HPCN2000, Amsterdam, May 8, 2000

Jacobi-Davidson QZ

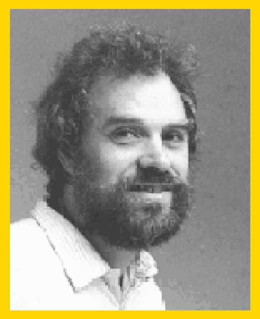
an efficient tool for solving Generalized Eigenproblems

Abstract

Jacobi-Davidson QZ is an efficient and robust algorithm for finding one or a few eigenvalues with specific properties (largest, with largest real part, closest to zero) and for finding the associated eigenvalues.

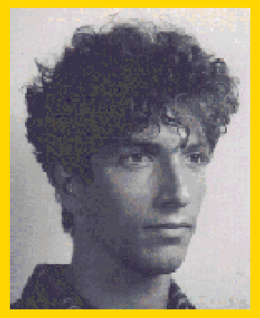
The method applies to all sorts of square matrices: symmetric, non-symmetric, non-normal, complex, etc.. In comparison with other methods it is specifically powerful for large sparse matrices.

There are extensions for solving quadric eigenproblems and for even higher order polynomial eigenproblems.

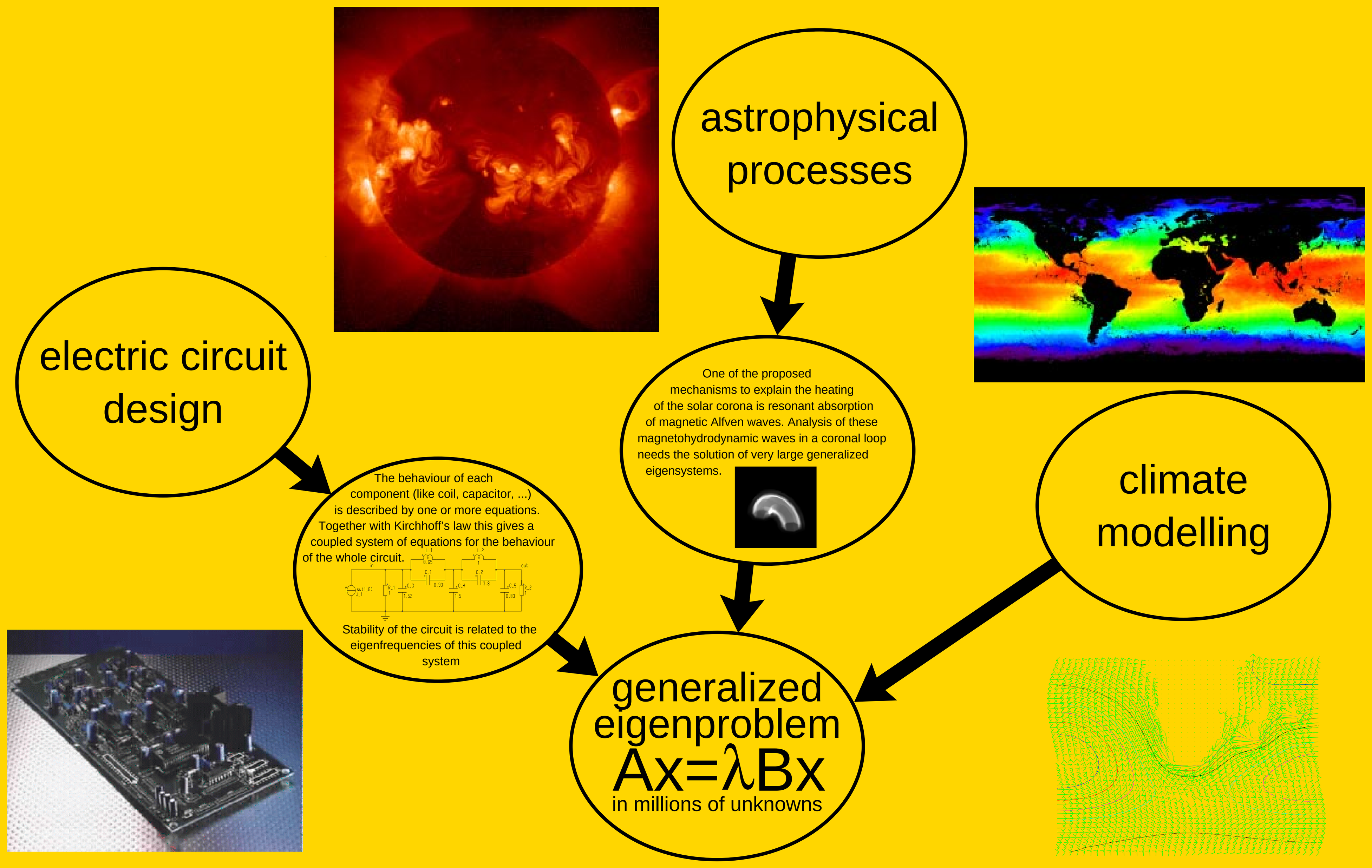


Gerard Sleijpen

Henk van der Vorst



Menno Genseberger



For given square n by n matrices $\mathbf{A}$ and $\mathbf{B}$ and some value $\tau \in \mathbb{C}$, find scalars $\lambda \in \mathbb{C}$ close to the target value τ and vectors $\mathbf{x}$ for which

$$\mathbf{Ax} = \lambda \mathbf{Bx}$$

λ is an eigenvalue, $\mathbf{x}$ is an eigenvector associated with λ.

High resolution discretizations of the modelling partial differential equations lead to generalized eigenproblems of very high dimension. In the project, there was a need for solvers of eigenproblems of dimension $600\,000$ and more. The conventional approaches could not cope well with such high dimensions.

Conventional Approaches

I. The QZ-algorithm

1. The QZ-algorithm produces square orthogonal matrices $\mathbf{Q}$ and $\mathbf{Z}$ for which

$$\mathbf{Z}^*\mathbf{AQ} = \mathbf{S} \quad \text{and} \quad \mathbf{Z}^*\mathbf{BQ} = \mathbf{T}$$

are upper triangular. QZ codes are available in software packages as LAPACK.

2. Then $\mathbf{Ax} = \lambda\mathbf{Bx}$ for $\lambda = \mathbf{S}_{i,i}/\mathbf{T}_{i,i}$ and $\mathbf{x} = \mathbf{Qy}$ where $\mathbf{y} \neq \mathbf{0}$ satisfies the upper triangular system $(\mathbf{S} - \lambda\mathbf{T})\mathbf{y} = \mathbf{0}$.

Only feasible for low dimensional problems ($n \lesssim 1000$)

II. Shift and Invert Arnoldi

1. Shift and invert to a standard eigenvalue problem:

$$\left[(\mathbf{A} - \tau\mathbf{B})^{-1}\mathbf{B}\right] \widetilde{\mathbf{x}} = \widetilde{\lambda}\widetilde{\mathbf{x}}$$

Then $\mathbf{Ax} = \lambda\mathbf{Bx}$ for $\lambda = \tau + 1/\widetilde{\lambda}$ and $\mathbf{x} = \widetilde{\mathbf{x}}$.

2. Arnoldi (or Lanczos) for the shifted and inverted problem can be used to compute the eigenvalue(s) λ closest to τ.

Requires exact solutions of systems $(\mathbf{A} - \tau\mathbf{B})\mathbf{t} = \mathbf{s}$. This is costly or unfeasible for high dimensions

New Approach Jacobi-Davidson

Characteristics

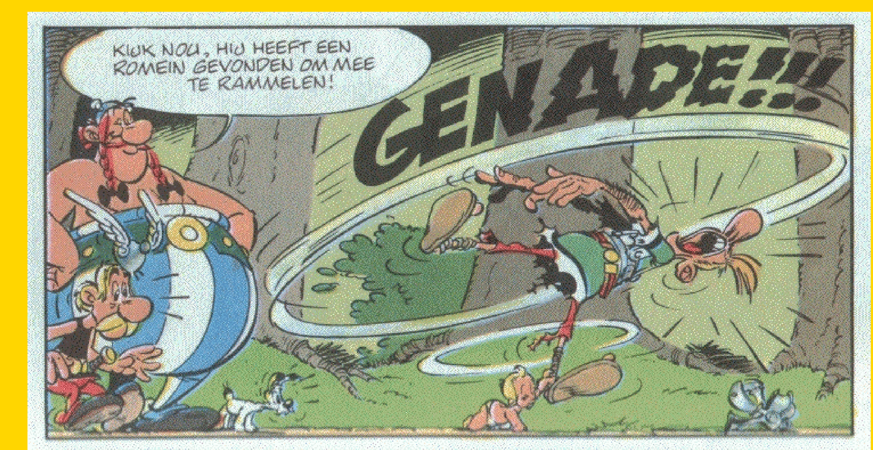
1. Applicable for high dimensional problems.
   The matrices may be complex and non-symmetric.
2. No exact solutions are needed of systems involving $\mathbf{A}$ and $\mathbf{B}$.
3. Iterative method that uses the following high dimensional operations
   - vector updates
   - DOT products
   - multiplications by $\mathbf{A}$ and $\mathbf{B}$ (no inversion or exact solves)
   - application of some preconditioner

   plus the QZ-algorithm on low dimensional problems.

   High dimensional operations are well parallelizable (how well depends on the sparsity structure of the matrices $\mathbf{A}$ and $\mathbf{B}$ and on the selected preconditioner).
4. Fast convergence.
5. Information from simpler models (preconditioners) can be exploited to speed up convergence.
6. Can also find eigenvalues with associated eigenvectors in the interior of the spectrum (= set of eigenvalues).
7. There are extensions (JDQZ) for computing a specific portion of the spectrum.
8. There are extensions for quadratic and higher order polynomial eigenproblems.
9. Convergence to specific eigenvalues can be guided.

A new and robust tool for finding eigenfrequencies

. . . reduced models can be exploited . . .



. . . operations are well parallelizable . . .

cartoons taken from Asterix & Obelix

Details on Jacobi-Davidson

Jacobi-Davidson searches for an approximate eigenvector in a low dimensional **search subspace** $\mathcal{V}_k$. It expands this subspace with one vector if the obtained approximation is not accurate enough. This procedure is repeated until convergence.

The low dimensionality allows efficient computations.

Fundamental questions

- How to **extract** approximate eigenvectors from $\mathcal{V}_k$? *Answer:* **Galerkin**
- How to **expand** $\mathcal{V}_k$? *Answer:* **Newton**

Both answers are optimal with respect to the information that is available. Jacobi-Davidson is (inexact) Newton with subspace acceleration in which all steps are optimally tailored for (generalized) eigenvalue problems.

EXTRACT — Petrov-Galerkin

Find $\mathbf{u} \in \mathcal{V}_k = \{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$ such that

$$\mathbf{r} = (\mathbf{A} - \theta\mathbf{B})\mathbf{u} \perp \mathcal{W}_k = \{\mathbf{w}_1, \ldots, \mathbf{w}_k\}.$$

$(\theta, \mathbf{u})$ is the **Petrov pair** w.r.t. the **search** subspace $\mathcal{V}_k$ and the **test** subsp. $\mathcal{W}_k$.

In practice. $k \ll n$

With both $\mathbf{V}_k \equiv [\,\mathbf{v}_1\,|\,\mathbf{v}_2\,|\,\ldots\,|\,\mathbf{v}_k\,]$ and $\mathbf{W}_k \equiv [\,\mathbf{w}_1\,|\,\mathbf{w}_2\,|\,\ldots\,|\,\mathbf{w}_k\,]$ orthonormal, and $\mathbf{u} = \mathbf{V}_k\,y$, we have a low dimensional generalized eigenproblem

$$(\mathbf{W}_k^*\mathbf{AV}_k - \theta\mathbf{W}_k^*\mathbf{BV}_k)\,y = 0.$$

Apply QZ for the low dimensional reduced matrices $\mathbf{W}_k^*\mathbf{AV}_k$ and $\mathbf{W}_k^*\mathbf{BV}_k$.

EXPAND — Inexact Newton

Compute the **residual** vector $\mathbf{r}$ and an auxiliary vector $\mathbf{z} = \widetilde{\mathbf{z}}/\|\widetilde{\mathbf{z}}\|_2$ where

$$\mathbf{r} \equiv \mathbf{Au} - \theta\mathbf{Bu} \quad \text{and} \quad \widetilde{\mathbf{z}} \equiv \beta\mathbf{Au} + \alpha\mathbf{Bu}.$$

Find an approximate solution $\widetilde{\mathbf{t}} \perp \mathbf{u}$ of the Jacobi-Davidson correction equation

$$\mathbf{u}^*\mathbf{t} = 0 \quad \text{and} \quad (\mathbf{I} - \mathbf{z}\,\mathbf{z}^*)\,(\mathbf{A} - \theta\,\mathbf{B})\,(\mathbf{I} - \mathbf{u}\,\mathbf{u}^*)\,\mathbf{t} = -\mathbf{r}$$

Expand $\mathcal{V}_k$ with $\widetilde{\mathbf{t}}$: orthonormalize $\widetilde{\mathbf{t}}$ w.r.t. $\mathbf{V}_k$ to $\mathbf{v}_{k+1}$.
Expand $\mathcal{W}_k$ with $\widetilde{\mathbf{s}} \equiv \alpha\mathbf{A}\widetilde{\mathbf{t}} + \beta\mathbf{B}\widetilde{\mathbf{t}}$: orthonormalize $\widetilde{\mathbf{s}}$ w.r.t. $\mathbf{W}_k$ to $\mathbf{w}_{k+1}$.

The Jacobi-Davidson corr. eq. is a Jacobian system of a Newton process. Therefore: quadratic convergence when solved exactly; but, due to the subspace acceleration, also fast convergence with inexact solutions.

In practice. Any linear solver can be used to solve the correction equation. Complication: the correction equation is singular in $\mathbb{C}^n$. However: the correction equation is non-singular from $\mathbf{u}^\perp$ to $\mathbf{z}^\perp$. Preconditioned iterative Krylov subspace solvers, as preconditioned GMRES, are very suitable for such situations: all iterates are orthogonal to $\mathbf{u}$ if that is the case for the initial guess. Often already a few steps lead to fast convergence.

$\mathbf{u}$ approximates the singular vector. The projections restrict to a subspace on which the system is well-conditioned.

Preconditioners

If $\mathbf{M}$ is a preconditioner for $\mathbf{A} - \theta\mathbf{B}$, then $\mathbf{M}_p$ is a preconditioner for $\mathbf{A}_p$, where

$$\mathbf{M}_p \equiv (\mathbf{I} - \mathbf{z}\,\mathbf{z}^*)\,\mathbf{M}\,(\mathbf{I} - \mathbf{u}\,\mathbf{u}^*) \quad \text{and} \quad \mathbf{A}_p \equiv (\mathbf{I} - \mathbf{z}\,\mathbf{z}^*)\,(\mathbf{A} - \theta\mathbf{B})\,(\mathbf{I} - \mathbf{u}\,\mathbf{u}^*)\,.$$

The action of the preconditioned operator $\mathbf{M}_p^{-1}\mathbf{A}_p$ on vectors $\mathbf{s}$ can be efficiently incorporated in solvers as GMRES:

First solve $\mathbf{y}$ from $\mathbf{My} = \mathbf{z}$ and compute $\mu \equiv \mathbf{u}^*\mathbf{y}$. Only once per linear solve!
Then, in each GMRES step, compute $\mathbf{t} \equiv \mathbf{M}_p^{-1}\mathbf{A}_p\mathbf{s}$ as

- $\mathbf{s} \leftarrow (\mathbf{A} - \theta\,\mathbf{B})\mathbf{s}$
- Solve $\mathbf{Ms}' = \mathbf{s}$
- $\mathbf{t} \leftarrow \mathbf{s}' - \mathbf{y}(\mu^{-1}(\mathbf{u}^*\mathbf{s}'))$

Only one (skew) projection!

Note that $\mathbf{M}$ can also be used for other nearby θ. It pays to make a good preconditioner.

Jacobi-Davidson QZ

Restart. Increasing storage or computational overhead, for increasing dimension k of the subspaces, may make it necessary to restart (if, say, $k > 30$). With a restart by a single vector, valuable information that is contained in the remaining part of the subspace may be discarded, leading to a slowdown of speed of convergence. Therefore, Jacobi-Davidson is restarted with a space spanned by a small number (say, 5) of Petrov vectors with Petrov values closest to the target τ. These are the vectors that contain the most valuable information for the wanted eigenpair.

Deflation. When a Petrov value is close enough to an eigenvalue, the remaining part of the current subspace will already have rich components in nearby eigenpairs. This information is used as basis for a subspace for the computation of the next eigenvector. In order to avoid that old eigenvectors will re-enter the computational process, the new search vectors are explicitly orthogonalized to the computed eigenvectors. This technique is called *deflation*.

Partial QZ-decomposition. For stability and for easy computation, working with orthonormal basis is preferred. Instead of eigenvectors, a partial QZ-decomposition is computed:

$$\mathbf{AQ}_m = \mathbf{Z}_m S_m \quad \text{and} \quad \mathbf{BQ}_m = \mathbf{Z}_m T_m.$$

Here the columns of the n by m matrix $\mathbf{Q}_m$ form an orthonormal basis of the space spanned by m wanted eigenvectors. The matrix $\mathbf{Z}_m$ is also n by m and orthonormal. S_m and T_m are square m by m upper triangular. As for the full QZ-decomposition, eigenvalues with associated eigenvector can easily be extracted from the partial decomposition.

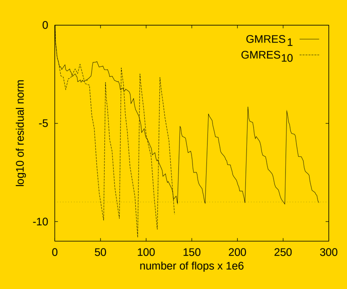
Jacobi-Davidson QZ. The use of the QZ-decomposition for the reduced matrices allows easy accommodation of restarts and deflation and leads in a natural way to a partial QZ-decomposition. The resulting scheme is called **Jacobi-Davidson QZ** and can be viewed as a truncated form of the QZ-algorithm for large generalized eigenvalue problems.

Solving the correction equation accurately: two small examples

The examples here are of relative low order and serve as illustration of the effect of solving of the correction equation more accurately. In both examples the five rightmost eigenvalues are computed. The figures show the $\log_{10}$ *of the norm of the residual as a function of the computational costs (expressed in mega floating point operations). Every time when the norm is less then* 10^{-8}*, an eigenvalue is determined and the iteration is continued with deflation for the next eigenvalue. The maximum dimension of the search subspace is 15. After restart the dimension is 10. One and ten steps of GMRES were used to compute an approximate solution of the correction equation.*

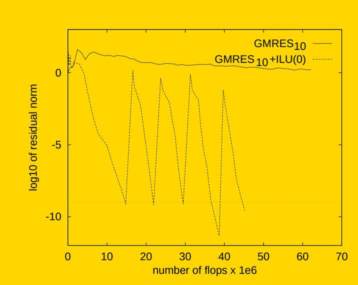
Both examples show that it pays to solve the correction equation more accurately. Although the projections become more expensive after convergence of a few eigenvectors, the computational costs to find the next eigenvectors does not increase. Apparently, the speed of convergence improves.

Example 1. *The eigenproblem of order 4096 comes from the stability analysis of an integrated circuit. No preconditioner has been used.*



Note that after an initial phase with two small bumps, JDQZ converges quite fast. For the next two eigenvalues there is no such initial stagnation. Apparently, in the iteration for the first eigenvalue, components for the next eigenvector are already collected in the search subspace.

Example 2. *The eigenproblem ("Brusselator") of order 2000 models the stability of concentration waves for reaction and transport interaction of chemical solutions in a tubular reactor.*



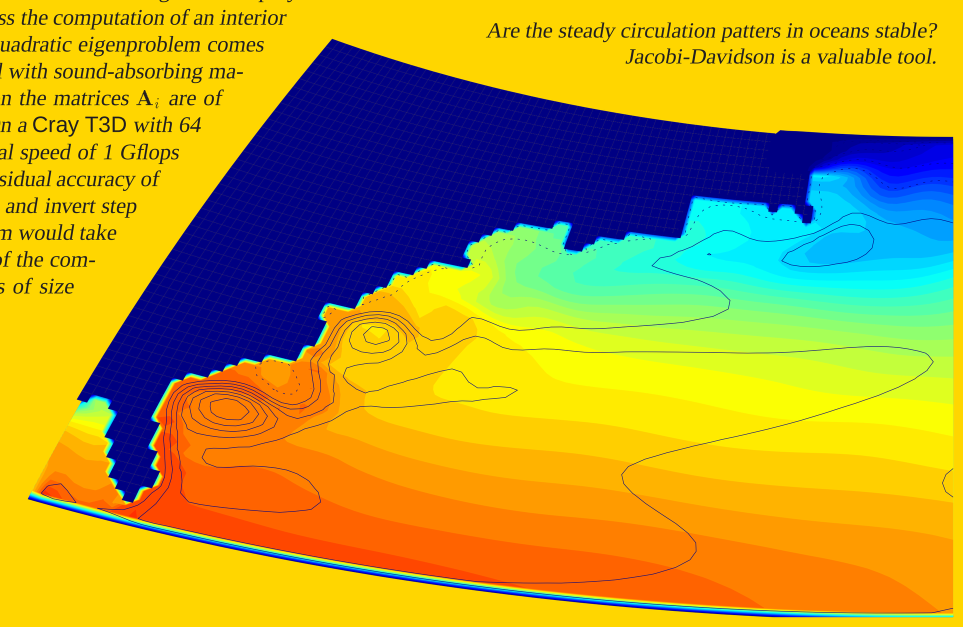
The preconditioner ILU(0) (incomplete LU of $\mathbf{A} - \mathbf{B}$*) for GMRES has been constructed only once, and appears to remain efficient also for eigenvalues farther away from the target* $\tau = 1$*.*

A large example

Jacobi-Davidson can also be used for interior eigenvalues and for higher order polynomial eigenproblems. As an illustration, we discuss the computation of an interior eigenvalue of $\mathbf{A}_0\mathbf{x} + \lambda\mathbf{A}_1\mathbf{x} + \lambda^2\mathbf{A}_2\mathbf{x} = \mathbf{0}$*. The quadratic eigenproblem comes from modelling sound waves in a room with a wall with sound-absorbing material. For the selected finite element discretization the matrices* $\mathbf{A}_i$ *are of size* $274\,625$*. Diagonal preconditioner was used. On a* Cray T3D *with 64 processors, it took 93.4 seconds with computational speed of 1 Gflops to compute the wanted eigenvalue with a relative residual accuracy of* 10^{-6}*. At peak performance (9.6 Gflops), one shift and invert step would take more than 10 minutes, the QZ-algorithm would take more than a month (if it would fit in the memory of the computer). The algorithm scales well as, for matrices of size* $136\,161$*, is shown by the table.*

Number of processors	*wall clock time*
16	*167.7*
32	*82.5*
64	*41.5*

Are the steady circulation patterns in oceans stable? Jacobi-Davidson is a valuable tool.



More Details

JD-algorithm — SLEIJPEN AND VAN DER VORST, SIMAX, 17 (1996), pp. 401–425.
JD general problems — SLEIJPEN, BOOTEN, FOKKEMA, VAN DER VORST, BIT, 36:3 (1996), pp. 595–633.
JDQR and JDQZ — FOKKEMA, SLEIJPEN, AND VAN DER VORST, SISC, 20 (1999), pp. 94–125.
Preconditioning — SLEIJPEN, VAN DER VORST, MEIJERINK, ETNA, 7 (1998), pp. 75–89.
SLEIJPEN AND WUBS, Preprint 1117, Dept. Math., UU, Aug. 1999.
GENSEBERGER, SLEIJPEN, VAN DER VORST, *Using domain decomposition in the Jacobi-Davidson Method*, in preparation

These papers and others: `www.math.uu.nl/people/sleijpen` `www.math.uu.nl/people/vorst`

Part of this work was perfomed in the NWO Massively Parallel Computing Project on 'Parallel Computational Magneto-Fluid Dynamcs' 95MPR04

Mathematical Institute
P.O. Box 80.010, 3508 TA Utrecht, NL
Universiteit Utrecht