Computing the smallest eigenvalue of symmetric tridiagonal matrices via semiseparable techniques.

Raf Vandebril, Marc Van Barel,
Nicola Mastronardi, Steven Delvaux & Yvette Vanberghen

Department of Computer Science
K.U.Leuven

July 2006

Outline

Introduction
- Goal of the talk
- Semiseparable matrices
- The QR-algorithm

QR-iterations on the inverse

An implicit approach

Conclusions & Future work

Outline

Introduction
- Goal of the talk
- Semiseparable matrices
- The QR-algorithm

QR-iterations on the inverse

An implicit approach

Conclusions & Future work

Accurately computing the smallest eigenvalue

- Suppose we have a symmetric tridiagonal $n \times n$ matrix T, with one small eigenvalue.

$$|\lambda_1| \approx 10^{-x} \quad \text{with} \quad 6 \leq x \leq 14$$
$$|\lambda_i| \approx 1 \quad \text{with} \quad 2 \leq i \leq n$$

  The smallest one often suffers from loss in accuracy.
- Our aim is to compute the eigenvalue λ_1 as accurate as possible, in linear time.
- We will base ourselves on the QR-algorithm.
- We will make use of semiseparable matrices. These matrices are the inverses of tridiagonal matrices.

Outline

Introduction
- Goal of the talk
- Semiseparable matrices
- The QR-algorithm

QR-iterations on the inverse

An implicit approach

Conclusions & Future work

A semiseparable matrix

- A matrix A is called semiseparable if all submatrices taken out of the lower triangular part have rank 1. (Similar for the upper triangular part.)

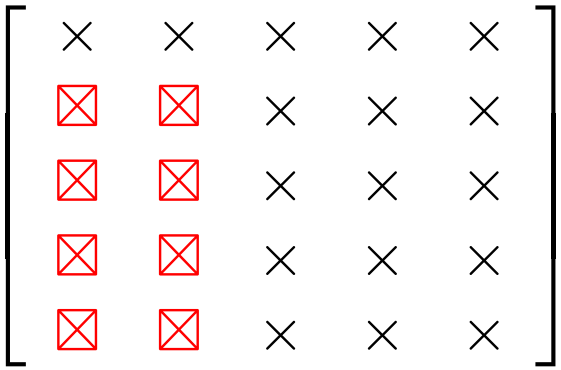
A semiseparable matrix

- A matrix A is called semiseparable if all submatrices taken out of the lower triangular part have rank 1. (Similar for the upper triangular part.)
- All the red blocks are of rank 1.



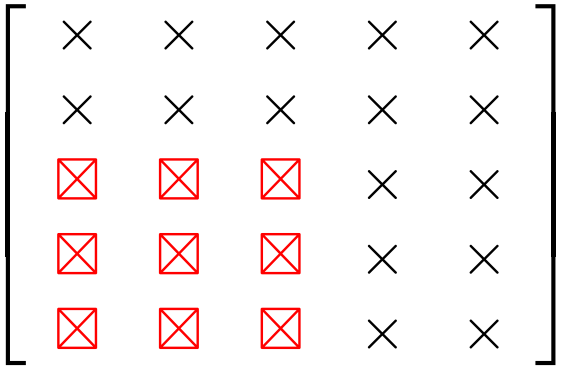
A semiseparable matrix

- A matrix A is called semiseparable if all submatrices taken out of the lower triangular part have rank 1. (Similar for the upper triangular part.)
- All the red blocks are of rank 1.



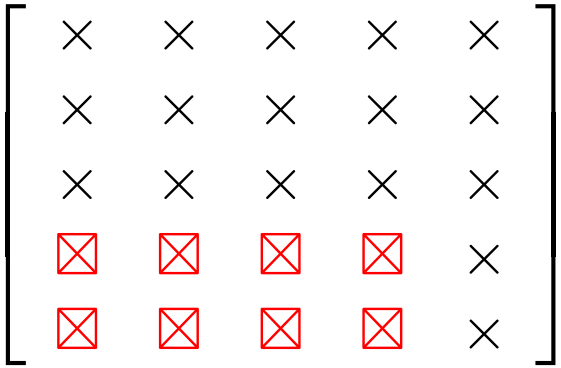
A semiseparable matrix

- A matrix A is called semiseparable if all submatrices taken out of the lower triangular part have rank 1. (Similar for the upper triangular part.)
- All the red blocks are of rank 1.



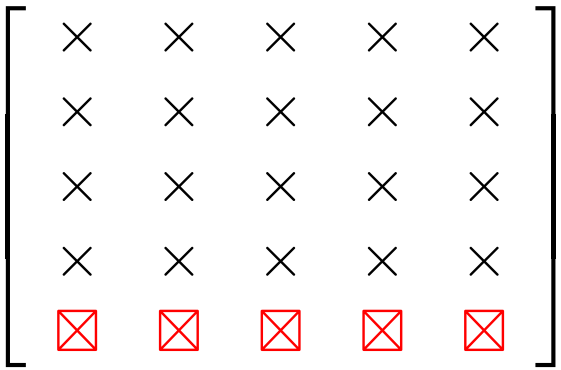
A semiseparable matrix

- A matrix A is called semiseparable if all submatrices taken out of the lower triangular part have rank 1. (Similar for the upper triangular part.)
- All the red blocks are of rank 1.



A semiseparable matrix

- A matrix A is called semiseparable if all submatrices taken out of the lower triangular part have rank 1. (Similar for the upper triangular part.)
- All the red blocks are of rank 1.



- The inverse of a tridiagonal matrix is semiseparable and can be computed in $O(n)$ operations.

Inversion

Suppose we have a matrix T, inverting this matrix is almost straightforward using:

- LU-factorization (L is lower bidiagonal, U is upper bidiagonal):

$$T = LU \quad \rightarrow \quad S = T^{-1} = U^{-1}L^{-1},$$

with U^{-1} of upper triangular semiseparable form and L^{-1} of lower triangular semiseparable form.

- QR-factorization (Q is a unitary Hessenberg matrix, R is upper triangular, with two nonzero superdiagonals):

$$T = QR \quad \rightarrow \quad S = T^{-1} = R^{-1}Q^{-1},$$

with Q^{-1} unitary upper Hessenberg and R^{-1} an upper triangular semiseparable matrix of semiseparability rank 2.

Inversion

Or via direct inversion, with

$$T = \begin{bmatrix} a_1 & b_1 & & \\ b_1 & a_2 & b_2 & \\ & b_2 & a_3 & b_3 \\ & & b_3 & a_4 \end{bmatrix}$$

and

$$S = \begin{bmatrix} c_1v_1 & c_2s_1v_1 & c_3s_2s_1v_1 & s_3s_2s_1v_1 \\ c_2s_1v_1 & c_2v_2 & c_3s_2v_2 & s_3s_2v_2 \\ c_3s_2s_1v_1 & c_3s_2v_2 & c_3v_3 & s_3v_3 \\ s_3s_2s_1v_1 & s_3s_2v_2 & s_3v_3 & v_4 \end{bmatrix}.$$

Outline

Introduction
- Goal of the talk
- Semiseparable matrices
- The QR-algorithm

QR-iterations on the inverse

An implicit approach

Conclusions & Future work

The QR-algorithm

- Suppose we have an $n \times n$ matrix A. Computing the matrix $\hat{A}$, gives us one step of the QR-algorithm with shift κ:

$$\begin{aligned} A - \kappa I &= QR \\ \hat{A} &= RQ + \kappa I. \end{aligned}$$

  $\hat{A}$ converges to the largest eigenvalue, and on the bottom right to the smallest eigenvalue, w.r.t. the shift!!
- Repeating this procedure gives us the standard QR-algorithm. This method lets $\hat{A}$ converge to a diagonal matrix, with the eigenvalues on the diagonal.
- One step of the QR-method applied on a semiseparable or on a tridiagonal matrix costs $O(n)$ operations.

Outline

Introduction

QR-iterations on the inverse
- QR-algorithm with shift
- QR-algorithm without shift

An implicit approach

Conclusions & Future work

Outline

Introduction

QR-iterations on the inverse
- QR-algorithm with shift
- QR-algorithm without shift

An implicit approach

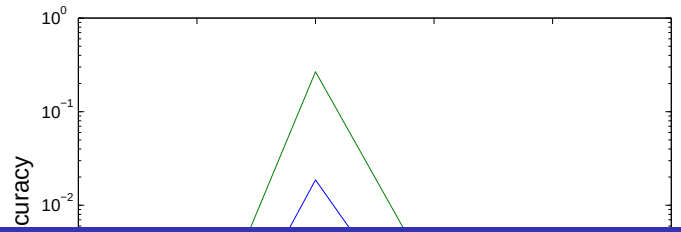
Conclusions & Future work

QR-algorithm with shift on the inverse

- QR-algorithm with shift on the matrix T and on its inverse S.
- Standard QR-algorithm (with shift) computes first the largest eigenvalues.
- This results in a lot of flops before the smallest one is computed in the tridiagonal case.

Results of QR-algorithm with shift

- The smallest eigenvalue via the semiseparable approach, is a little bit more accurate ($\lambda_1 \approx 10^{-12}$).
- But, we do loose accuracy when inverting the tridiagonal matrix. (Blue is semiseparable, Green is tridiagonal)



Outline

Introduction

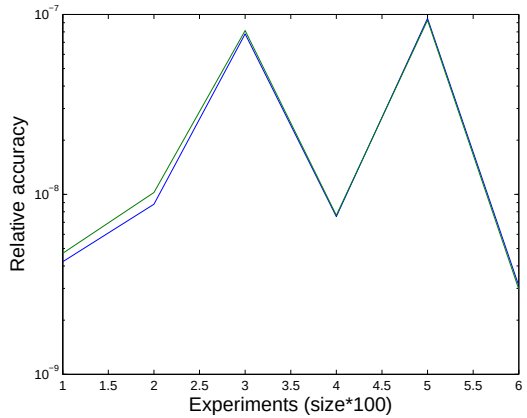
QR-iterations on the inverse
- QR-algorithm with shift
- QR-algorithm without shift

An implicit approach

Conclusions & Future work

QR-algorithm without shift

- Without shift means no forced convergence to the largest eigenvalue ($\lambda_1 \approx 10^{-12}$).
- Both approaches perform equally well.



Outline

Introduction

QR-iterations on the inverse

An implicit approach
- The explicit approach
- The implicit approach
- Combined QR-iterations without shift
- The implicit QR-algorithm of a tridiagonal
- Algoritmic idea to compute the smallest eigenvalue

Conclusions & Future work

Outline

Introduction

QR-iterations on the inverse

An implicit approach
- The explicit approach
- The implicit approach
- Combined QR-iterations without shift
- The implicit QR-algorithm of a tridiagonal
- Algoritmic idea to compute the smallest eigenvalue

Conclusions & Future work

The QR-algorithm, the explicit approach

Suppose we have a shift κ, and we calculate the QR-decomposition of the matrix $A - \kappa I$:

$$A - \kappa I = QR$$

and let us denote with $\tilde{A}$ the matrix:

$$\tilde{A} = RQ + \kappa I.$$

The matrix $\tilde{A}$ is the result of performing one step of the QR-algorithm on the matrix A.

When we perform the operations as presented here, this is called the explicit approach.

Outline

Introduction

QR-iterations on the inverse

An implicit approach
- The explicit approach
- The implicit approach
- Combined QR-iterations without shift
- The implicit QR-algorithm of a tridiagonal
- Algoritmic idea to compute the smallest eigenvalue

Conclusions & Future work

The implicit approach

We can rewrite the formulas from before as:

$$(A - \kappa I) = QR \Rightarrow Q^{-1}(A - \kappa I) = R$$

substituting R in the next equation

$$\tilde{A} = RQ + \kappa I$$

gives us

$$\tilde{A} = Q^{-1}AQ.$$

When we apply the transformations Q and Q^{-1} at the same time, we have an implicit algorithm.

Why an implicit algorithm?

- If only the eigenvalues are desired, we do not need to store Q.
- We do not lose accuracy due to the adding and subtracting of the shift matrix κI, compared to the explicit QR-algorithm with shift.

Outline

Introduction

QR-iterations on the inverse

An implicit approach
- The explicit approach
- The implicit approach
- Combined QR-iterations without shift
- The implicit QR-algorithm of a tridiagonal
- Algoritmic idea to compute the smallest eigenvalue

Conclusions & Future work

Combined QR-iterations

- Inverting causes loss in accuracy!
- Suppose we have a tridiagonal matrix T with its inverse $S = T^{-1}$.
- If S converges to the dominant eigenvalue in $\hat{S} = Q^H S Q$, we have that:

$$\begin{aligned} T &= S^{-1} \\ Q^H T Q &= Q^H S^{-1} Q = \left(Q^H S Q\right)^{-1} \\ \left[\begin{array}{c|c} \hat{T} & \hat{\epsilon} \\ \hline \hat{\epsilon}^T & \lambda_1 \end{array}\right] &= \left[\begin{array}{c|c} \hat{S} & \epsilon \\ \hline \epsilon^T & 1/\lambda_1 \end{array}\right]^{-1} \end{aligned}$$

A first attempt

- As seen before, when using the QR-algorithm with shift, we get quick convergence to the largest eigenvalue of S.
- Using now this Q to compute:

$$\hat{T} = Q^H T Q,$$

would force the matrix T to converge to its smallest eigenvalue.

- Numerical results are not satisfactory.
- Due to small rounding errors, causing the factor Q, not to be exactly equal to the factor Q for T we wanted.

Outline

Introduction

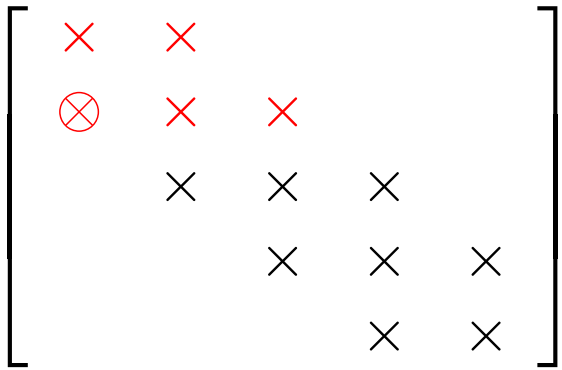
QR-iterations on the inverse

An implicit approach
- The explicit approach
- The implicit approach
- Combined QR-iterations without shift
- The implicit QR-algorithm of a tridiagonal
- Algoritmic idea to compute the smallest eigenvalue

Conclusions & Future work

The QR-factorization of tridiagonal matrices

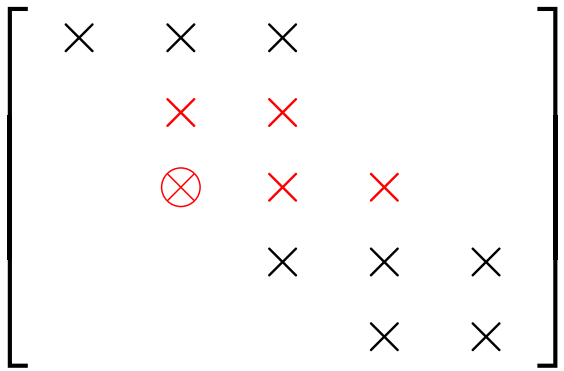
The QR-factorization of the tridiagonal matrix $A - \kappa I$, consists of $n - 1$ Givens transformations performed from the bottom to the top.



- Apply a Givens transformation on the left.
- Involving row 1 and 2.

The QR-factorization of tridiagonal matrices

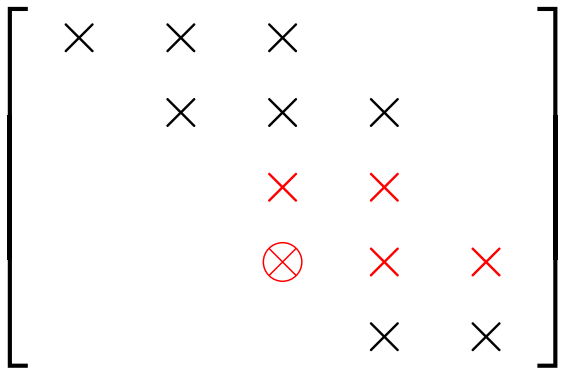
The QR-factorization of the tridiagonal matrix $A - \kappa I$, consists of $n - 1$ Givens transformations



- Apply a Givens transformation on the left.
- Involving row 2 and 3.

The QR-factorization of tridiagonal matrices

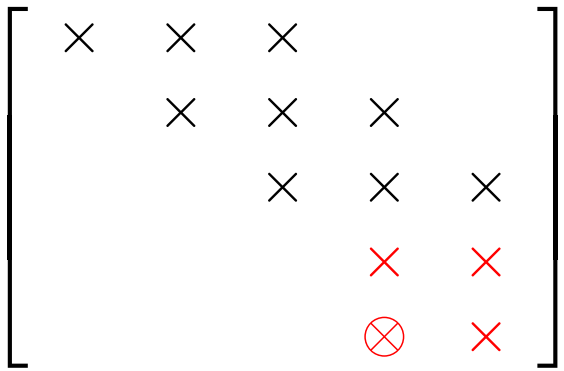
The QR-factorization of the tridiagonal matrix $A - \kappa I$, consists of $n - 1$ Givens transformations



- Apply a Givens transformation on the left.
- Involving row 3 and 4.

The QR-factorization of tridiagonal matrices

The QR-factorization of the tridiagonal matrix $A - \kappa I$, consists of $n - 1$ Givens transformations



- Apply a Givens transformation on the left.
- Involving row 4 and 5.

The QR-factorization of tridiagonal matrices

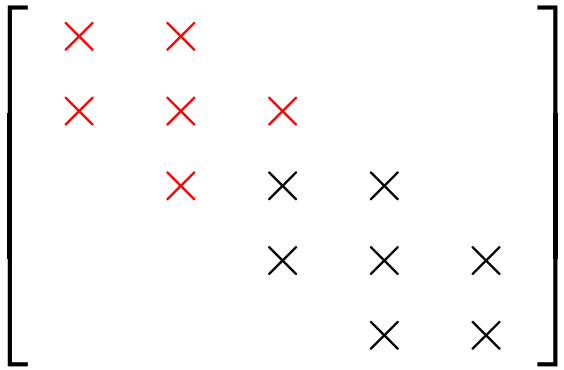
The QR-factorization of the tridiagonal matrix $A - \kappa I$, consists of $n - 1$ Givens transformations

$$\begin{bmatrix} \times & \times & \times & & \\ & \times & \times & \times & \\ & & \times & \times & \times \\ & & & \times & \times \\ & & & & \times \end{bmatrix}$$

- We have now an upper triangular matrix R.

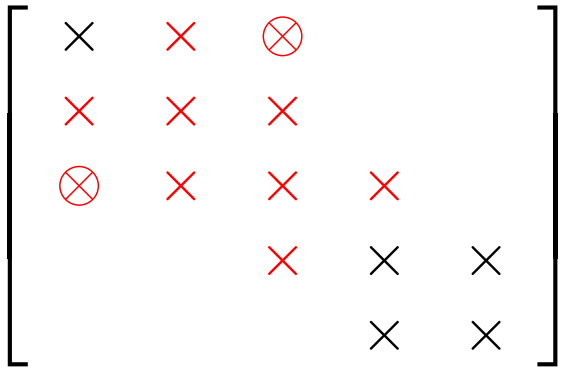
Implicit QR-algorithm for tridiagonal matrices

As mentioned before we want to apply the transformations at both sides at the same time



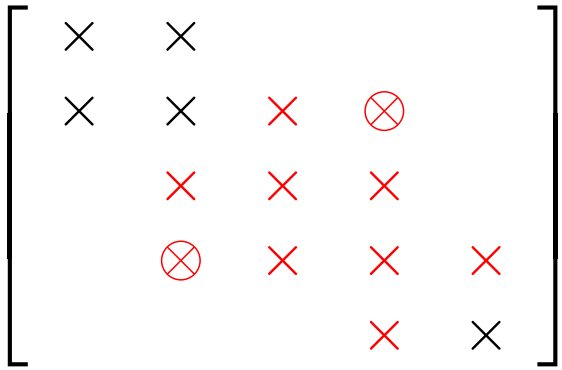
- Apply the first Givens transformation on both sides of the matrix.
- Involving row 1 and 2.
- Involving column 1 and 2.

Implicit QR-algorithm for tridiagonal matrices



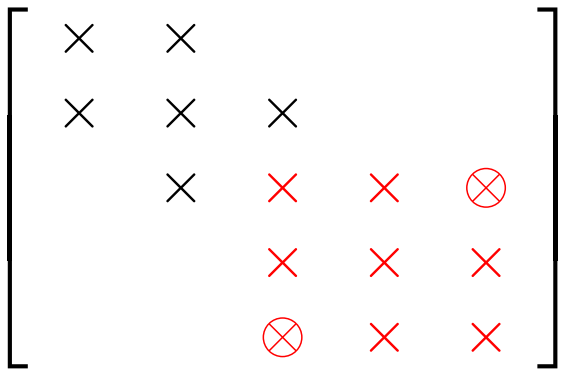
- Apply a Givens transformation to annihilate the bulge.
- Involving row 2 and 3.
- Involving column 2 and 3.

Implicit QR-algorithm for tridiagonal matrices



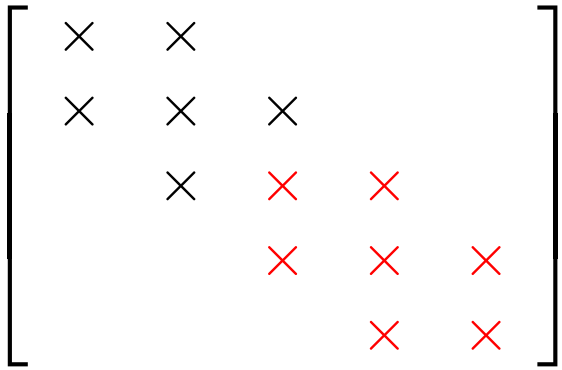
- Apply a Givens transformation to annihilate the bulge.
- Involving row 3 and 4.
- Involving column 3 and 4.

Implicit QR-algorithm for tridiagonal matrices



- Apply a Givens transformation to annihilate the bulge.
- Involving row 4 and 5.
- Involving column 4 and 5.

Implicit QR-algorithm for tridiagonal matrices



- We have again a tridiagonal matrix.

The implicit Q-theorem

- In the implicit approach, it seemed that only one Givens transformation was needed to start the process.
- So the algorithm is as follows:
  - Compute the first Givens transformation of the QR-factorization of $T - \kappa I$.
  - Apply it on both sides of the matrix T.
  - Chase the bulge.
- The implicit Q-theorem proves that this approach performs a step of the QR-algorithm on T.

Outline

Introduction

QR-iterations on the inverse

An implicit approach
- The explicit approach
- The implicit approach
- Combined QR-iterations without shift
- The implicit QR-algorithm of a tridiagonal
- Algoritmic idea to compute the smallest eigenvalue

Conclusions & Future work

The idea

- We know that if S converges to the largest via the QR-algorithm, that T converges to the smallest.

- Determine the first Givens transformation of the implicit approach for T, via the QR-factorization of S.

Algorithm 1

A first algorithm has the following structure:

1. Invert the matrix T. Denote $S = T^{-1}$.
2. Compute the initial Givens transformation for S.
3. Perform the implicit QR-step on T.
   Store this in T.
4. Perform the implicit QR-step on S.
   Store this in S.
5. Start again at step 2.

Remarks on algorithm

Remarks on this algorithm:

1. Due to small rounding errors, we get more and more divergence between the intial Givens of S and the initial Givens we need in T.
2. This can result in slow convergence.
3. Slow convergence causes loss in accuracy.

Algorithm 2

A more stable approach

1. Invert the matrix T. Denote $S = T^{-1}$.
2. Compute the initial Givens transformation for S.
3. Perform the implicit QR-step on T.
   Store this in T.
4. Start again at step 1.

The complexity is similar to the complexity of the first algorithm.

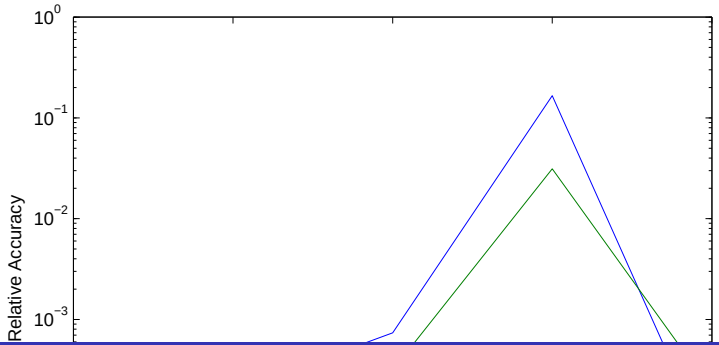
Remarks on algorithm

Remarks on this algorithm:

1. More stable due to the forced inversion of T.
2. This leads to a more accurate first Givens transformation.
3. Faster convergence.
4. Less loss in accuracy.

Remarks on algorithm

Blue is the new approach, green represents the traditional approach.



Outline

Introduction

QR-iterations on the inverse

An implicit approach

Conclusions & Future work

Conclusions

- No dramatic increase in accuracy.
- Improving convergence speed can be done via
  - More accurate inversion of the tridiagonal.
  - Using shifted QR for the semiseparable to determine the first Givens.
- Improving accuracy can be done via
  - Iterative refinement.
  - More accurate first Givens transformation in the implicit approach, i.e. more accurate inversion.