

# ITERATIVE PROJECTION METHODS FOR SPARSE LINEAR SYSTEMS AND EIGENPROBLEMS

## CHAPTER 15 : PROJECTING NONLINEAR PROBLEMS

Heinrich Voss

voss@tu-harburg.de

Hamburg University of Technology  
Institute of Numerical Simulation



# Sparse nonlinear eigenproblems

Let  $T : D \rightarrow \mathbb{C}^{n \times n}$  be a family of matrices. Find  $\lambda \in D$  and  $x \in \mathbb{C}^n$ ,  $x \neq 0$  and/or  $y \in \mathbb{C}^n$ ,  $y \neq 0$  such that

$$T(\lambda)x = 0 \quad \text{and/or} \quad y^H T(\lambda) = 0.$$

For linear eigenvalue problems there are two types of iterative projection methods:

- Krylov subspace methods, which take advantage of normal forms of matrices (Schur form), the validity of some (Lanczos-, Arnoldi-) recurrence, and the fact that elements of  $\mathcal{K}_k(u, A)$  can be represented as  $x = p(A)u$  for some polynomial  $p \in \Pi_{k-1}$ .
- Methods where the search space is expanded by directions which have a high approximation potential for the eigenvector wanted next (like the Davidson or Jacobi-Davidson method).

Since normal forms are not known for general nonlinear eigenproblems generalizations of iterative projection methods to this type of problems always have to be of the second type

# Iterative projection methods for nonlinear problems

- nonlinear rational Krylov:  
Ruhe (2000,2005),  
Jarlebring, V. (2005).
- Arnoldi method:  
V. (2003, 2004);  
quadratic probl.: Meerbergen (2001)
- Jacobi-Davidson  
for polynomial problems:  
Sleijpen, Booten, Fokkema & van der Vorst (1996),  
Hwang, Lin, Wang, Wang (2003,2004)  
for general problems:  
Betcke, V. (2004) V. (2004),

# Dense nonlinear eigenproblems

In each step of an iterative projection method one has to solve a dense nonlinear eigenproblem of small dimension. We briefly survey methods for this type of problems.

For polynomial or rational eigenproblems the simplest approach is to use linearization and to apply standard methods for linear eigenproblems.

For general nonlinear eigenvalue problems, the classical approach is to formulate the eigenvalue problem as a system of nonlinear equations and to use variations of Newton's method or the inverse iteration method.

# Characteristic equation

For the characteristic equation

$$\det T(\lambda) = 0,$$

it was suggested by [Kublanovskaya](#) (1969,1970) to use a *QR*-decomposition with column pivoting  $T(\lambda)P(\lambda) = Q(\lambda)R(\lambda)$ , where  $P(\lambda)$  is a permutation matrix which is chosen such that the diagonal elements  $r_{jj}(\lambda)$  of  $R(\lambda)$  are decreasing in magnitude, i.e.  $|r_{11}(\lambda)| \geq |r_{22}(\lambda)| \geq \dots \geq |r_{nn}(\lambda)|$ . Then  $\lambda$  is an eigenvalue if and only if  $r_{nn}(\lambda) = 0$ .

Applying Newton's method to this equation, one obtains the iteration

$$\lambda_{k+1} = \lambda_k - \frac{1}{e_n^H Q(\lambda_k)^H T'(\lambda_k) P(\lambda_k) R(\lambda_k)^{-1} e_n}$$

for approximations to an eigenvalue.

Approximations to left and right eigenvectors can be obtained from

$$y_k = Q(\lambda_k) e_n \quad \text{and} \quad x_k = P(\lambda_k) R(\lambda_k)^{-1} e_n.$$

# Characteristic equation ct.

An improved version of Kublanovskaya's method was suggested by [Jain, Singhal & Huseyin](#) (1983), and also quadratic convergence was shown.

A similar approach was presented by [Yang](#) (1983), via a representation of Newton's method using the  $LU$ -factorization of  $T(\lambda)$ .

Other variations of this method can be found in the books of [Zurmühl & Falk](#) (1984, 1985).

However, this relatively simple idea is not efficient, since it computes eigenvalues one at a time and needs several  $\mathcal{O}(n^3)$  factorizations per eigenvalue. It is, however, useful in the context of iterative refinement of computed eigenvalues and eigenvectors.

# Successive linear approximations

Let  $\lambda_k$  be an approximation to an eigenvalue of  $T(\lambda)x = 0$ . Linearizing  $T(\lambda_k - \theta)x = 0$  yields

$$T(\lambda_k)x = \theta T'(\lambda_k)x.$$

This suggests the method

- 1: start with an approximation  $\lambda_1 \in D$  to an eigenvalue of  $T(\lambda)x = 0$
- 2: **for**  $k = 1, 2, \dots$  until convergence **do**
- 3:   solve the linear eigenproblem  $T(\lambda_k)x = \theta T'(\lambda_k)x$
- 4:   choose suitable eigenvalue  $\theta$  (usually smallest in modulus)
- 5:   set  $\lambda_{k+1} = \lambda_k - \theta$
- 6: **end for**

# Successive linear approximations ct.

## THEOREM

Let  $T(\lambda)$  be twice continuously differentiable, and let  $\hat{\lambda}$  be an eigenvalue of  $T(\lambda)x = 0$  such that  $T'(\hat{\lambda})$  is nonsingular and 0 is an algebraically simple eigenvalue of  $T'(\hat{\lambda})^{-1}T(\hat{\lambda})$ . Then the method of successive linear problems converges quadratically to  $\hat{\lambda}$ .

Sketch of proof: For

$$\Phi(x, \theta, \lambda) := \begin{pmatrix} T(\lambda)x - \theta T'(\lambda)x \\ \ell^H x - 1 \end{pmatrix}$$

it holds  $\Phi(\hat{x}, 0, \hat{\lambda}) = 0$ , and by the implicit function theorem  $\Phi(x, \theta, \lambda) = 0$  defines differentiable functions  $x : U(\hat{\lambda}) \rightarrow \mathbb{C}^n$  and  $\theta : U(\hat{\lambda}) \rightarrow \mathbb{C}$  on a neighborhood of  $\hat{\lambda}$  such that  $\Phi(x(\lambda), \theta(\lambda), \lambda) = 0$ .

$\theta'(\hat{\lambda}) = 1$  yields the quadratic convergence the method  $\lambda_{k+1} = \lambda_k - \theta(\lambda_k)$ .

# Inverse iteration

Applying Newton's method to the nonlinear system

$$F \begin{pmatrix} x \\ \lambda \end{pmatrix} = \begin{pmatrix} T(\lambda)x \\ \ell^H x - 1 \end{pmatrix},$$

( $\ell$  suitable scaling vector) yields

$$\begin{pmatrix} T(\lambda_k) & T'(\lambda_k)x_k \\ \ell^H & 0 \end{pmatrix} \begin{pmatrix} x_{k+1} - x_k \\ \lambda_{k+1} - \lambda_k \end{pmatrix} = - \begin{pmatrix} T(\lambda_k)x_k \\ \ell^H x_k - 1 \end{pmatrix}.$$

Assuming that  $x_k$  is already scaled such that  $\ell^H x_k = 1$  the second equation reads  $\ell^H x_{k+1} = \ell^H x_k = 1$ , and the first one

$$x_{k+1} = -(\lambda_{k+1} - \lambda_k) T(\lambda_k)^{-1} T'(\lambda_k)x_k.$$

Multiplying by  $\ell$  yields

$$\lambda_{k+1} = \lambda_k - \frac{\ell^H x_k}{\ell^H T(\lambda_k)^{-1} T'(\lambda_k)x_k}.$$

# Inverse iteration ct.

- 1: start with an approximation  $\lambda_1 \in D$  to an eigenvalue of  $T(\lambda)x = 0$  and an approximation  $x_1$  to an appropriate eigenvector
- 2: **for**  $k = 1, 2, \dots$  until convergence **do**
- 3:   solve  $T(\lambda_k)x_{k+1} = T'(\lambda_k)x_k$  for  $x_{k+1}$
- 4:   set  $\lambda_{k+1} = \lambda_k - \ell^H x_k / \ell^H x_{k+1}$
- 5:   normalize  $x_{k+1} \leftarrow x_{k+1} / \|x_{k+1}\|$
- 6: **end for**

## THEOREM

Let  $\hat{\lambda}$  be an eigenvalue of  $T(\cdot)$  such that  $\mu = 0$  is an algebraically simple eigenvalue of  $T'(\hat{\lambda})^{-1}T(\hat{\lambda})y = \mu y$ , and let  $\hat{x}$  be a corresponding eigenvector with  $\ell^H \hat{x} = 1$ . Then the inverse iteration converges locally and quadratically to  $(\hat{x}, \hat{\lambda})$ .

**Sketch of proof:** Only have to show that the only solution of

$$F'(\hat{x}, \hat{\lambda}) \begin{pmatrix} y \\ \mu \end{pmatrix} = \begin{pmatrix} T(\hat{\lambda}) & T'(\hat{\lambda})\hat{x} \\ \ell^H & 0 \end{pmatrix} \begin{pmatrix} y \\ \mu \end{pmatrix} = 0$$

is the trivial one.

# Inverse iteration ct.

The normalization condition can be modified in each step of inverse iteration. It was suggested in [Ruhe](#) (1973) to use  $\ell^k = T(\lambda_k)^H y^k$  for the normalization, where  $y^k$  is an approximation to a left eigenvector.

Then the update for  $\lambda$  becomes

$$\lambda_{k+1} = \lambda_k - \frac{(y^k)^H T(\lambda_k) x^k}{(y^k)^H T'(\lambda_k) x^k},$$

which is the Rayleigh functional for general nonlinear eigenproblems proposed in [Lancaster](#) (2002), and which can be interpreted as one Newton step for solving the equation  $f_k(\lambda) := (y^k)^H T(\lambda) x^k = 0$ .

For linear Hermitian eigenproblems this gives cubic convergence if  $\lambda_k$  is updated by the Rayleigh quotient. The same is true (cf. [Rothe](#) 1989) for symmetric nonlinear eigenproblems having a Rayleigh functional if we replace statement 4 by  $\lambda_{k+1} = \rho(u_{k+1})$ .

# Inverse iteration ct.

[Osborne](#) (1978) considers Newton's method for the complex function  $\beta(\lambda)$  defined by

$$T(\lambda)u = \beta(\lambda)x, \quad v^H u = \kappa,$$

where  $\kappa$  is a given constant, and  $x$  and  $u$  are given vectors.

This approach generalizes the method of Kublanovskaya, inverse iteration, and a method proposed in [Osborne & Michelson](#) (1964).

It was proved that the rate of convergence is quadratic, and that cubic convergence can be obtained if not only  $\lambda$ , but also  $x$  and/or  $s$  are updated appropriately, thus unifying the results in [Anselone & Rall](#) (1968), [Kublanovskaya](#) (1970), [Lancaster](#) (2002), [Osborne](#) (1964), and [Osborne & Michelson](#) (1964).

# Inverse iteration ct.

The disadvantage of inverse iteration with respect to efficiency is the large number of factorizations that are needed for each of the eigenvalues.

The obvious idea then is to use a version of a simplified Newton method, where the shift  $\sigma$  is kept fixed during the iteration, i.e. to use,

$$x_{k+1} = T(\sigma)^{-1} T'(\lambda_k) x_k \quad \text{for some fixed shift } \sigma$$

However, in general this method does not converge in the nonlinear case.

The iteration converges to an eigenpair of a linear problem

$$T(\sigma)x = \gamma T'(\tilde{\lambda})x,$$

from which one cannot recover an eigenpair of the nonlinear problem

$$T(\lambda)x = 0.$$

# Residual inverse iteration

If  $T(\lambda)$  is twice continuously differentiable,

$$\begin{aligned}
 x^k - x^{k+1} &= x^k + (\lambda_{k+1} - \lambda_k) T(\lambda_k)^{-1} T'(\lambda_k) x^k \\
 &= T(\lambda_k)^{-1} (T(\lambda_k) + (\lambda_{k+1} - \lambda_k) T'(\lambda_k)) x^k \\
 &= T(\lambda_k)^{-1} T(\lambda_{k+1}) x^k + \mathcal{O}((\lambda_{k+1} - \lambda_k)^2)
 \end{aligned}$$

Neglecting second order terms yields the update

$$x^{k+1} = x^k - T(\lambda_k)^{-1} T(\lambda_{k+1}) x^k,$$

replacing  $\lambda_k$  by a fixed shift  $\sigma$  yields an update

$$x^{k+1} = x^k - T(\sigma)^{-1} T(\lambda_{k+1}) x^k,$$

without misconvergence.

## Residual inverse iteration ct.

- 1: start with an approximation  $x_1 \in D$  to an eigenvector of  $T(\lambda)x = 0$
- 2: **for**  $k = 1, 2, \dots$  until convergence **do**
- 3: solve  $\ell^H T(\sigma)^{-1} T(\lambda_{k+1})x^k = 0$  for  $\lambda_{k+1}$   
(or  $(x^k)^H T(\lambda_{k+1})x^k = 0$  if  $T(\cdot)$  is Hermitian)
- 4: compute the residual  $r^k = T(\lambda_{k+1})x^k$
- 5: solve  $T(\sigma)d^k = r^k$
- 6: set  $x^{k+1} = x^k - d^k$ ,  $x^{k+1} = x^{k+1} / \|x^{k+1}\|$
- 7: **end for**

## THEOREM (Neumaier 1985)

Let  $T(\lambda)$  be twice continuously differentiable. Assume that  $\hat{\lambda}$  is a simple eigenvalue of  $T(\lambda)x = 0$ , and let  $\hat{x}$  be a corresponding eigenvector normalized by  $\|\hat{x}\| = 1$ . Then the residual inverse iteration converges for all  $\sigma$  sufficiently close to  $\hat{\lambda}$ , and it holds

$$\frac{\|x^{k+1} - \hat{x}\|}{\|x^k - \hat{x}\|} = \mathcal{O}(|\sigma - \hat{\lambda}|), \quad \text{and} \quad |\lambda_{k+1} - \hat{\lambda}| = \mathcal{O}(\|x^k - \hat{x}\|^t).$$

Here,  $t = 1$  in the general case, and  $t = 2$  if  $T(\cdot)$  is Hermitian.

# Safeguarded iteration

Let  $T(\lambda) \in \mathbb{C}^{n \times n}$ ,  $\lambda \in J \subset \mathbb{R}$  be a family of Hermitian matrices, such that the assumptions of the minmax characterization hold, and let  $p : D \rightarrow J$  denote the corresponding Rayleigh functional.

The minimum in

$$\lambda_j = \min_{\dim V=j, V \cap D \neq \emptyset} \max_{v \in V \cap D, v \neq 0} p(v)$$

is attained by the invariant subspace of  $T(\lambda_j)$  corresponding to the  $j$  largest eigenvalues, and the maximum by every eigenvector corresponding to the eigenvalue 0. This suggests

## Safeguarded iteration

- 1: Start with an approximation  $\mu_1$  to the  $\ell$ -th eigenvalue of  $T(\lambda)x = 0$
- 2: **for**  $k = 1, 2, \dots$  until convergence **do**
- 3:     determine eigenvector  $u$  corresponding to the  $\ell$ -largest eigenvalue of  $T(\mu_k)$
- 4:     evaluate  $\mu_{k+1} = p(u)$
- 5: **end for**

# Convergence of safeguarded iteration (V., Werner 1982)

- If  $\lambda_1 \in J$ , then for  $\ell = 1$  the safeguarded iteration converges globally to  $\lambda_1$
- If  $\lambda_\ell \in J$  is a simple eigenvalue, then the safeguarded iteration converges locally and quadratically.
- If  $T'(\lambda)$  is positive definite and  $x^k$  in Step 3 is replaced by an eigenvector of

$$T(\sigma_k)x = \mu T'(\sigma_k)x$$

corresponding to the  $\ell$ -th largest eigenvalue, then the convergence is even cubic.

- A variant exists which is globally convergent also for higher eigenvalues.

# Expansion of subspace

Given a search space  $V \subset \mathbb{C}^n$ . Expand  $V$  by a direction with high approximation potential for the next wanted eigenvector.

Let  $\theta$  be an eigenvalue of the projected problem

$$V^H T(\lambda) V y = 0$$

and  $x = V y$  corresponding Ritz vector, then inverse iteration yields a suitable candidate

$$v := T(\theta)^{-1} T'(\theta) x$$

**BUT:** In each step have to solve large linear system with varying matrix

# Arnoldi method

The residual inverse iteration requires to solve several linear systems with the same system matrix.

If  $\theta$  is an eigenvalue of the projected problem  $V^H T(\lambda) V y = 0$  and  $x = V y$  is a corresponding Ritz vector, then expand  $V$  by new direction

$$v = x - T(\sigma)^{-1} T(\theta) x$$

for some fixed shift  $\sigma$ .

In projection methods the new direction is orthonormalized against the previous ansatz vectors. Since the Ritz vector  $x$  is contained in  $\text{span } V$  we expand  $V$  by

$$v = T(\sigma)^{-1} T(\theta) x.$$

For the linear problem  $T(\lambda) = \lambda I - A$  this is exactly the Cayley transformation and the corresponding iterative projection method is the shift-and-invert Arnoldi method.

Therefore the resulting iterative projection method is called **nonlinear Arnoldi method** although no Krylov space is constructed and no Arnoldi recursion holds

# Nonlinear Arnoldi Method

- 1: start with initial basis  $V$ ,  $V^H V = I$ ; set  $k = m = 1$
- 2: determine preconditioner  $M \approx T(\sigma)^{-1}$ ,  $\sigma$  close to first wanted eigenvalue
- 3: **while**  $m \leq$  number of wanted eigenvalues **do**
- 4:   solve  $V^H T(\mu) V y = 0$  for  $(\mu, y)$  and set  $u = Vy$ ,  $r_k = T(\mu)u$
- 5:   **if**  $\|r_k\|/\|u\| < \epsilon$  **then**
- 6:     Accept eigenpair  $\lambda_m = \mu$ ,  $x_m = u$ ,
- 7:     **if**  $m ==$  number of wanted eigenvalues **then STOP end if**
- 8:      $m = m + 1$
- 9:     **if**  $\|r_{k-1}\|/\|r_k\| > \text{tol}$  **then**
- 10:       choose new pole  $\sigma$ , determine preconditioner  $M \approx T(\sigma)^{-1}$
- 11:     **end if**
- 12:     restart if necessary
- 13:     Choose approximations  $\mu$  and  $u$  to next eigenvalue and eigenvector
- 14:     determine  $r = T(\mu)u$  and set  $k = 0$
- 15:   **end if**
- 16:    $v = Mr$ ,  $k = k + 1$
- 17:    $v = v - VV^H v$ ,  $\tilde{v} = v/\|v\|$ ,  $V = [V, \tilde{v}]$  and reorthogonalize if necessary
- 18:   update projected problem
- 19: **end while**

# Comments on line 1

Here preinformation on eigenvectors can be introduced into the algorithm (approximate eigenvectors of contiguous problems in reanalysis, e.g.)

If no information on eigenvectors is at hand, and if we are interested in eigenvalues close to the parameter  $\sigma \in D$ :

choose initial vector at random, and execute a few Arnoldi steps for the linear eigenproblem  $T(\sigma)u = \theta u$  or  $T(\sigma)u = \theta T'(\sigma)u$ , and choose the eigenvector corresponding to the smallest eigenvalue in modulus or a small number of Schur vectors as initial basis of the search space.

Starting with a random vector without this preprocessing usually will yield a value  $\mu$  in step 3 which is far away from  $\sigma$  and will avert convergence.

# Comments on line 4

Since the dimension of the projected problem is small it can be solved by any dense solver like

- methods based on characteristic function  $\det T(\lambda) = 0$
- inverse iteration
- residual inverse iteration
- successive linear problems

For conservative gyroscopic systems: Determine Schur vectors of a (particular) linearization ([Meerbergen \(2001\)](#))

For symmetric problems allowing a minmax characterization of its eigenvalues this property is inherited by projected problems: therefore, eigenvalues can be determined one after the other by safeguarded iteration ([Betcke, V. \(2003\)](#)).

# Comments on line 4 ct.

For general problems if one wants eigenvalues in vicinity of a fixed  $\sigma_0$ , determine eigenvalue one after the other by successive linear approximations:

If  $j - 1$  eigenvalues are found and  $\lambda$  is an approximation to the next eigenvalue then iterate (V. (2003)):

```
while abs(incr) > tol do
  [v,  $\theta$ ] = eig(T( $\lambda$ ), T'( $\lambda$ ));
  [ $\mu$ , i] = sort(abs(diag( $\sigma_0 + \theta - \lambda$ )));
  incr =  $\theta(i(j), i(j))$ ;
  vv = v(:, i(j));
   $\lambda = \lambda - \text{incr}$ ;
end while;
```

# Comments on line 4 ct.

For a (small) perturbation  $T(\lambda)x = 0$  of a linear problem  $\tilde{T}(\lambda)x = Kx - \lambda Mx = 0$  (for instance for a finite element model of a vibrating structure with nonproportional damping) often the eigenmodes of both problems do not differ very much whereas the eigenvalues do.

Hence, it is at hand to determine an eigenvector  $y$  of the projected linear problem

$$V^H(K - \lambda M)Vy = 0$$

corresponding to a particular eigenvalue ( $m$ -smallest, e.g.), determine an approximate eigenvalue  $\tilde{\lambda}$  of the nonlinear problem from one of the complex equations

$$\ell^H V^H T(\sigma)^{-1} T(\lambda_m) Vy = 0 \quad \text{or} \quad y^H V^H T(\lambda) Vy = 0,$$

and correct  $(\tilde{\lambda}, y)$  by (residual) inverse iteration.

# Comments on lines 9 – 11:

Residual inverse iteration converges (at least) linearly where the contraction rate satisfies

$$\mathcal{O}(|\sigma - \hat{\lambda}|).$$

Therefore, the preconditioner is updated if the convergence (measured by the reduction of the residual norm in the final step before convergence) has become too large.

## Comments on line 12:

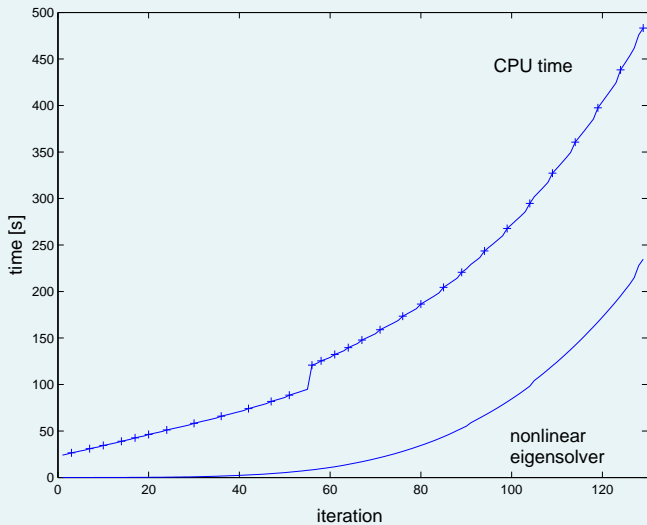
As the subspaces expand in the course of the algorithm the increasing storage or the computational cost for solving the projected eigenvalue problems may make it necessary to restart the algorithm and purge some of the basis vectors.

Since a restart destroys information on the eigenvectors and particularly on the one the method is just aiming at, we restart only if an eigenvector has just converged.

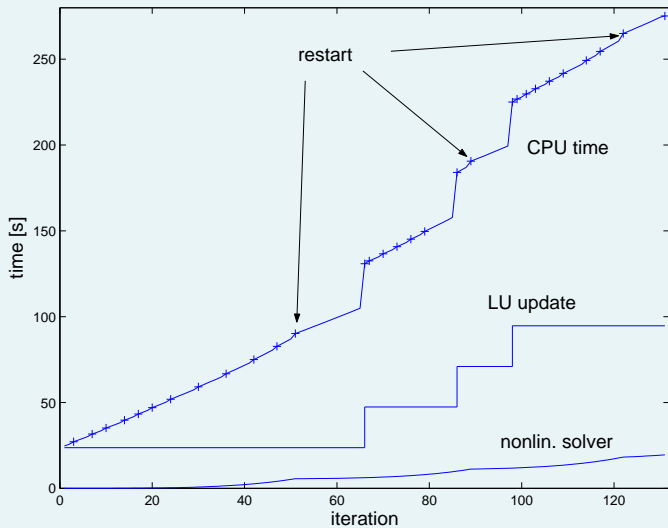
Reasonable search spaces after restart are

- the space spanned by the already converged eigenvectors (or a space slightly larger)
- an invariant space of  $T(\sigma)$  or eigenspace of  $T(\sigma)y = \mu T'(\sigma)y$  corresponding to small eigenvalues in modulus.

# Restarts



# Restarts



# Comment on line 13

If the projected problem in step 3. is solved by

- linearization
- the method of successive linear problems
- safeguarded iteration

which solve in each iteration step a linear eigenproblem then at the same time one gets approximations to further eigenpairs of the nonlinear problem which can be exploited to get a good initial approximation to the next wanted eigenpair.

# Comment on line 18

Often the family of matrices  $T(\lambda)$  has the form

$$T(\lambda) = \sum_{j=1}^p f_j(\lambda) C_j$$

with differentiable complex functions  $f_j$  and fixed matrices  $C_j \in \mathbb{C}^{n \times n}$ .

Then the projected problem has the form

$$T_{V_k}(\lambda) = \sum_{j=1}^p f_j(\lambda) V_k^H C_j V_k =: \sum_{j=1}^p f_j(\lambda) C_{j,k}$$

and the matrices  $C_{j,k}$  can be updated according to

$$C_{j,k+1} = \begin{pmatrix} C_{j,k} & V_k^H C_j V \\ V_k^H C_j V_k & V_k^H C_j V \end{pmatrix}.$$

# Locking of converged eigenvectors

A major problem with iterative projection methods for nonlinear eigenproblems when approximating more than one eigenvalue is to inhibit the method from converging to an eigenpair which was detected already previously.

- linear problems: (incomplete) Schur decomposition
- quadratic problems: Meerbergen (2001) based on linearization and Schur form of linearized problem (lock 2 vectors in each step)
- cubic problem: Hwang, Lin, Liu, Wang (2005) based on linearization and knowledge of all eigenvalues of the linearized problem
- Approach of Hwang et al. can be generalized directly to general problems if all eigenvalues of projected problems are determined
- for Hermitian problems one can often take advantage of a variational characterization of eigenvalues

# Jacobi-Davidson type expansion

Expand  $V$  by an approximate to  $v = x + \alpha T(\theta)^{-1} T'(\theta)x$ , where  $\alpha$  is chosen such that  $x^H v = 0$ , i.e.

$$v = x - \frac{x^H x}{x^H T(\theta)^{-1} T'(\theta)x} T(\theta)^{-1} T(\theta)x.$$

$v$  solves the [correction equation](#)

$$\left(I - \frac{T'(\theta)xx^H}{x^H T'(\theta)x}\right) T(\theta)(I - xx^H)t = -r, \quad t \perp x$$

For polynomial eigenvalue problems this expansion was introduced (and motivated in a different way) by [Sleijpen, Booten, Fokkema & van der Vorst \(1996\)](#), for general nonlinear eigenvalue problems by [Betcke, V. \(2004\)](#) and [V. \(2004\)](#).

# Nonlinear Jacobi-Davidson

- 1: Start with an initial basis  $V$ ,  $V^H V = I$ ;  $m = 1$
- 2: determine preconditioner  $K \approx T(\sigma)^{-1}$ ,  $\sigma$  close to first wanted eigenvalue
- 3: **while**  $m \leq$  number of wanted eigenvalues **do**
- 4:   compute approximation to  $m$ -th wanted eigenvalue  $\lambda_m$  and corresponding eigenvector  $x_m$  of projected problem  $V^H T(\lambda) V x = 0$
- 5:   determine Ritz vector  $u = V x_m$  and residual  $r = T(\lambda_m) u$
- 6:   **if**  $\|r\|/\|u\| < \epsilon$  **then**
- 7:       accept approximate eigenpair  $(\lambda_m, u)$ ; increase  $m = m + 1$ ;
- 8:       reduce search space  $V$  if necessary
- 9:       choose approximation  $(\lambda_m, u)$  to next eigenpair
- 10:      compute residual  $r = T(\lambda_m) u$ ; **end if**
- 11:   solve approximately (by preconditioned GMRES or BiCGStab, e.g.)  

$$(I - \frac{T'(\lambda_m) u u^H}{u^H T'(\lambda_m) u}) T(\lambda_m) (I - \frac{u u^H}{u^H u}) t = -r, \quad t \perp u.$$
- 12:   orthogonalize  $t = t - V V^H t$ ,  $v = t/\|t\|$ , and expand subspace  $V = [V, v]$
- 13:   determine new preconditioner  $K \approx T(\lambda_m)^{-1}$  if necessary
- 14:   update projected problem
- 15: **end while**

# Comment on step 11

The correction equation

$$\left(I - \frac{T'(\theta)xx^T}{x^T T'(\theta)x}\right)T(\theta)(I - xx^T)t = -r, \quad t \perp x$$

is solved approximately by a few steps of an iterative solver (GMRES or BiCGStab).

Similarly as for the linear eigenproblem preconditioning by

$$\tilde{K} := \left(I - \frac{px^H}{x^H p}\right)K\left(I - \frac{xx^H}{x^H x}\right)$$

where  $K \approx T(\sigma)$  is a preconditioner of  $T(\sigma)$  and  $p := T'(\theta)x$  can be implemented easily.

Taking into account the projectors in the preconditioner, i.e. using  $\tilde{K}$  instead of  $K$ , raises the cost of the preconditioned Krylov solver only slightly.

To initialize one has to solve the linear system  $K\tilde{p} = p$  and to determine the scalar product  $\alpha := x^H \tilde{p} = x^H K^{-1}p$ . These computations have to be executed just once.

# Comment on step 11

Hwang, Lin, Wang, Wang (2004) considered a non-iterative solution (which was already contained in the paper of Sleijpen et al. (1996), but was not considered in subsequent papers):

Solve

$$\left(I - \frac{T'(\theta)uu^H}{u^HT'(\theta)u}\right) \cancel{T(\theta)}^M \left(I - \frac{uu^H}{u^Hu}\right)t = -r, \quad t \perp u$$

approximately by computing

$$t = M^{-1}r + \alpha M^{-1}T'(\theta)u \quad \text{with } \alpha := \frac{u^HM^{-1}r}{u^HM^{-1}T'(\theta)u}$$

where  $M$  is a preconditioner of  $T(\theta)$ .

Method combines preconditioned Arnoldi method ( $M^{-1}r$ ) and simplified inverse iteration ( $M^{-1}T'(\theta)u$ )

# Comment on step 13

We preconditioned by an LU factorization of  $T(\sigma)$  for a shift close to the wanted eigenvalue. Any other preconditioner is fine, but the speed of convergence is reduced considerably with incomplete LU, e.g.

In the  $k$ -th outer iteration step for the  $n$ -th eigenvalue we iterated by GMRES until the residual was reduced by  $2^{-k}$ , but we allowed at most 10 GMRES steps.

If GMRES needed more than 5 iteration steps to satisfy these accuracy requirements we updated the preconditioner, but we allowed at most one update for every eigenvalue.

# Rational Krylov method

**Ruhe** (2000,2006) generalized the rational Krylov approach for linear eigenproblems to sparse nonlinear eigenvalue problems by nesting the linearization of  $T(\lambda)x = 0$  by Regula falsi and the solution of the resulting linear eigenproblem by Arnoldi's method, where the Regula falsi iteration and the Arnoldi recursion are knit together.

Similarly as in the rational Krylov process a sequence  $V_k$  of subspaces of  $\mathbb{C}^n$  is constructed, and at the same time Hessenberg matrices  $H_k$  are updated which approximate the projection of  $T(\sigma)^{-1}T(\lambda_k)$  to  $V_k$ .

Here  $\sigma$  denotes a shift and  $\lambda_k$  an approximation to the wanted eigenvalue of  $T(\cdot)$ .

Then a Ritz vector of  $H_k$  corresponding to an eigenvalue of small modulus approximates an eigenvector of the nonlinear problem from which a (hopefully) improved eigenvalue approximation of  $T(\lambda)x = 0$  is obtained.

Hence, in this approach the two numerical subtasks reducing the large dimension to a much smaller one and solving a nonlinear eigenproblem which are solved separately in the Arnoldi and the Jacobi–Davidson methods are attacked simultaneously.

# Rational Krylov method ct.

Linearizing the nonlinear family  $T(\lambda)$  by Lagrange interpolation between two points  $\mu$  and  $\sigma$  one gets

$$T(\lambda) = \frac{\lambda - \mu}{\sigma - \mu} T(\sigma) + \frac{\lambda - \sigma}{\mu - \sigma} T(\mu) + \text{higher order terms.} \quad (1)$$

Keeping  $\sigma$  fixed for several steps, iterating on  $\mu$ , neglecting the remainder in the Lagrange interpolation, and multiplying by  $T(\sigma)^{-1}$  from the right one obtains

$$T(\sigma)^{-1} T(\lambda_{j-1}) w = \theta w \quad \text{with } \theta = \frac{\lambda_j - \lambda_{j-1}}{\lambda_j - \sigma} \quad (2)$$

predicting a singularity at

$$\lambda_j = \lambda_{j-1} + \frac{\theta}{1 - \theta} (\lambda_{j-1} - \sigma). \quad (3)$$

If the dimension  $n$  is small then this linear eigenproblem can be used to approximate an eigenvalue of the nonlinear problem, and choosing the smallest eigenvalue of (2) in modulus for every  $j$  one can expect convergence to an eigenvalue close to the initial approximation  $\lambda_1$ .

# Rational Krylov method ct.

For large and sparse matrices Ruhe suggested to combine the linearization (1) with a linear Arnoldi process. Assume that the method has performed  $j$  steps, yielding approximations  $\lambda_1, \dots, \lambda_j$  to an eigenvalue, orthonormal vectors  $v^1, \dots, v^j$ , and an upper Hessenberg matrix  $H_{j,j-1} \in \mathbb{C}^{j \times (j-1)}$  such that the Arnoldi recursion

$$T(\sigma)^{-1} T(\lambda_{j-1}) V_{j-1} = V_j H_{j,j-1}, \quad (4)$$

is fulfilled (at least approximately), where  $V_j = [v^1, \dots, v^j]$ .

Updating the matrix  $H_{j,j-1}$  according to the linear theory yields

$$\tilde{H}_{j+1,j} = \begin{pmatrix} H_{j,j-1} & k^j \\ 0 & \|r_\perp\| \end{pmatrix} \quad (5)$$

where  $k^j = V_j^H r^j$ ,  $r^j = T(\lambda_j) v^j$ , and  $r_\perp = r^j - V_j V_j^H v^j$  which due to the nonlinearity of  $T(\cdot)$  violates the next Arnoldi relation

$$T(\sigma)^{-1} T(\lambda_j) V_j = V_{j+1} \tilde{H}_{j+1,j}, \quad v^{j+1} = r_\perp / \|r_\perp\|.$$

# Rational Krylov method ct.

To satisfy it at least approximately Ruhe took advantage of Lagrangean interpolation

$$A(\lambda_j) \approx \frac{\lambda_j - \sigma}{\lambda_{j-1} - \sigma} A(\lambda_{j-1}) - \frac{\lambda_j - \lambda_{j-1}}{\lambda_{j-1} - \sigma} I = \frac{1}{1 - \theta} A(\lambda_{j-1}) - \frac{\theta}{1 - \theta} I,$$

where  $A(\lambda) := T(\sigma)^{-1} T(\lambda)$ , and updated  $H$  according to

$$H_{j+1,j} = \begin{pmatrix} \frac{1}{1-\theta} H_{j,j-1} - \frac{\theta}{1-\theta} I_{j,j-1} & k_j \\ 0 & \|r_{\perp}\| \end{pmatrix} \quad (6)$$

arriving at a first version of the rational Krylov method:

- 1: Start with initial vector  $v^1$  with  $\|v^1\| = 1$ , and initial  $\lambda_1$  and  $\sigma$
- 2:  $r = T(\sigma)^{-1} T(\lambda_1) v^1$
- 3: **for**  $j = 1, 2, \dots$  **until convergence do**
- 4: orthogonalize  $h^j = V^H r$ ,  $r_{\perp} = r - V h^j$ ,  $h^{j+1,j} = \|r_{\perp}\|$
- 5:  $\theta = \min \text{eig } H_{j,j}$  with corresponding eigenvector  $s$
- 6:  $\lambda_{j+1} = \lambda_j + \frac{\theta}{1-\theta} (\lambda_j - \sigma)$
- 7:  $H_{j+1,j} = \frac{1}{1-\theta} H_{j+1,j} - \frac{\theta}{1-\theta} I_{j+1,j}$
- 8:  $v_{j+1} = r_{\perp} / \|r_{\perp}\|$
- 9:  $r = T(\sigma)^{-1} T(\lambda_{j+1}) v^{j+1}$  **end for**

# Rational Krylov method ct.

Since the method turned out to be inefficient Ruhe suggested to modify  $\lambda$ ,  $H$  and  $s$  in an inner iteration until the residual  $r = T(\sigma)^{-1} T(\lambda) V_j s$  is enforced to be orthogonal to  $V_j$ , and to expand the search space only after the inner iteration has converged.

If  $H_{j,j}$  has already been updated according to step 7: then  $H_{j,j} s = 0$ , and with

$$k^j = V_j^H T(\sigma)^{-1} T(\lambda) V_j s = V_j^H r$$

we have approximately

$$T(\sigma)^{-1} T(\lambda) V_j \begin{bmatrix} I_{j-1} & \tilde{s} \\ 0 & s_j \end{bmatrix} = V_j [H_{j,j-1}, k^j] + r(e^j)^T$$

where  $\tilde{s}$  is the leading  $j-1$  vector of  $s$ .

Multiplying by the inverse of the matrix in brackets from the right and by  $V_j^H$  from the left one gets the new Hessenberg matrix

$$\hat{H}_{j,j} = [H_{j,j-1}, k^j] \begin{bmatrix} I_{j-1} & -s_j^{-1} \tilde{s} \\ 0 & s_j^{-1} \end{bmatrix} = [H_{j,j-1}, -s_j^{-1} H_{j,j-1} \tilde{s} + s_j^{-1} k^j],$$

and  $H_{j,j-1} \tilde{s} + s_j h_s = 0$  finally yields that the last column of  $H_{j,j}$  has to be replaced by  $h^j + s_j^{-1} k^j$ .

# Rational Krylov algorithm

- 1: start with initial vector  $V = [v^1]$  with  $\|v^1\| = 1$ , initial  $\lambda$  and  $\sigma$ ; set  $j = 1$
- 2: set  $h^j = 0_j$ ;  $s = e^j$ ;  $x = v^j$ ;
- 3: compute  $r = T(\sigma)^{-1} T(\lambda)x$  and  $k^j = V_j^H r$
- 4: **while**  $\|k^j\| > \text{ResTol}$  **do**
- 5:   orthogonalize  $r = r - V_j^H k^j$
- 6:   set  $h^j = h^j + k^j s_j^{-1}$
- 7:    $\theta = \min \text{ eig } H_{j,j}$  with corresponding eigenvector  $s$
- 8:    $x = V_j s$
- 9:   update  $\lambda = \lambda + \frac{\theta}{1-\theta}(\lambda - \sigma)$
- 10:   update  $H_{j,j} = \frac{1}{1-\theta} H_{j,j} - \frac{1}{1-\theta} I$
- 11:   compute  $r = T(\sigma)^{-1} T(\lambda)x$  and  $k^j = V_j^H r$
- 12: **end while**
- 13: compute  $h_{j+1,j} = \|r\|$
- 14: **if**  $|h_{j+1,j} s_j| > \text{EigTol}$  **then**
- 15:    $v^{j+1} = r/h_{j+1,j}$ ;  $j = j + 1$ ; GOTO 2:
- 16: **end if**
- 17: Accept eigenvalue  $\lambda_i = \lambda$  and eigenvector  $x^i = x$
- 18: If more eigenvalues wanted, choose next  $\theta$  and  $s$ , and GOTO 8:

# Rational Krylov method ct.

Ruhe motivated the inner iteration and the requirement to make sure that the residual is orthogonal to the search space only by analogy to the linear case where it is satisfied automatically not being aware that the inner iteration is nothing else but a solver of the projected problem

$$V_j^H T(\sigma)^{-1} T(\lambda) V_j s = 0.$$

Hence, although motivated in a completely different way the rational Krylov method is an iterative projection method where the nonlinear eigenproblem  $T(\sigma)^{-1} T(\lambda)x = 0$  is projected to a search space  $V$ , and  $V$  is expanded by (the orthogonal complement of) the residual  $r = T(\sigma)^{-1} T(\lambda) V s$  of the Ritz pair (with respect to  $V$ ).

- 1: start with initial vector  $V = [v^1]$  with  $\|v^1\| = 1$ , initial  $\lambda$  and  $\sigma$
- 2: **for**  $j = 1, 2, \dots$  until convergence **do**
- 3:   solve projected eigenproblem  $V^H T(\sigma)^{-1} T(\lambda) V s = 0$  for  $(\lambda, s)$   
by inner iteration
- 4:   compute Ritz vector  $x = V s$  and residual  $r = T(\sigma)^{-1} T(\lambda) x$
- 5:   orthogonalize  $r = r - V V^H r$
- 6:   expand searchspace  $V = [V, r/\|r\|]$
- 7: **end for**

# Observations

- The inner iteration in step 3: can be replaced by any dense solver for nonlinear eigenproblems
- Numerical examples demonstrate (cf. [Jarlebring, V. 2005](#)) that the method can be accelerated considerably this way.
- But on the other hand, the dense solvers need the explicit form of the projected problem whereas the inner iteration of Ruhe only needs a procedure that yields the vector  $T(\sigma)^{-1} T(\lambda)x$  for a given  $x$ .
- A disadvantage of the rational Krylov method is that symmetry properties which the original problem may have are destroyed if the projected problem  $V^H T(\sigma)^{-1} T(\lambda)Vs = 0$  is considered instead of  $V_j^H T(\lambda)V_j s = 0$  in the Arnoldi method or the Jacobi–Davidson algorithm.

# AMLS for nonlinear eigenproblems

AMLS for linear eigenproblems is a one shot projection method which constructs a suitable subspace by condensation and mode truncation on several levels. This suggests for the nonlinear problem  $T(\lambda)x = 0$  the following approach:

Identify (essential) linear part

$$T(\lambda) := -K + \lambda M + R(\lambda),$$

where  $K \in \mathbb{C}^{n \times n}$  and  $M \in \mathbb{C}^{n \times n}$  are Hermitian and positive definite matrices, and

$$R(\lambda) = K - \lambda M - T(\lambda)$$

is a perturbation of the linear eigenproblem  $Kx = \lambda Mx$ , which is small in the eigenparameter set of interest.

- apply AMLS to linear pencil  $(K, M)$
- project  $T(\lambda)x = 0$  to a search space suggested by first step
- solve projected nonlinear eigenproblem

# AMLS for nonlinear eigenproblems ct.

Once the multi-level substructuring transformation of the linear pencil  $(K, M)$  has been accomplished with a given cut-off frequency we obtain a matrix  $\Phi_{\text{AMLS}}$  of substructure modes, and a projected eigenproblem

$$\mathcal{K}y = \lambda \mathcal{M}y$$

of much smaller dimension where  $\mathcal{K} = \Phi_{\text{AMLS}}^H K \Phi_{\text{AMLS}}$  and  $\mathcal{M} = \Phi_{\text{AMLS}}^H M \Phi_{\text{AMLS}}$ .

This information can be used in two ways to solve the nonlinear eigenvalue problem approximately:

(1) Project the nonlinear eigenproblem to the subspace of  $\mathbb{C}^n$  spanned by substructure modes which were kept in the AMLS reduction, i.e.

$$S(\lambda)y := \Phi_{\text{AMLS}}^H T(\lambda) \Phi_{\text{AMLS}} y = \mathcal{K}y - \lambda \mathcal{M}y - \Phi_{\text{AMLS}}^H R(\lambda) \Phi_{\text{AMLS}} y = 0.$$

In particular this projection can be performed easily if the remainder  $R(\lambda)$  has the form

$$R(\lambda) = \sum_{j=1}^p f_j(\lambda) C_j$$

where  $f_j(\lambda)$  are given complex functions and  $C_j \in \mathbb{C}^{n \times n}$  are given matrices.

# AMLS for nonlinear eigenproblems ct.

(2) Determine Ritz pairs  $(\lambda_j, x_j) := (\lambda_j, \Phi_{\text{AMLS}} y_j)$ ,  $j = 1, \dots, m$  of the linear problem  $Kx = \lambda Mx$  corresponding to eigenvalues in the wanted region, and project the nonlinear problem to the subspace spanned by these Ritz vectors.

Thus we get

$$S(\lambda)z := X^H T(\lambda) Xz = \Lambda z - \lambda z - X^H R(\lambda) Xz = 0$$

where  $\Lambda = \text{diag}\{\lambda_1, \dots, \lambda_m\}$  and  $X = (x_1, \dots, x_m)$ .

Projected problem is equivalent to the projection of the AMLS problem of (1) to the space spanned by the eigenvectors  $y_1, \dots, y_m$  of  $\mathcal{K}y = \lambda \mathcal{M}y$  corresponding to  $\lambda_1, \dots, \lambda_m$ .

Hence, we can expect, that the first approach will yield better approximations. Examples, however, demonstrate that the loss of accuracy is often negligible.

# AMLS for nonlinear eigenproblems ct.

In either case we arrive at a projected nonlinear eigenvalue problem of much smaller dimension which which preserves the structure of the original problem

Depending on the dimension it can be solved by

- a dense solver (based on characteristic function  $\det S(\lambda) = 0$ , inverse iteration, residual inverse iteration, successive linear problems, e.g.)
- dense method taking advantage of symmetry structure (safeguarded iteration, structure preserving linearization, e.g.)
- iterative projection methods (Arnoldi, Jacobi–Davidson, e.g.)

# Fluid-solid structure

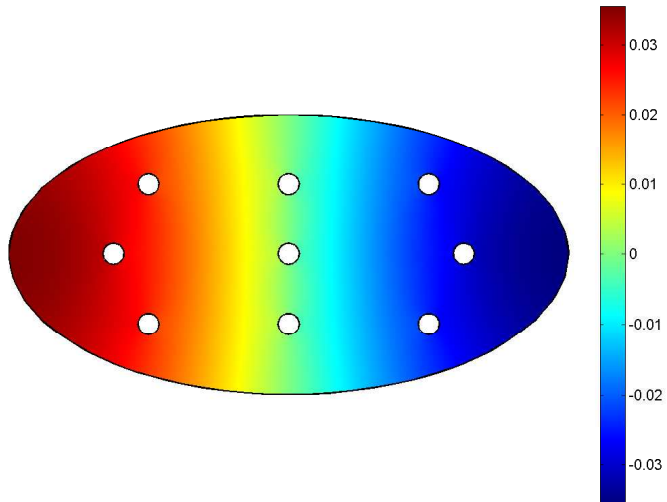
Consider a rational eigenvalue governing vibrations of a fluid-structure problem (elliptic cavity containing 9 structures)

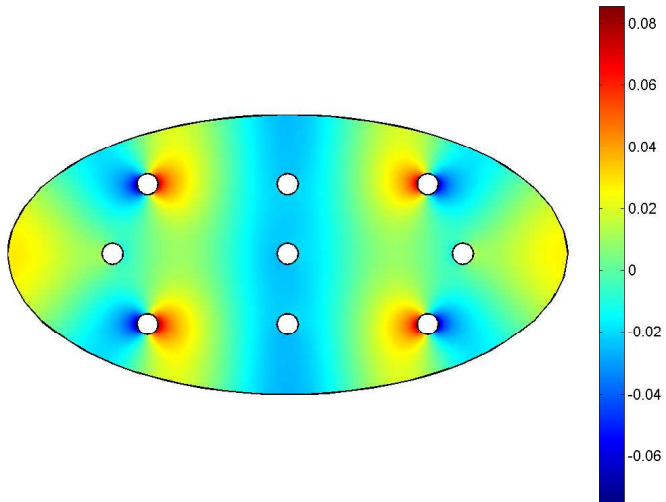
$$T(\lambda)x := -Kx + \lambda Mx + \frac{\lambda}{1-\lambda} Bx = 0 \quad (1)$$

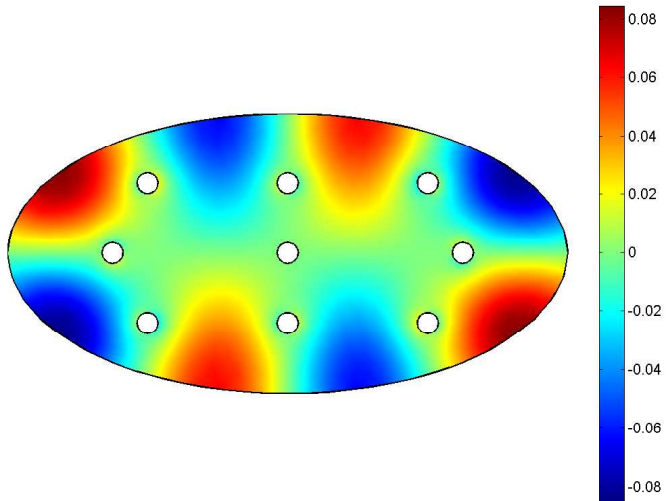
where  $K$ ,  $M$  and  $B$  are symmetric matrices of dimension 36040,  $K$  and  $B$  are positive semidefinite, and  $M$  is positive definite.

Problem (1) has 28 eigenvalues  $0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_{28} < 1$  in  $[0, 1)$  and a large number of eigenvalues  $1 < \tilde{\lambda}_{11} \leq \tilde{\lambda}_{12} \leq \dots$  in  $(1, \infty)$ , 20 of which are contained in the interval  $(1, 3)$ .

Notice that the linear eigenvalue problem  $Kx = \lambda Mx$  contains only 12 eigenvalues in  $(0, 1)$ . Thus, the rational eigenvalue problem (1) is not just a small perturbation of the linear problem which is obtained by neglecting the rational term.

$\lambda_2 = 0.0533306$  Surface:  $u$ 

$N\lambda(18)=0.767418$  Surface:  $u(u)$ 

$N\lambda(28)=0.905111$  Surface:  $u(u)$ 

# Theorem (Mazurenko, V., 2006)

Let  $n_k$  be the number of eigenvalues  $\lambda_j$  of the reduced problem:  
 Find  $\lambda \in \mathbb{R}$  and  $x \in H_k := \{x \in H : B_k x = 0\}$  such that

$$\left(K + \sum_{j=1}^{k-1} \frac{\sigma_k}{\sigma_k - \sigma_j} B_j\right) x = \lambda \left(M + \sum_{j=k+1}^p \frac{1}{\sigma_j - \sigma_k} B_j\right) x,$$

satisfying  $\lambda_j \leq \sigma_k$ , and let  $r_k$  be the codimension of  $H_k$ .

Then the rational eigenproblem

$$Kx = \lambda \left(M + \sum_{j=1}^p \frac{1}{\sigma_j - \lambda} B_j\right) x,$$

has  $n_{k+1} + r_{k+1} - n_k$  eigenvalues in  $(\sigma_k, \sigma_{k+1}]$ .

# Sketch of proof

For  $\mu \in (\sigma_k, \sigma_{k+1})$  consider the linear eigenproblem

$$\left( K + \sum_{j=1}^k \frac{\mu}{\mu - \sigma_j} B_j \right) x = \lambda \left( M + \sum_{j=k+1}^p \frac{1}{\sigma_j - \mu} B_j \right) x,$$

and denote by  $\lambda_m(\mu)$  the  $m$ -smallest eigenvalue. Then  $\lambda_m(\cdot) : (\sigma_k, \sigma_{k+1}) \rightarrow \mathbb{R}$  is a continuous and monotonely decreasing function, and  $\tilde{\lambda} \in (\sigma_k, \sigma_{k+1})$  is an eigenvalue of the nonlinear problem, if and only if it is a fixed point of  $\lambda_m$ .

It can be shown that  $\lim_{\mu \rightarrow \sigma_k^+} \lambda_m(\mu)$  is the  $m$ -smallest eigenvalue of the  $k$ -th restricted problem, that the  $n_{k+1}$  smallest eigenvalues  $\lambda_m(\mu)$  converge to 0 for  $\mu \rightarrow \sigma_{k+1}^-$ , and that the  $m + n_{k+1}$  smallest eigenvalue converges to the  $m$  smallest eigenvalue of the  $(k + 1)$ -st restricted problem.

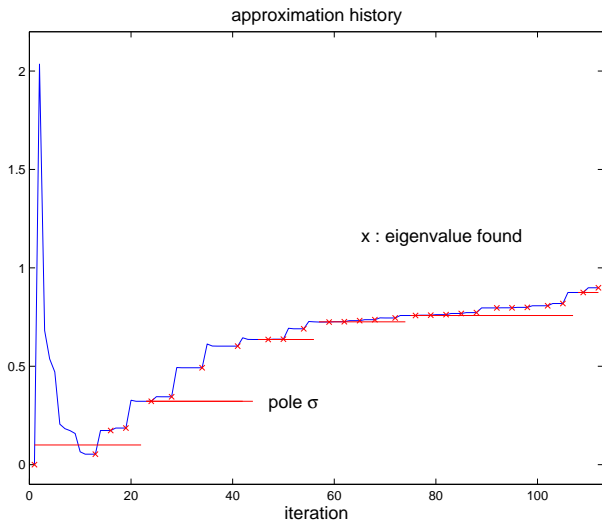


# Jacobi–Davidson method

The experiments were run under MATLAB 6.5 on a Pentium 4 processor with 2 GHz and 1 GB RAM.

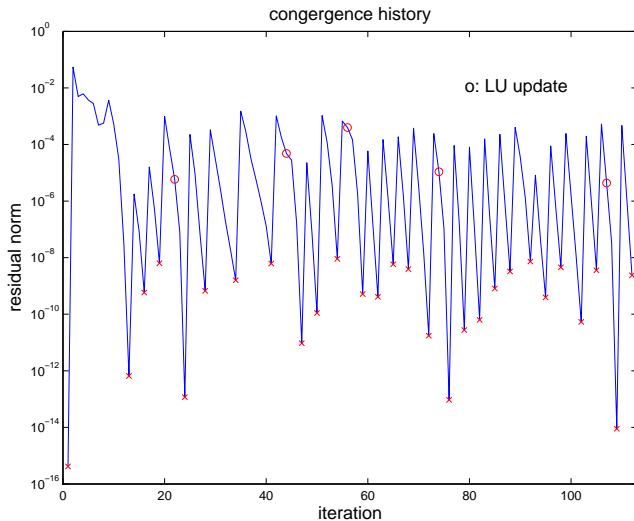
- outer iteration is terminated if the norm of the residual was less than  $10^{-8}$ . By the approximation properties of the Rayleigh functional then the eigenvalues are determined with full accuracy.
- correction equation is solved by preconditioned GMRES until residual is reduced by  $2^{-k}$  in  $k$ -th outer iteration; allowed at most 10 GMRES steps
- preconditioning by LU factorization of  $T(\sigma)$
- choose new preconditioner if more than 5 GMRES steps were necessary to reduce residual by  $2^{-k}$

# Approximation history

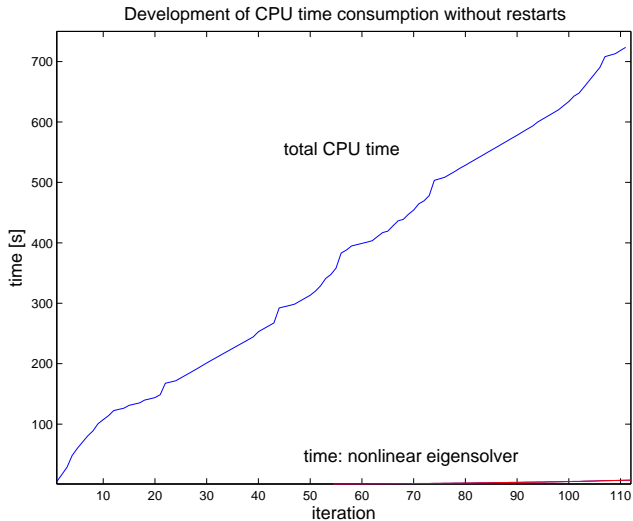


6 LU factorizations, 320 MatrixVector products in 209 GMRES steps

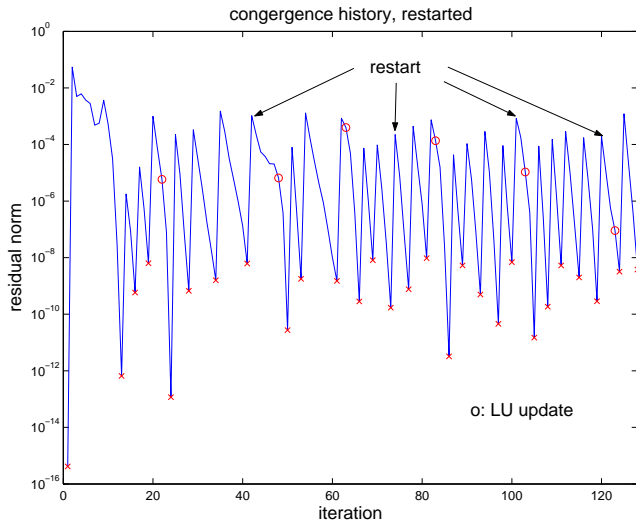
# Convergence history



# Time consumption



# Convergence history: restarted



## Second interval

For eigenvalues greater than 1 we started the Jacobi–Davidson method with the subspace spanned by the eigenvectors of the linear problems

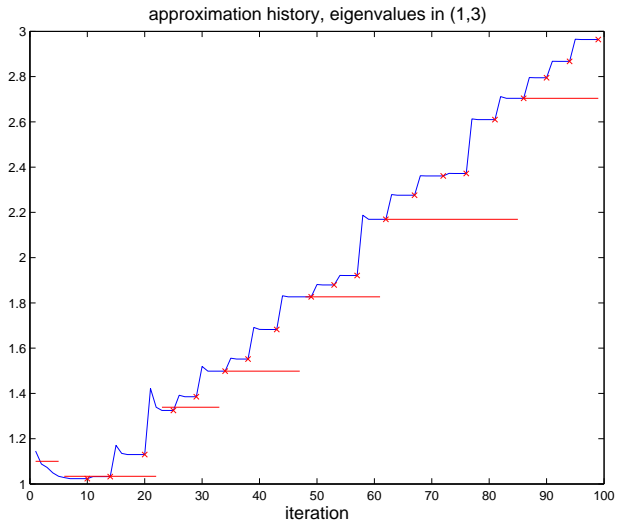
$$(K + \frac{\sigma}{\sigma - 1}B)x = \lambda Mx, \quad \sigma = 1.01$$

corresponding to eigenvalues smaller than  $\sigma$ .

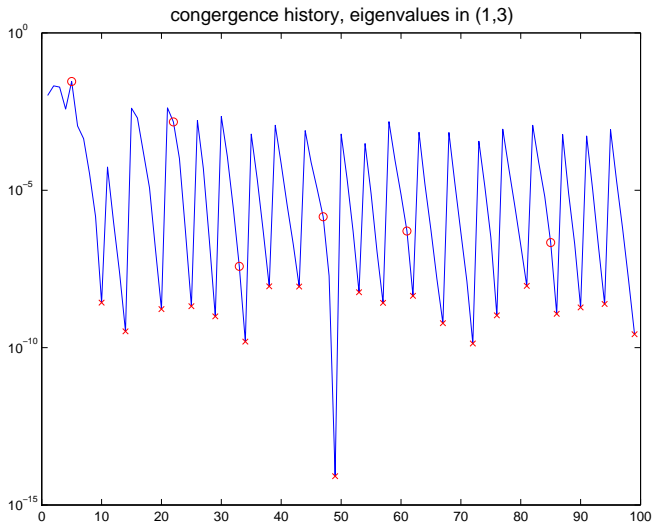
The method behaved similarly as for the interval  $(0, 1)$ . It found all eigenvalues  $\lambda_{11}, \dots, \lambda_{30}$  requiring 99 iteration steps, 165 GMRES steps with 264 matrix-vector products, 6 LU updates, and a CPU time of 628 seconds.

Restarting if the dimension of the search space exceeded 40 it required 110 iteration steps, 211 GMRES steps with 321 matrix-vector products, 8 LU updates, and a CPU time of 759 seconds.

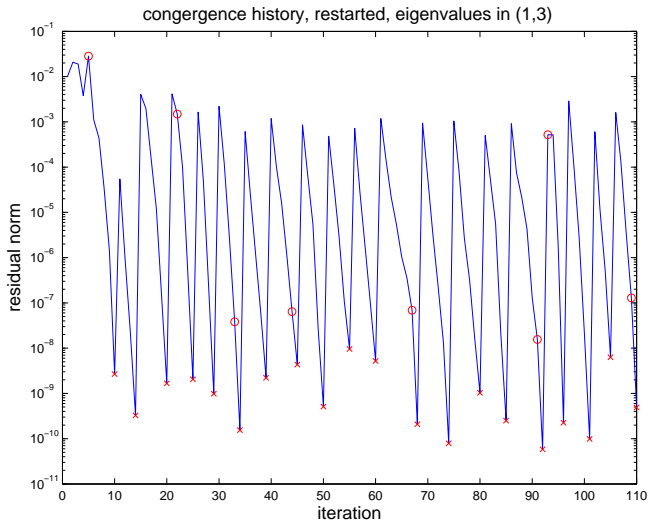
# Approximation history



# Convergence history



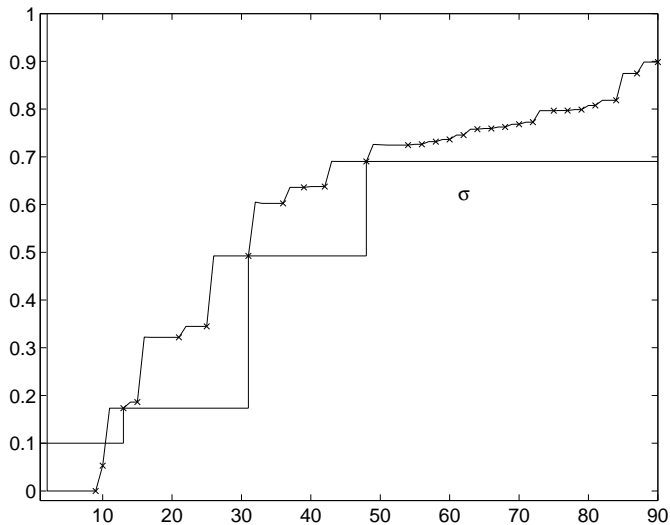
# Convergence history; restarted



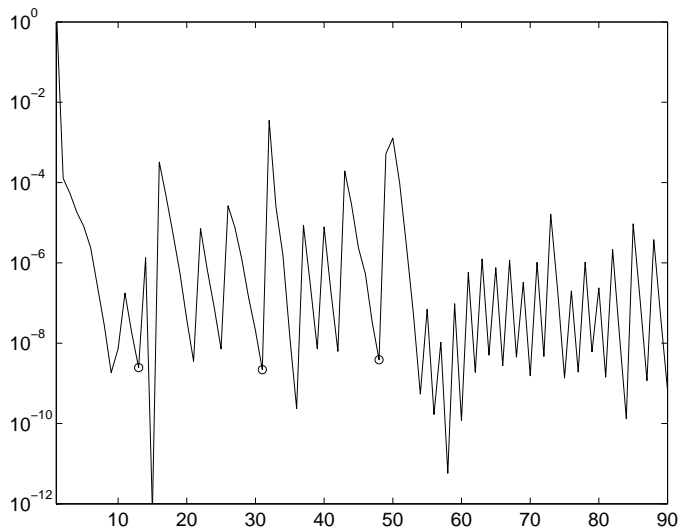
# Arnoldi method

Starting with the initial shift  $\sigma = 0.1$  and a random vector the algorithm without restarts needed 90 iteration steps and a CPU time of 87.7 seconds to approximate all 28 eigenvalues in the interval  $[0, 1)$ , and with the tolerance  $\text{tol} = 10^{-1}$  only 3 updates of the preconditioner were necessary.

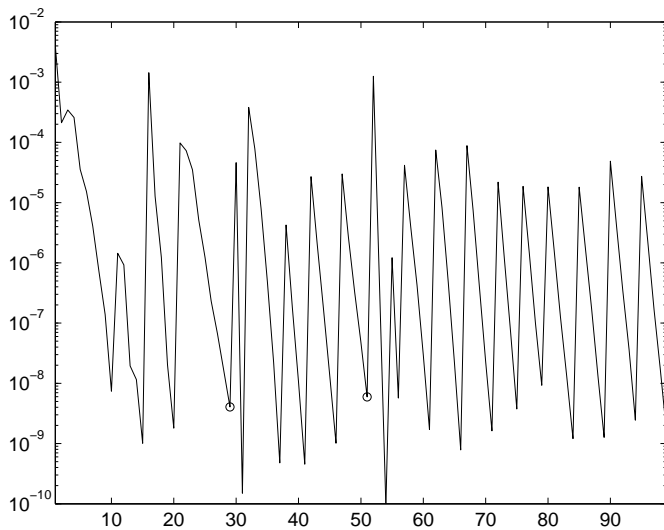
# Approximation history



# Convergence history



# Convergence history; interval (1, 3)



# Incomplete LU factorization

Preconditioning by incomplete LU factorization with drop tolerance 0.01 and restarting and updating the preconditioner whenever an eigenvalue has converged the algorithm finds all 28 eigenvalues in the  $[0, 1)$  requiring 1722 iterations and 1112 seconds, 4% of which are required to solve the projected nonlinear eigenproblems and 2.5% to update the preconditioner.

# Quantum dot

pyramidal quantum dot: baselength: 12.4 nm, height: 6.2 nm  
 cubic matrix:  $24.8 \times 24.8 \times 18.6 \text{ nm}^3$

Parameters (Hwang, Lin, Wang, Wang 2004)

$$P_1 = 0.8503, \quad g_1 = 0.42, \quad \delta_1 = 0.48, \quad V_1 = 0.0$$

$$P_2 = 0.8878, \quad g_2 = 1.52, \quad \delta_2 = 0.34, \quad V_2 = 0.7$$

Discretization by FEM or FVM yields rational eigenproblem

$$S(\lambda)x = \lambda Mx - \frac{1}{m_1(\lambda)}A_1x - \frac{1}{m_2(\lambda)}A_2x - Bx = 0$$

where  $S(\lambda)$  is symmetric and satisfies conditions of minmax characterization for  $\lambda \geq 0$

All timings for MATLAB 7.0.4 on AMD Opteron Processor  $248 \times 860_{64}$  with 2.2 GHz and 4 GB RAM

# Numerical example ct.

## FVM: Hwang, Lin, Wang, Wang (2004)

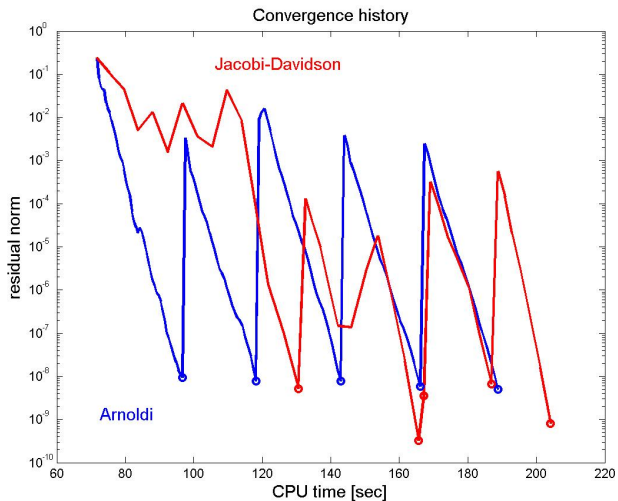
dim	$\lambda_1$	$\lambda_{2/3}$	$\lambda_4$	$\lambda_5$	CPU time
2'475	0.41195	0.58350	0.67945	0.70478	0.68 s
22'103	0.40166	0.57668	0.68418	0.69922	8.06 s
186'543	0.39878	0.57477	0.68516	0.69767	150.92 s
1'532'255	0.39804	0.57427	0.68539	0.69727	4017.67 s
12'419'775	0.39785	0.57415			overnight

## FEM: Cubic Lagrangian elements on tetrahedral grid

dimension: 96'640 ((Dof<sub>QD</sub>, Dof<sub>mat</sub>, Dof<sub>interf</sub>) = (43'615, 43'897, 9'128))

dim	$\lambda_1$	$\lambda_2$	$\lambda_3$	$\lambda_4$	$\lambda_5$	CPU time
96'640	0.39779	0.57411	0.57411	0.68547	0.69714	
Arnoldi	44 it.	29 it.	29 it.	24 it.	21 it.	188.8 s
JD	15 it.	9 it.	1 it.	7 it.	7 it.	204.4 s
HLWW	45 it.	49 it.	5 it.	24 it.	21 it.	226.7 s

# Convergence history



# Preconditioning

## incomplete LU with cut-off threshold $\tau$

$\tau$	JD	Arnoldi	HLWW	precond.
0.1	261.4	1084.1	1212.4	3.4
0.01	132.7	117.1	155.7	71.7
0.001	118.9	61.2	96.0	246.6
0.0001	155.6	46.6	71.1	665.6

## Sparse approximate inverse

$\tau$	JD	Arnoldi	HLWW	precond.
0.4	968.6	3105.7	4073.4	314.0
0.3	819.9	2027.6	2744.7	641.2
0.2	718.1	1517.5	2157.2	1557.0
0.1	694.9	1461.5	2124.4	1560.9

# Damped vibrations of a structure

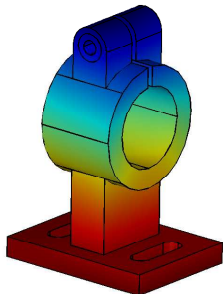
Using a viscoelastic constitutive relation to describe the material behavior in the equations of motion yields a rational eigenvalue problem in the case of free vibrations.

Discretizing by finite elements yields

$$T(\lambda)x := \left( \omega^2 M + K - \sum_{j=1}^K \frac{1}{1 + b_j \omega} \Delta K_j \right) x = 0$$

where  $M$  is the consistent mass matrix,  $K$  is the stiffness matrix with the instantaneous elastic material parameters used in Hooke's law, and  $\Delta K_j$  collects the contributions of damping from elements with relaxation parameter  $b_j$ .

# Nonproportional damping ct.



## FE model of feeder clamp

linear Lagrangian elements on tetrahedral grid

dimension	193'617
nnz(K)	7'670'533
nnz(M)	2'557'851

Determine 30 eigenvalues with maximal negative imaginary part.

Using a viscoelastic constitutive relation to describe the material behavior in the equations of motion yields a rational eigenvalue problem in the case of free vibrations.

# Nonproportional damping ct.

## Iterative projection methods

Preconditioner threshold		Arnoldi		Jacobi–Davidson		rational Krylov	
	CPU	# iter.	CPU	# iter.	CPU	# iter.	CPU
$10^{-2}$	41	601	4137	166	1736	412	9033
$10^{-3}$	127	226	2004	129	1144	163	3053
$10^{-4}$	348	109	847	105	1203	78	2268

## AMLS

cut-off freq.	CPU time	dim.	CPU solve	max. rel. error
$1.2e7$	837 sec.	1262	7.1 sec.	$2.1e-3$
$2.4e7$	996 sec.	2808	24.1 sec.	$8.1e-4$

# Fluid–solid vibrations

Consider the fluid–solid vibration problem for an elliptic cavity with 9 tubes of 3 different types

$$T(\lambda)x = \left(-K + \lambda M + \frac{\lambda}{1-\lambda} C_1 C_1^T + \frac{\lambda}{2-\lambda} C_2 C_2^T + \frac{\lambda}{3-\lambda} C_3 C_3^T\right)x = 0$$

of dimension  $n = 143063$ .

This Problem has 18, 15, and 14 eigenvalues in the interval  $J_1 = (0, 1)$ ,  $J_2 = (1, 2)$ , and  $J_3 = (2, 3)$ , respectively, and a large number of eigenvalues in  $(3, \infty)$ , 18 of which are contained in  $J_4 := (3, 5)$ .

# Problem ct.

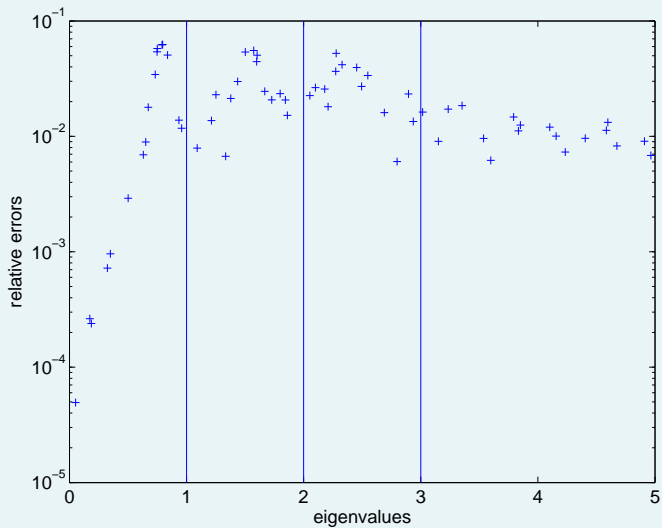
In each of the intervals  $J_j$  the eigenvalues can be characterized as minmax values of a Rayleigh functional, and they can be determined one after the other by iterative projection methods.

method	CPU time
Arnoldi	420 sec.
Jacobi-Davidson	1033 sec.

Reducing the problem by AMLS with the base problem  $Kx = \lambda Mx$  and a cut-off frequency of 100 generates a rational eigenvalue problem of the same structure of dimension 888 which can be solved by the nonlinear Arnoldi method.

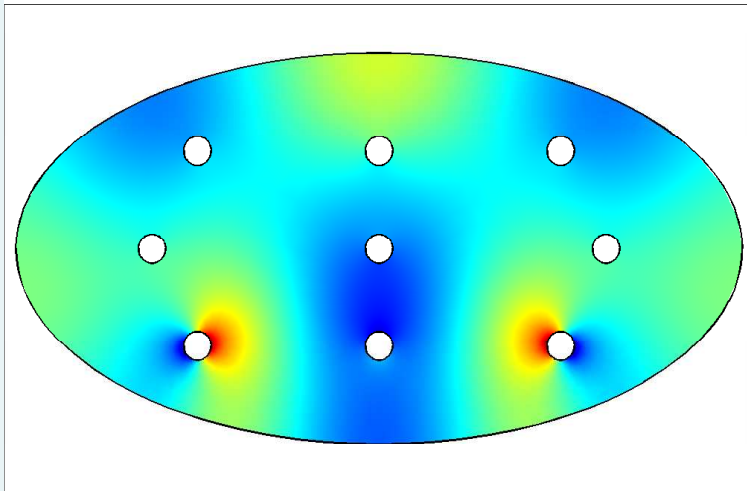
CPU time: Reduction (231 sec.)+Arnoldi (35 sec.) = 266 sec.

# Relative errors



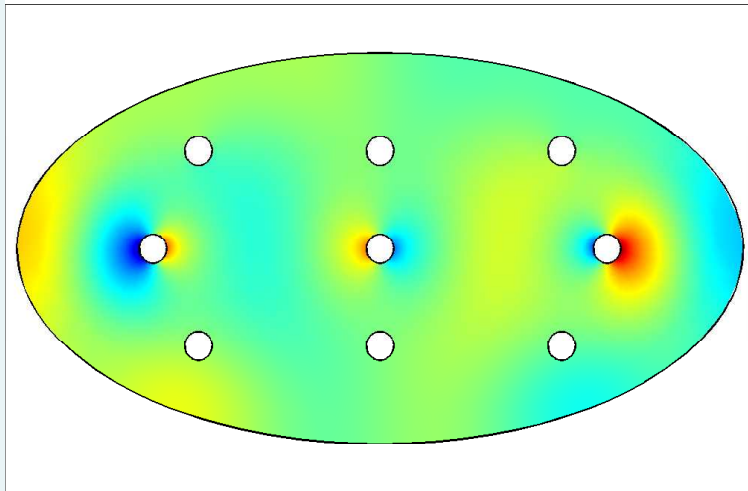
# Typical eigenvector

$\lambda_{13} = 0.7506$



# Typical eigenvector

$\lambda(27)=1.5988$



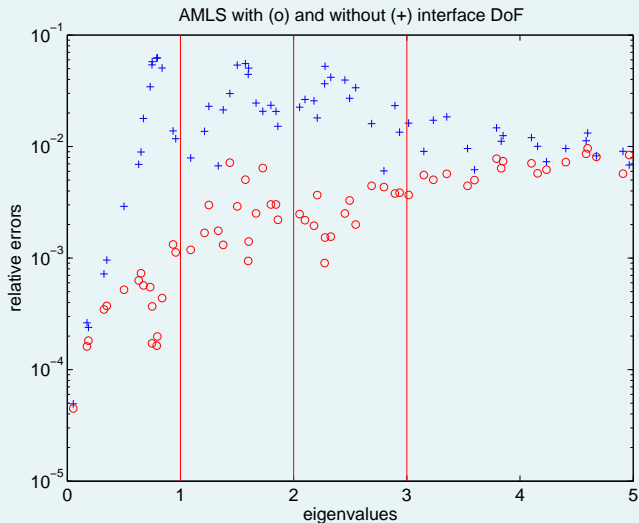
# Problem ct.

Complement the 503 interface degrees of freedom on the coarsest level generated by METIS for the base problem  $Kx = \lambda Mx$  by the 1728 components corresponding to nonzero row entries of the matrix  $[C_1, C_2, C_3]$ .

Dimension of the AMLS projected problem: 902

CPU time: Reduction (341 sec.)+Arnoldi (35 sec.) = 376 sec.

# Relative errors



# Gyroscopic eigenproblems

Consider the gyroscopic eigenvalue problem

$$Q(\omega)x := Kx + i\omega Gx - \omega^2 Mx = 0 \quad (1)$$

governing eigenvibrations of rotating structures. Here  $K$  is the stiffness matrix modified by the presence of centripetal forces,  $M$  is the mass matrix, and  $G$  is the gyroscopic matrix stemming from the Coriolis force. Clearly,  $K$  and  $M$  are symmetric and positive definite, and  $G$  is skew-symmetric.

For example, this problem arises when modeling noise of rolling tires which is the major source of traffic noise for passenger cars at speed exceeding 60 km/h. Due to the complicated interior structure of a belted tire the matrices  $K$ ,  $M$  and  $G$  of a sufficiently accurate FE model are very large and sparse. Moreover, for the acoustic analysis many eigenpairs not necessarily at the end of the spectrum are needed. Therefore, well-established sparse eigensolvers of Arnoldi type with shift and invert techniques for a linearization or iterative projection methods for nonlinear eigenproblems are very costly since LU factorizations of complex valued matrices  $Q(\omega_j)$  for several parameters  $\omega_j$  are required.

# Gyroscopic eigenproblems ct.

Since the influence of the gyroscopic matrix  $G$  on the eigenvalues is usually not very high compared to the mass and stiffness matrix, it is reasonable to neglect the linear term  $i\omega G$  when defining the essential linear eigenproblem  $Kx = \omega^2 Mx$ .

Since the sparsity pattern of  $G$  matches the ones of  $K$  and  $M$  one gets the reduced model

$$Ky + i\omega Gy - \omega^2 My = 0,$$

when applying the AMLS reduction to  $Kx = \omega^2 Mx$ , and projecting the matrix  $G$  simultaneously.

Here the stiffness and mass matrix have the same structure as in the linear case, and the gyroscopic matrix  $G$  is a skew-symmetric block matrix containing diagonal blocks corresponding to the (reduced) substructures and interfaces, and only off-diagonal blocks describing the coupling of a substructure and its interface contain non-zero elements. **Notice, that all projectors are real, and therefore the reduction can be performed in real arithmetic.**

# Gyroscopic eigenproblems ct.

If the dimension of the reduced problem is very small, a method at hand is to consider the linearization

$$\begin{pmatrix} i\mathcal{G} & \mathcal{K} \\ \mathcal{K} & \mathbf{O} \end{pmatrix} \begin{pmatrix} \omega y \\ y \end{pmatrix} = \omega \begin{pmatrix} \mathcal{M} & \mathbf{O} \\ \mathbf{O} & \mathcal{K} \end{pmatrix} \begin{pmatrix} \omega y \\ y \end{pmatrix}$$

and to apply any dense solver.

For very large gyroscopic problems (for instance a realistic model of a rolling tire) the dimension of the projected problem will still be quite large. In this case the reduced problem can be solved by an iterative projection method taking advantage of the minmax characterization of its positive eigenvalues or by a sparse solver of the linearized problem like ARPACK. In both cases the solution requires complex arithmetic.

# Numerical example

Consider a tire model with 39204 brick elements, 124992 degrees of freedom and 20 different material groups, rotating with 50 km/h. Our aim is to determine approximations to the smallest 180 eigenvalues with relative error less than 1% and the corresponding eigenvectors.

Linearizing in the usual way

$$\begin{pmatrix} -iG & -K \\ I & O \end{pmatrix} \begin{pmatrix} \omega x \\ x \end{pmatrix} = \omega \begin{pmatrix} M & O \\ O & I \end{pmatrix} \begin{pmatrix} \omega x \\ x \end{pmatrix}$$

or by the Hermitian problem

$$\begin{pmatrix} iG & K \\ K & O \end{pmatrix} \begin{pmatrix} \omega x \\ x \end{pmatrix} = \omega \begin{pmatrix} M & O \\ O & K \end{pmatrix} \begin{pmatrix} \omega x \\ x \end{pmatrix}$$

and applying the shift-and-invert Arnoldi method requires an LU factorization of  $Q(\omega)$  for every shift  $\omega$ , which is a complex matrix.

# Numerical example ct.

Determining the factorization by SuperLU requires a memory of 6.04 GByte and a CPU time of 3910 seconds on one PA-RISC (750 MHz) processor of an HP superdome.

Applying the nonlinear Arnoldi method the preconditioners can be chosen as real matrices  $K - \omega^2 M$ , the LU factorization of which requires 2.7 GByte storage and 1940 seconds with SuperLU, and 2.86 GByte storage and 1080 seconds with the multi frontal solver MA57 of HSL.

Since the LU factorization has to be updated several times a total CPU time of more than 12 hours results on one processor of the superdome.

# Numerical example ct.

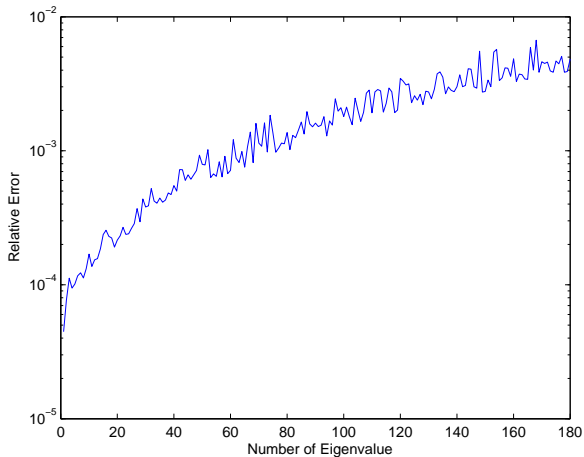
AMLS demands much less storage and the problem under consideration can be solved on a personal computer, namely a Pentium 4 processor with 3.0 GHz and 1 GByte storage.

With a cut-off frequency of  $\omega_c = 2 \times 10^5$  the problem is projected to a gyroscopic eigenproblem of dimension  $n_c = 2697$  requiring a CPU time of 1187 seconds.

Solving the linearization of the projected problem by `eigs` (i.e. by ARPACK) under MATLAB 7.0 requires another 166.1 seconds.

# Numerical example ct.

The relative errors of 180 eigenvalues are all less than 0.67%.



# Numerical example ct.

Projecting to the subspace spanned by the Ritz vectors corresponding to eigenvalues of the linear problem

$$\mathcal{K}y = \lambda \mathcal{M}y$$

not exceeding  $1.5\omega_{\max}^2$  where  $\omega_{\max} = 12000$  is the maximal wanted eigenvalue one gets a gyroscopic eigenproblem of dimension 262.

The accuracy of the approximations to the 180 smallest eigenvalues is deteriorated only slightly. The maximum relative error is raised only to 0.69%.

Thus, the solution time of the projected problem is reduced to 39.6 seconds.