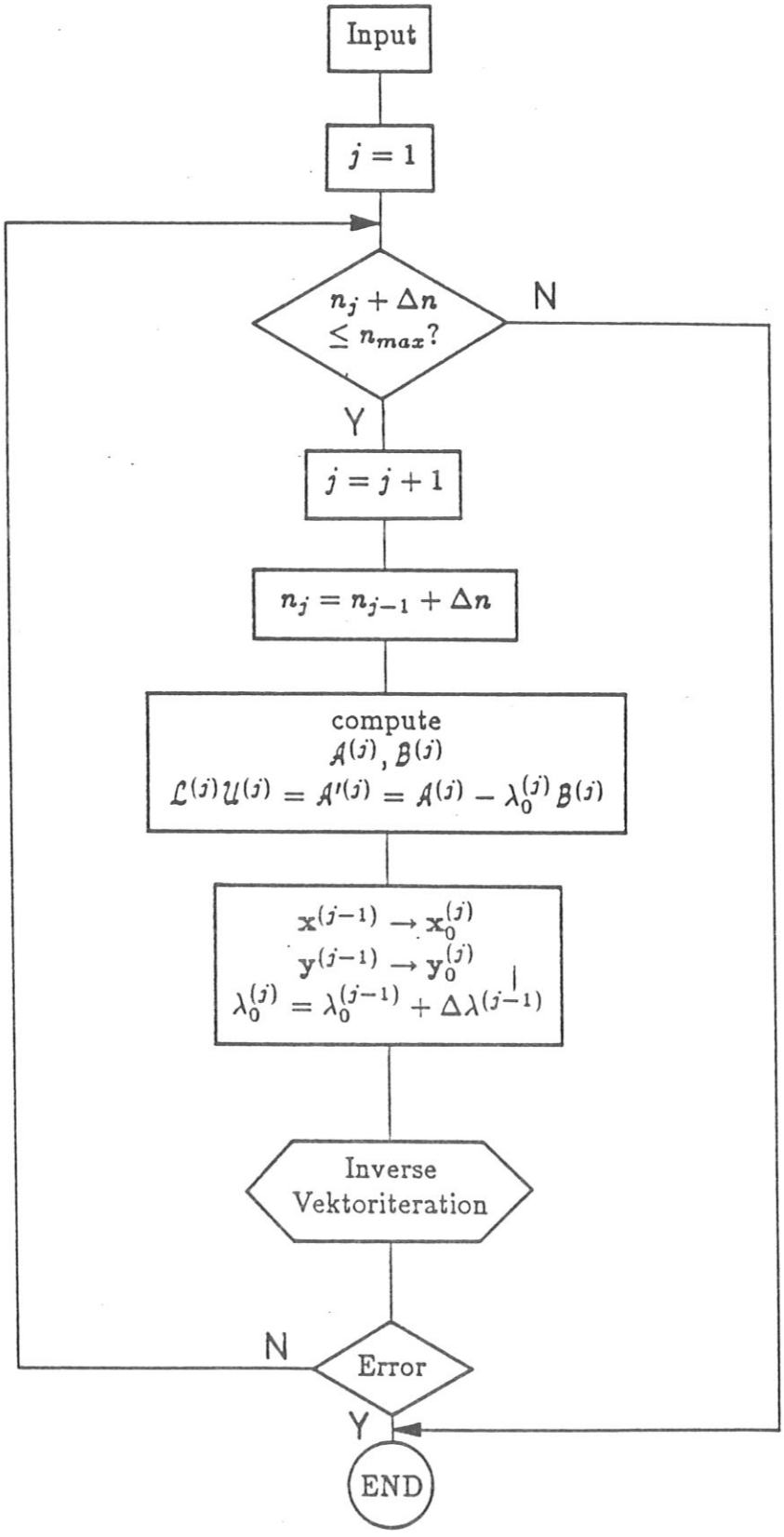
Input
$j = 1$
$n_j + \Delta n \leq n_{max}$?
N
Y
$j = j + 1$
$n_j = n_{j-1} + \Delta n$
compute
$A^{(j)}, B^{(j)}$
$L^{(j)} U^{(j)} = A'^{(j)} = A^{(j)} - \lambda_0^{(j)} B^{(j)}$
$x^{(j-1)} \rightarrow x_0^{(j)}$
$y^{(j-1)} \rightarrow y_0^{(j)}$
$\lambda_0^{(j)} = \lambda_0^{(j-1)} + \Delta\lambda^{(j-1)}$
Inverse
Vektoriteration
N
Error
Y
END

Fig. 2

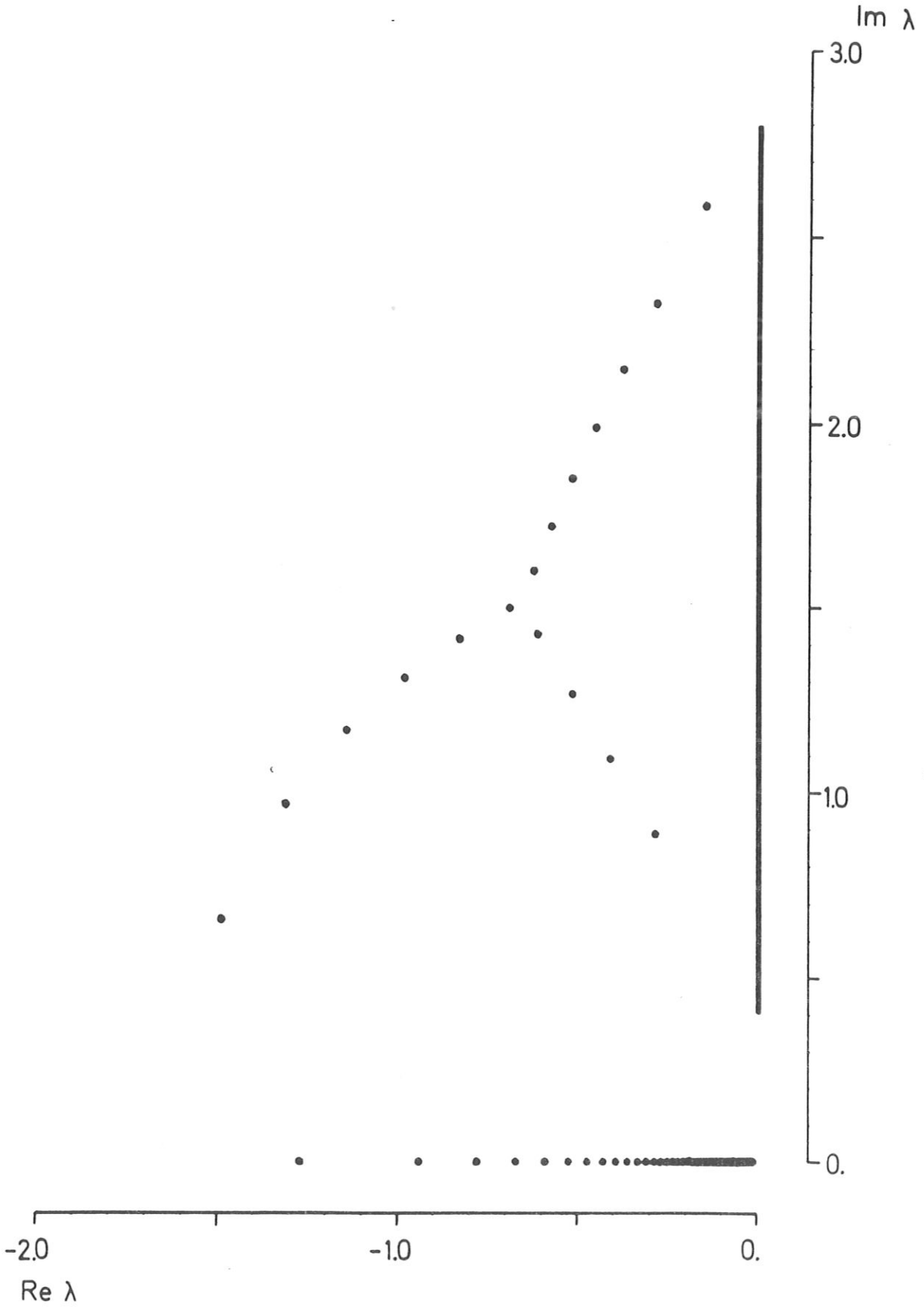

Fig. 3

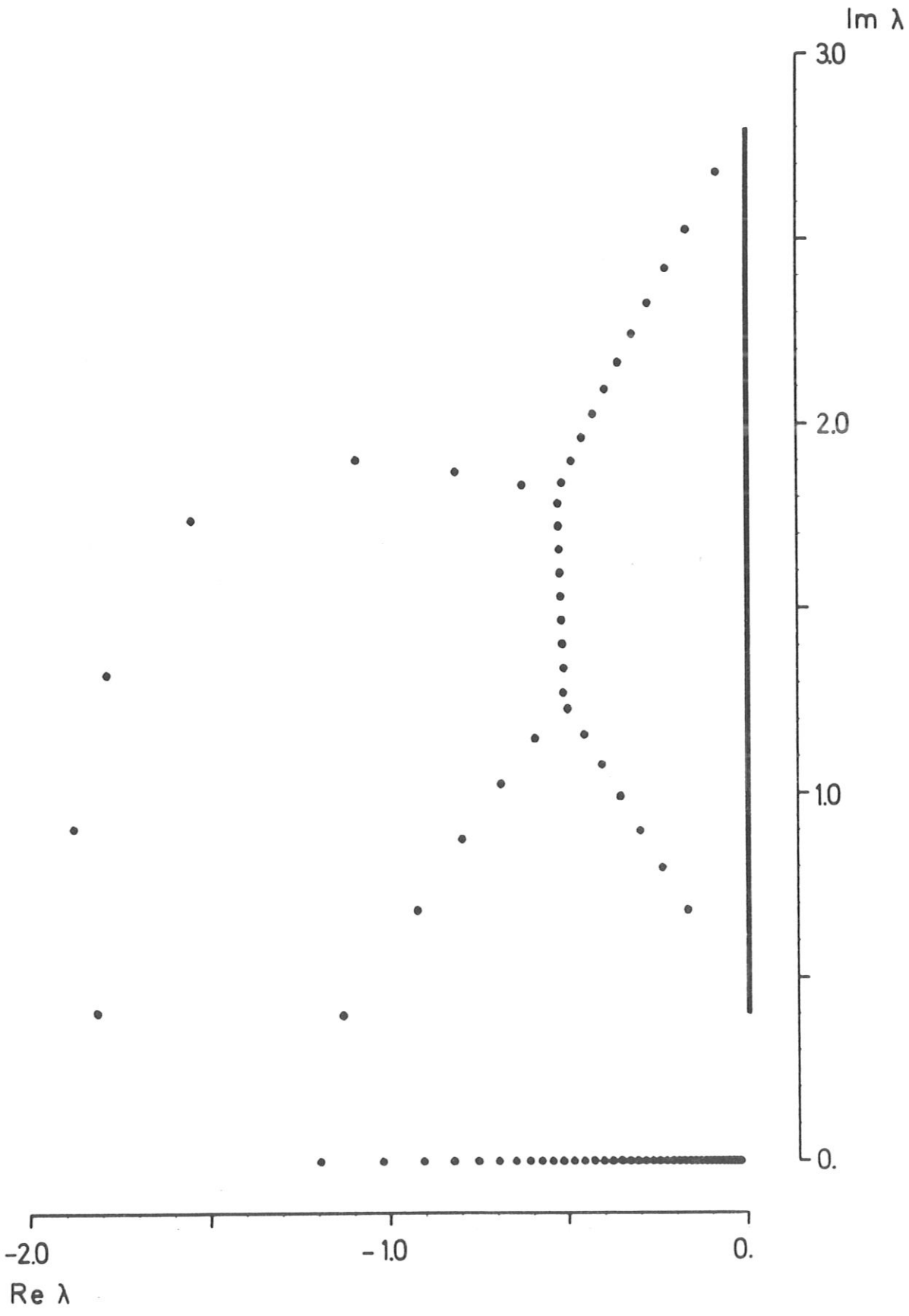

Fig. 4

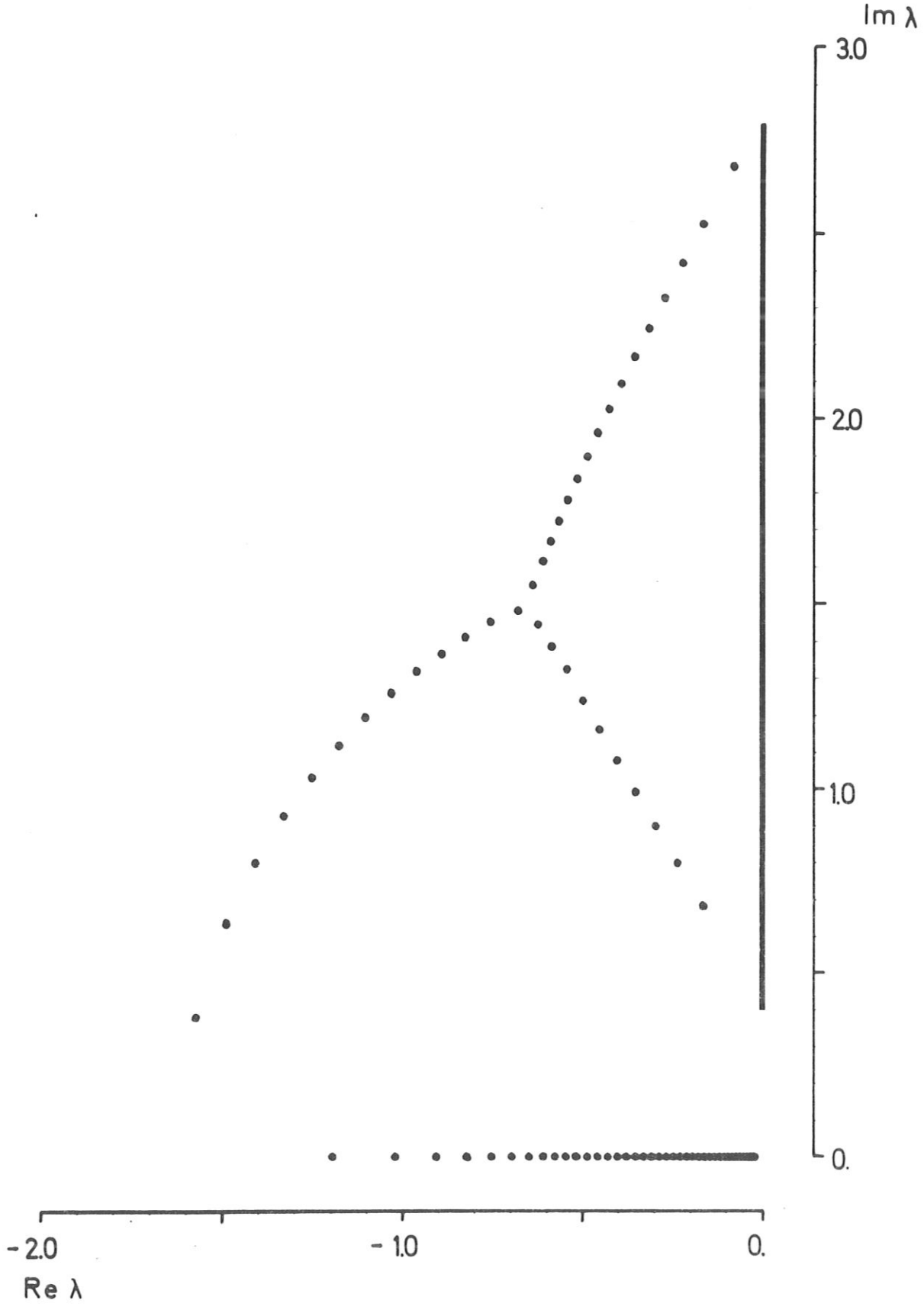

Fig. 5

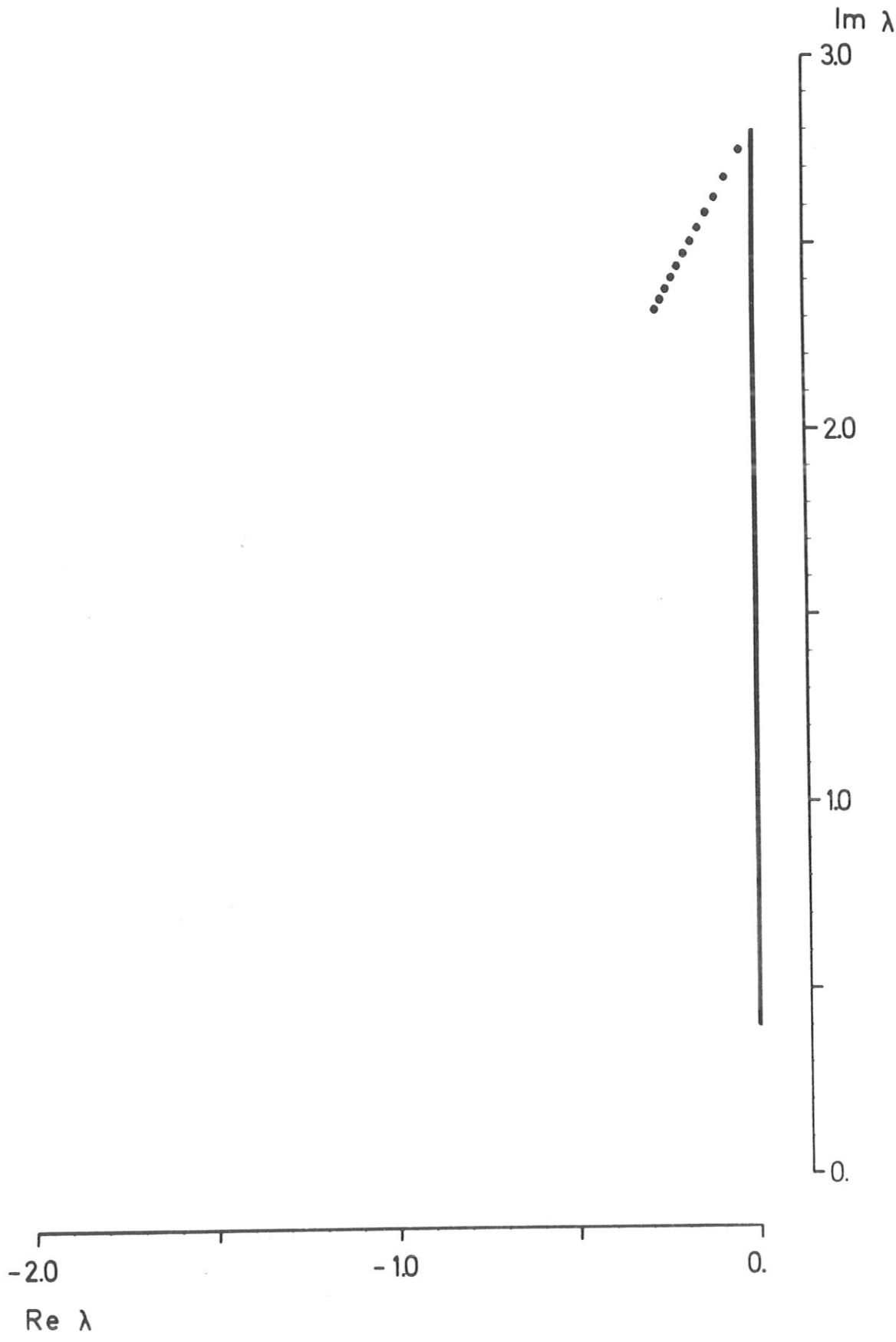

Fig. 6

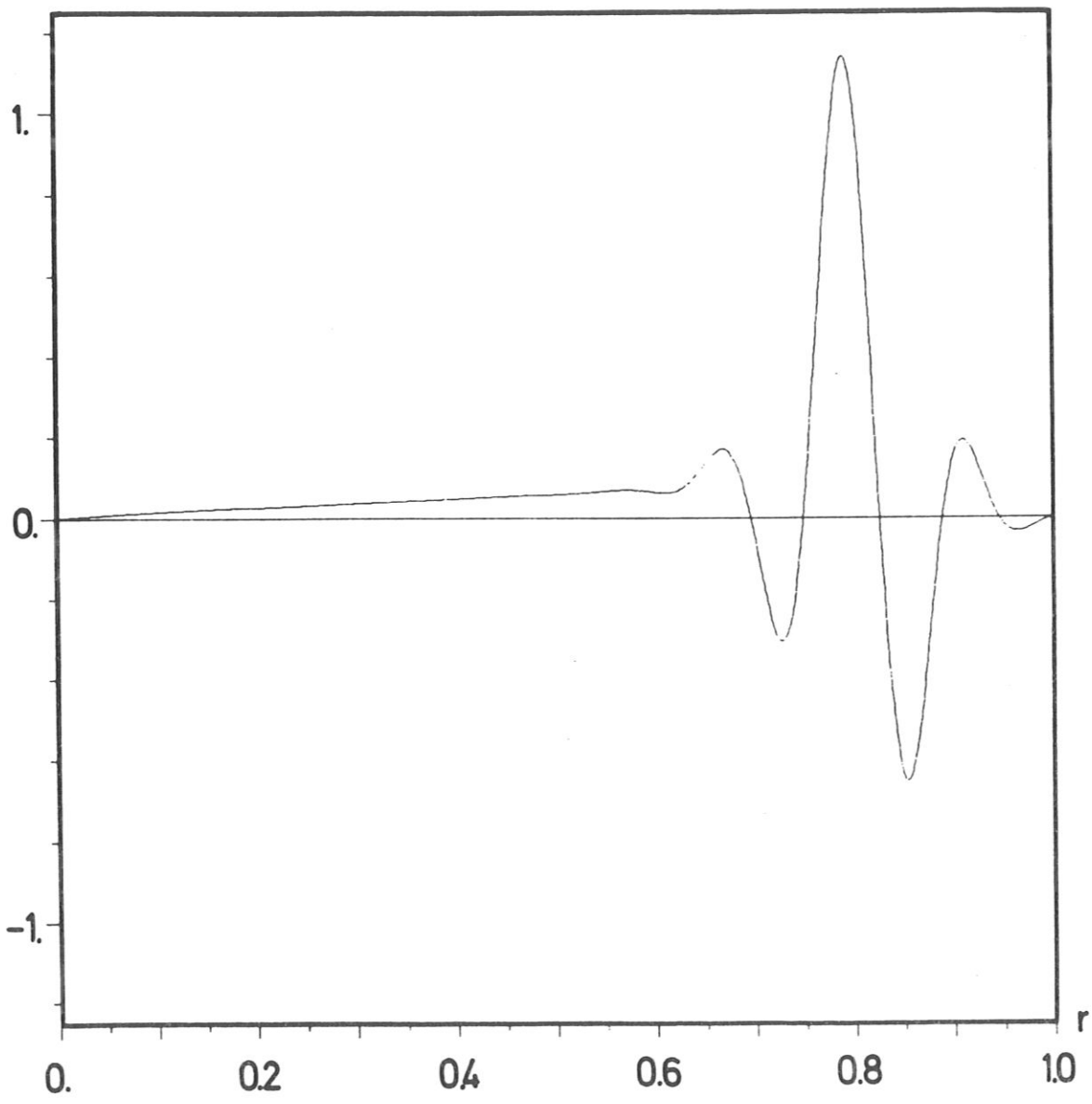

Fig. 7a

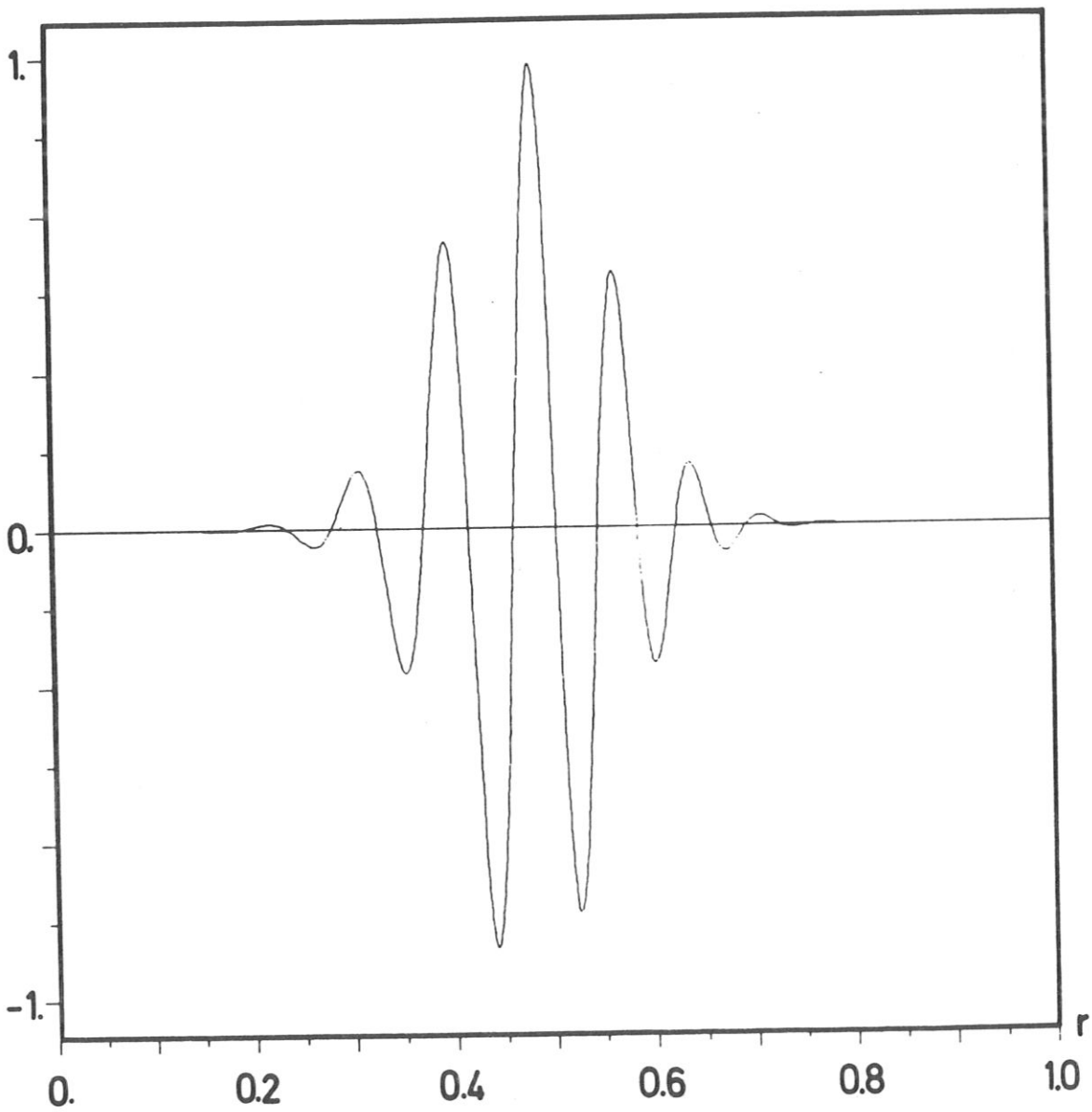

Fig. 7b

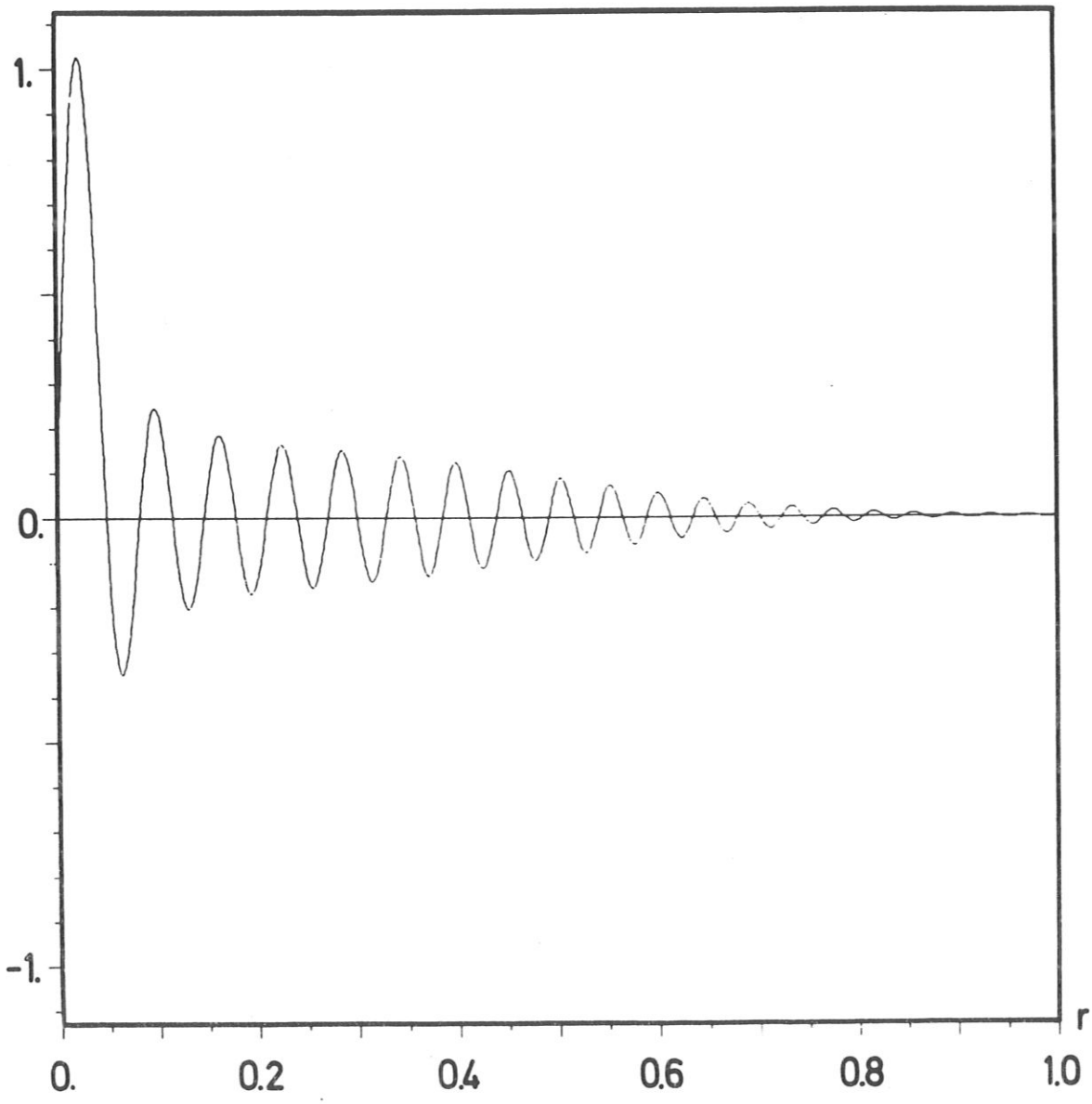

Fig. 7c

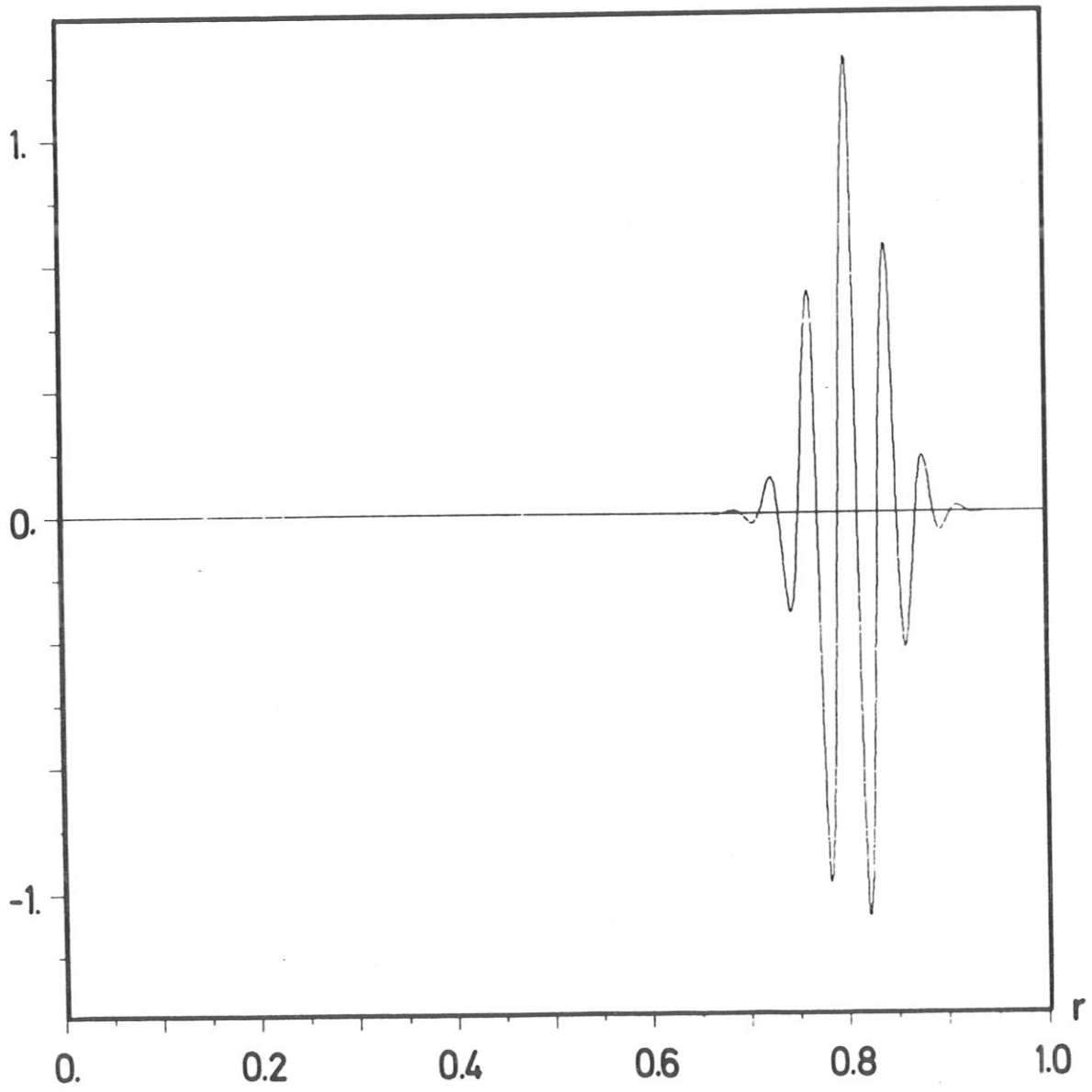

Fig. 7d