Francis’s Algorithm

David S. Watkins
watkins@math.wsu.edu

Department of Mathematics
Washington State University
Eigenvalue Problem: \[ Av = \lambda v \]
- Eigenvalue Problem: \( Av = \lambda v \)

- How to solve?
Eigenvalue Problem: \[ Av = \lambda v \]

How to solve?

\[ \lambda = \text{eig}(A) \]
Eigenvalue Problem: \( Av = \lambda v \)

How to solve?

\[ \lambda = \text{eig}(A) \]

How does \text{eig} do it?
Eigenvalue Problem: \[ Av = \lambda v \]

How to solve?

\[
\text{lambda} = \text{eig}(A)
\]

How does \text{eig} do it?

Francis’s algorithm,
Eigenvalue Problem:  \( Av = \lambda v \)

How to solve?

\[
\text{lambda} = \text{eig}(A)
\]

How does \text{eig} do it?

Francis’s algorithm, aka
Eigenvalue Problem: \[ Av = \lambda v \]

How to solve?

\[
\text{lambda} = \text{eig}(A)
\]

How does \text{eig} do it?

Francis’s algorithm, aka
the implicitly shifted \(QR\) algorithm
Eigenvalue Problem: \( Av = \lambda v \)

How to solve?

\[
\lambda = \text{eig}(A)
\]

How does \text{eig} do it?

Francis’s algorithm, aka the implicitly shifted \( QR \) algorithm

50 years!
Eigenvalue Problem: \[ A \mathbf{v} = \lambda \mathbf{v} \]

How to solve?

\[ \lambda = \text{eig}(A) \]

How does \text{eig} do it?

Francis’s algorithm, aka

the implicitly shifted \( QR \) algorithm

50 years!

Top Ten of the century (Dongarra and Sullivan)
John Francis
Who is John Francis?
Who is John Francis?

- born near London in 1934
Who is John Francis?

- born near London in 1934
- employed in late 50’s, Pegasus computer
Who is John Francis?

- born near London in 1934
- employed in late 50’s, Pegasus computer
- linear algebra, eigenvalue routines
Who is John Francis?

- born near London in 1934
- employed in late 50’s, Pegasus computer
- linear algebra, eigenvalue routines
- primitive computer
Who is John Francis?

- born near London in 1934
- employed in late 50’s, Pegasus computer
- linear algebra, eigenvalue routines
- primitive computer
- no software
Who is John Francis?

- born near London in 1934
- employed in late 50’s, Pegasus computer
- linear algebra, eigenvalue routines
- primitive computer
- no software
- experimented with a variety of methods
Who is John Francis?

- born near London in 1934
- employed in late 50’s, Pegasus computer
- linear algebra, eigenvalue routines
- primitive computer
- no software
- experimented with a variety of methods
- invented His algorithm and programmed it
Who is John Francis?

- born near London in 1934
- employed in late 50’s, Pegasus computer
- linear algebra, eigenvalue routines
- primitive computer
- no software
- experimented with a variety of methods
- invented His algorithm and programmed it
- moved on to other things
Some History
Some History

- Rutishauser (q-d 1954, LR 1958)
Some History

- Rutishauser (q-d 1954, LR 1958)
- Francis’s first paper (QR)
Some History

- Rutishauser (q-d 1954, LR 1958)
- Francis’s first paper (QR)
  \[ A - \rho I = QR, \quad RQ + \rho I = \hat{A} \]
Some History

- Rutishauser (q-d 1954, LR 1958)
- Francis’s first paper (QR)
  - $A - \rho I = QR$, $RQ + \rho I = \hat{A}$ repeat!
Some History

- Rutishauser (q-d 1954, LR 1958)
- Francis’s first paper (QR)
  - $A - \rho I = QR$, $RQ + \rho I = \hat{A}$ repeat!
- Kublanovskaya
Some History

- Rutishauser (q-d 1954, LR 1958)
- Francis’s first paper (QR)
  - $A - \rho I = QR$, $RQ + \rho I = \hat{A}$ repeat!
- Kublanovskaya
- …but this is not “Francis’s Algorithm”
Francis’s Algorithm
Francis’s Algorithm

- Second paper of Francis
Francis’s Algorithm

- Second paper of Francis
- real matrices
Francis’s Algorithm

- Second paper of Francis
- real matrices with complex pairs of eigenvalues
Francis’s Algorithm

- Second paper of Francis
- real matrices with complex pairs of eigenvalues
- complex shifts
Francis’s Algorithm

- Second paper of Francis
- real matrices with complex pairs of eigenvalues
- complex shifts
- want to stay in real arithmetic
Francis’s Algorithm

- Second paper of Francis
- real matrices with complex pairs of eigenvalues
- complex shifts
- want to stay in real arithmetic
- two steps at once
Francis’s Algorithm

- Second paper of Francis
- real matrices with complex pairs of eigenvalues
- complex shifts
- want to stay in real arithmetic
- two steps at once
- double-shift $QR$ algorithm
Francis’s Algorithm

- Second paper of Francis
- real matrices with complex pairs of eigenvalues
- complex shifts
- want to stay in real arithmetic
- two steps at once
- double-shift $QR$ algorithm
- radically different from basic QR
Francis’s Algorithm

- Second paper of Francis
- Real matrices with complex pairs of eigenvalues
- Complex shifts
- Want to stay in real arithmetic
- Two steps at once
- Double-shift QR algorithm
- Radically different from basic QR
- Usual justification: Francis’s implicit-Q theorem
Francis’s Algorithm
Francis’s Algorithm

- upper Hessenberg form
Francis’s Algorithm

- upper Hessenberg form
- pick some shifts $\rho_1, \ldots, \rho_m$ $(m = 1, 2, 4, 6)$
Francis’s Algorithm

- upper Hessenberg form
- pick some shifts $\rho_1, \ldots, \rho_m$ ($m = 1, 2, 4, 6$)
- $p(A) = (A - \rho_1 I) \cdots (A - \rho_m I)$
Francis’s Algorithm

- upper Hessenberg form
- pick some shifts $\rho_1, \ldots, \rho_m$ \hspace{1cm} ($m = 1, 2, 4, 6$)
- $p(A) = (A - \rho_1 I) \cdots (A - \rho_m I)$ \hspace{1cm} expensive!
Francis’s Algorithm

- upper Hessenberg form
- pick some shifts $\rho_1, \ldots, \rho_m$ ($m = 1, 2, 4, 6$)
- $p(A) = (A - \rho_1 I) \cdots (A - \rho_m I)$ expensive!
- compute $p(A)e_1$
Francis’s Algorithm

- upper Hessenberg form
- pick some shifts $\rho_1, \ldots, \rho_m$  ($m = 1, 2, 4, 6$)
- $p(A) = (A - \rho_1 I) \cdots (A - \rho_m I)$ expensive!
- compute $p(A)e_1$ cheap!
Francis’s Algorithm

- upper Hessenberg form
- pick some shifts $\rho_1, \ldots, \rho_m$ ($m = 1, 2, 4, 6$)
- $p(A) = (A - \rho_1 I) \cdots (A - \rho_m I)$ expensive!
- compute $p(A)e_1$ cheap!
- Build unitary $Q_0$ with $q_1 = \alpha p(A)e_1$. 
Francis’s Algorithm

- upper Hessenberg form
- pick some shifts $\rho_1, \ldots, \rho_m$ ($m = 1, 2, 4, 6$)
- $p(A) = (A - \rho_1 I) \cdots (A - \rho_m I)$ expensive!
- compute $p(A)e_1$ cheap!
- Build unitary $Q_0$ with $q_1 = \alpha p(A)e_1$.
- Perform similarity transform $A \rightarrow Q_0^{-1}AQ_0$. 
Francis’s Algorithm

- upper Hessenberg form
- pick some shifts $\rho_1, \ldots, \rho_m$ ($m = 1, 2, 4, 6$)
- $p(A) = (A - \rho_1 I) \cdots (A - \rho_m I)$ expensive!
- compute $p(A)e_1$ cheap!
- Build unitary $Q_0$ with $q_1 = \alpha p(A)e_1$.
- Perform similarity transform $A \rightarrow Q_0^{-1}AQ_0$.
- Hessenberg form is disturbed.
An Upper Hessenberg Matrix
After the Transformation \((Q_0^{-1}AQ_0)\)
After the Transformation \((Q_0^{-1}AQ_0)\)

Now return the matrix to Hessenberg form.
Chasing the Bulge
Chasing the Bulge
Done
Done

The Francis iteration is complete!
Summary of Francis Iteration
Summary of Francis Iteration

- Pick some shifts.
Summary of Francis Iteration

- Pick some shifts.
- Compute $p(A)e_1$. ($p$ determined by shifts)
Summary of Francis Iteration

- Pick some shifts.
- Compute $p(A) e_1$. ($p$ determined by shifts)
- Build $Q_0$ with first column $q_1 = \alpha p(A) e_1$. 
Summary of Francis Iteration

- Pick some shifts.
- Compute $p(A)e_1$.  ($p$ determined by shifts)
- Build $Q_0$ with first column $q_1 = \alpha p(A)e_1$.
- Make a bulge.  ($A \rightarrow Q_0^{-1}AQ_0$)
Summary of Francis Iteration

- Pick some shifts.
- Compute $p(A)e_1$.  ($p$ determined by shifts)
- Build $Q_0$ with first column $q_1 = \alpha p(A)e_1$.
- Make a bulge.  ($A \rightarrow Q_0^{-1}AQ_0$)
- Chase the bulge.  (return to Hessenberg form)
Summary of Francis Iteration

- Pick some shifts.
- Compute $p(A)e_1$. ($p$ determined by shifts)
- Build $Q_0$ with first column $q_1 = \alpha p(A)e_1$.
- Make a bulge. ($A \rightarrow Q_0^{-1}AQ_0$)
- Chase the bulge. (return to Hessenberg form)
- $\hat{A} = Q^{-1}AQ$
Quicker Summary
Quicker Summary

- Make a bulge.
Quicker Summary

- Make a bulge.
- Chase it.
Remarks

- This is pretty simple.
Remarks

- This is pretty simple.
- no $QR$ decomposition in sight!
Remarks

- This is pretty simple.
- no $QR$ decomposition in sight!
- Why call it the $QR$ algorithm?
Remarks

- This is pretty simple.
- no $QR$ decomposition in sight!
- Why call it the $QR$ algorithm?
- Confusion!
Remarks

- This is pretty simple.
- no $QR$ decomposition in sight!
- Why call it the $QR$ algorithm?
- Confusion!
- Can we think of another name?
Remarks

- This is pretty simple.
- no $QR$ decomposition in sight!
- Why call it the $QR$ algorithm?
- Confusion!
- Can we think of another name?
- I’m calling it Francis’s Algorithm.
Remarks

- This is pretty simple.
- no QR decomposition in sight!
- Why call it the QR algorithm?
- Confusion!
- Can we think of another name?
- I’m calling it Francis’s Algorithm.
- This is not a radical move.
Question
Question

- How should we view Francis’s algorithm?
Question

- How should we view Francis’s algorithm?
- Do we have to start with the basic $QR$ algorithm?
Question

- How should we view Francis’s algorithm?
- Do we have to start with the basic $QR$ algorithm?
- Couldn’t we just as well introduce Francis’s algorithm directly?
Question

- How should we view Francis’s algorithm?
- Do we have to start with the basic $QR$ algorithm?
- Couldn’t we just as well introduce Francis’s algorithm directly? …bypassing the basic $QR$ algorithm entirely?
Question

- How should we view Francis’s algorithm?
- Do we have to start with the basic $QR$ algorithm?
- Couldn’t we just as well introduce Francis’s algorithm directly? . . . \textit{bypassing the basic} $QR$ algorithm \textit{entirely}?
- . . . and the answer is:
Question

- How should we view Francis’s algorithm?
- Do we have to start with the basic $QR$ algorithm?
- Couldn’t we just as well introduce Francis’s algorithm directly? ...bypassing the basic $QR$ algorithm entirely?
- ...and the answer is: Why not?
Question

- How should we view Francis’s algorithm?
- Do we have to start with the basic $QR$ algorithm?
- Couldn’t we just as well introduce Francis’s algorithm directly? ...bypassing the basic $QR$ algorithm entirely?
- ...and the answer is: Why not?
- This simplifies the presentation.
Question

- How should we view Francis’s algorithm?
- Do we have to start with the basic $QR$ algorithm?
- Couldn’t we just as well introduce Francis’s algorithm directly? …bypassing the basic $QR$ algorithm entirely?
- …and the answer is: Why not?
- This simplifies the presentation.
- I’m putting my money where my mouth is.
I’m putting my money where my mouth is ...
I’m putting my money where my mouth is …
… and saving one entire section!
Pedagogical Pathway
Pedagogical Pathway

- reduction to Hessenberg form
Pedagogical Pathway

- reduction to Hessenberg form
- Francis’s algorithm
Pedagogical Pathway

- reduction to Hessenberg form
- Francis’s algorithm
- Try it out!
Pedagogical Pathway

- reduction to Hessenberg form
- Francis’s algorithm
- Try it out!
- It works great!
Pedagogical Pathway

- reduction to Hessenberg form
- Francis’s algorithm
- Try it out!
- It works great!
- Why does it work?
Ingredients of Francis’s Algorithm
Ingredients of Francis’s Algorithm

- subspace iteration  (power method)
Ingredients of Francis’s Algorithm

- subspace iteration (power method)
- subspace iteration with changes of coordinate system
Ingredients of Francis’s Algorithm

- subspace iteration (power method)
- subspace iteration with changes of coordinate system
- Krylov subspaces
Ingredients of Francis’s Algorithm

- subspace iteration (power method)
- subspace iteration with changes of coordinate system
- Krylov subspaces (instead of the implicit-\(Q\) theorem)
Ingredients of Francis’s Algorithm

- subspace iteration (power method)
- subspace iteration with changes of coordinate system
- Krylov subspaces (instead of the implicit-$Q$ theorem)
- Krylov subspaces and subspace iteration
Ingredients of Francis’s Algorithm

- subspace iteration  (power method)
- subspace iteration with changes of coordinate system
- **Krylov subspaces**  (instead of the implicit-$Q$ theorem)
- Krylov subspaces and subspace iteration
- Krylov subspaces and Hessenberg form
Power Method, Subspace Iteration
Power Method, Subspace Iteration

- $v, Av, A^2v, A^3v, \ldots$
Power Method, Subspace Iteration

- $v, Av, A^2v, A^3v, \ldots$
- convergence rate $|\lambda_2/\lambda_1|$
Power Method, Subspace Iteration

- $v, Av, A^2v, A^3v, \ldots$
- convergence rate $|\lambda_2/\lambda_1|$  
- $S, AS, A^2S, A^3S, \ldots$
Power Method, Subspace Iteration

- $v, Av, A^2v, A^3v, \ldots$
- convergence rate $|\lambda_2/\lambda_1|$
- $S, AS, A^2S, A^3S, \ldots$
- subspaces of dimension $j$
Power Method, Subspace Iteration

- $v, Av, A^2v, A^3v, \ldots$
- convergence rate $|\lambda_2/\lambda_1|$
- $S, AS, A^2S, A^3S, \ldots$
- subspaces of dimension $j$ ($|\lambda_{j+1}/\lambda_j|$)
Power Method, Subspace Iteration

- $v$, $Av$, $A^2v$, $A^3v$, ...
- convergence rate $|\lambda_2/\lambda_1|
- S$, $AS$, $A^2S$, $A^3S$, ...
- subspaces of dimension $j$ ($|\lambda_{j+1}/\lambda_j|$)
- Substitute $p(A)$ for $A$
Power Method, Subspace Iteration

- $v, Av, A^2v, A^3v, \ldots$
- convergence rate $|\lambda_2/\lambda_1|$
- $S, AS, A^2S, A^3S, \ldots$
- subspaces of dimension $j$ ($|\lambda_{j+1}/\lambda_j|$)
- Substitute $p(A)$ for $A$ (shifts, multiple steps)
Power Method, Subspace Iteration

- \( v, Av, A^2v, A^3v, \ldots \)
- convergence rate \( |\lambda_2/\lambda_1| \)
- \( S, AS, A^2S, A^3S, \ldots \)
- subspaces of dimension \( j \) \( (|\lambda_{j+1}/\lambda_j|) \)
- Substitute \( p(A) \) for \( A \)  (shifts, multiple steps)
- \( S, p(A)S, p(A)^2S, p(A)^3S, \ldots \)
Power Method, Subspace Iteration

- $v, Av, A^2v, A^3v, \ldots$
- convergence rate $|\lambda_2/\lambda_1|$
- $S, AS, A^2S, A^3S, \ldots$
- subspaces of dimension $j$ $|\lambda_{j+1}/\lambda_j|$
- Substitute $p(A)$ for $A$ (shifts, multiple steps)
- $S, p(A)S, p(A)^2S, p(A)^3S, \ldots$
- convergence rate $|p(\lambda_{j+1})/p(\lambda_j)|$
Subspace Iteration with changes of coordinate system
Subspace Iteration with changes of coordinate system

- take $S = \text{span}\{e_1, \ldots, e_j\}$
Subspace Iteration
with changes of coordinate system

- take $S = \text{span}\{e_1, \ldots, e_j\}$

$$p(A)S = \text{span}\{p(A)e_1, \ldots, p(A)e_j\}$$

$$= \text{span}\{q_1, \ldots, q_j\} \quad \text{(orthonormal)}$$
Subspace Iteration
with changes of coordinate system

- take $S = \text{span}\{e_1, \ldots, e_j\}$

  $p(A)S = \text{span}\{p(A)e_1, \ldots, p(A)e_j\}$

  $= \text{span}\{q_1, \ldots, q_j\}$ (orthonormal)

- build unitary $Q = [q_1 \cdots q_j \cdots]$
Subspace Iteration with changes of coordinate system

- take $S = \text{span}\{e_1, \ldots, e_j\}$

  $$p(A)S = \text{span}\{p(A)e_1, \ldots, p(A)e_j\}$$

  $$= \text{span}\{q_1, \ldots, q_j\} \text{ (orthonormal)}$$

- build unitary $Q = [q_1 \cdots q_j \cdots]$

- change coordinate system: $\hat{A} = Q^{-1}AQ$
Subspace Iteration with changes of coordinate system

- take $S = \text{span}\{e_1, \ldots, e_j\}$

  $p(A)S = \text{span}\{p(A)e_1, \ldots, p(A)e_j\}$
  
  $= \text{span}\{q_1, \ldots, q_j\}$ (orthonormal)

- build unitary $Q = [q_1 \cdots q_j \cdots]$

- change coordinate system: $\hat{A} = Q^{-1}AQ$

- $q_k \rightarrow Q^{-1}q_k = Q^*q_k = e_k$
Subspace Iteration with changes of coordinate system

- take $S = \text{span}\{e_1, \ldots, e_j\}$

  \[ p(A)S = \text{span}\{p(A)e_1, \ldots, p(A)e_j\} = \text{span}\{q_1, \ldots, q_j\} \text{ (orthonormal)} \]

- build unitary $Q = [q_1 \cdots q_j \cdots]$ 

- change coordinate system: $\hat{A} = Q^{-1}AQ$

- $q_k \to Q^{-1}q_k = Q^*q_k = e_k$

- $\text{span}\{q_1, \ldots, q_j\} \to \text{span}\{e_1, \ldots, e_j\}$
Subspace Iteration with changes of coordinate system

- take $S = \text{span}\{e_1, \ldots, e_j\}$

  \[ p(A)S = \text{span}\{p(A)e_1, \ldots, p(A)e_j\} \]

  \[ = \text{span}\{q_1, \ldots, q_j\} \text{ (orthonormal)} \]

- build unitary $Q = [q_1 \cdots q_j \cdots]$

- change coordinate system: $\hat{A} = Q^{-1}AQ$

- $q_k \rightarrow Q^{-1}q_k = Q^{*}q_k = e_k$

- $\text{span}\{q_1, \ldots, q_j\} \rightarrow \text{span}\{e_1, \ldots, e_j\}$

- ready for next iteration
This version of subspace iteration . . .
This version of subspace iteration . . .

- . . . holds the subspace fixed
This version of subspace iteration . . .

- . . . holds the subspace fixed
- while the matrix changes.
This version of subspace iteration . . .

- ... holds the subspace fixed
- while the matrix changes.
- ... moving toward a matrix under which

\[ \text{span}\{e_1, \ldots, e_j\} \]

is invariant.
This version of subspace iteration . . .

- ... holds the subspace fixed
- while the matrix changes.
- ... moving toward a matrix under which

\[ \text{span}\{e_1, \ldots, e_j\} \]

is invariant.

- \[ A \rightarrow \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix} \] (\(A_{11}\) is \(j \times j\).)
Application to Francis’s Iteration (first pass)
Application to Francis’s Iteration (first pass)

\[ \hat{A} = Q^{-1} AQ \quad \text{where} \quad q_1 = \alpha p(A)e_1. \]
Application to Francis’s Iteration (first pass)

\[ \hat{A} = Q^{-1}AQ \quad \text{where} \quad q_1 = \alpha p(A)e_1. \]

- power method
Application to Francis’s Iteration (first pass)

\[ \hat{A} = Q^{-1}AQ \quad \text{where} \quad q_1 = \alpha p(A)e_1. \]

- power method + change of coordinates
Application to Francis’s Iteration (first pass)

\[ \hat{A} = Q^{-1} A Q \quad \text{where} \quad q_1 = \alpha p(A)e_1. \]

- power method + change of coordinates
- \( q_1 \rightarrow Q^{-1}q_1 = e_1 \)
Application to Francis’s Iteration (first pass)

\[ \hat{A} = Q^{-1}AQ \quad \text{where} \quad q_1 = \alpha p(A)e_1. \]

- power method + change of coordinates
- \( q_1 \rightarrow Q^{-1}q_1 = e_1 \)
- case \( j = 1 \) of subspace iteration with a change of coordinate system
Application to Francis’s Iteration (first pass)

\[ \hat{A} = Q^{-1}AQ \quad \text{where} \quad q_1 = \alpha p(A)e_1. \]

- power method + change of coordinates
- \( q_1 \rightarrow Q^{-1}q_1 = e_1 \)
- case \( j = 1 \) of subspace iteration with a change of coordinate system
- … but this is just a small part of the story.
Krylov Subspaces ...
Krylov Subspaces ...
... and Subspace Iteration
Krylov Subspaces . . .

. . . and Subspace Iteration

Def: \( \mathcal{K}_j(A, q) = \text{span}\{q, Aq, A^2q, \ldots, A^{j-1}q\} \)
Krylov Subspaces . . .

. . . and Subspace Iteration

Def: $\mathcal{K}_j(A, q) = \text{span}\{q, Aq, A^2q, \ldots, A^{j-1}q\}$

$j = 1, 2, 3, \ldots$ (nested subspaces)
Krylov Subspaces ...
... and Subspace Iteration

- Def: $\mathcal{K}_j(A, q) = \text{span}\{q, Aq, A^2q, \ldots, A^{j-1}q\}$
  
  $j = 1, 2, 3, \ldots$ (nested subspaces)

- $\mathcal{K}_j(A, q)$ are “determined by $q$”.

Francis’s Algorithm – p. 2
Krylov Subspaces ... 
... and Subspace Iteration

- Def: $ \mathcal{K}_j(A, q) = \text{span}\{q, Aq, A^2q, \ldots, A^{j-1}q\}$
  
  $j = 1, 2, 3, \ldots$ (nested subspaces)

- $\mathcal{K}_j(A, q)$ are “determined by $q$”.

- $p(A)\mathcal{K}_j(A, q) = \mathcal{K}_j(A, p(A)q)$
Krylov Subspaces . . .

...and Subspace Iteration

■ Def: \( \mathcal{K}_j(A, q) = \text{span}\{q, Aq, A^2q, \ldots, A^{j-1}q\} \)
  
  \( j = 1, 2, 3, \ldots \) (nested subspaces)

■ \( \mathcal{K}_j(A, q) \) are “determined by \( q \)”.

■ \( p(A)\mathcal{K}_j(A, q) = \mathcal{K}_j(A, p(A)q) \)

■ . . . because \( p(A)A = Ap(A) \)
Krylov Subspaces . . .
...and Subspace Iteration

- Def: $\mathcal{K}_j(A, q) = \text{span}\{q, Aq, A^2q, \ldots, A^{j-1}q\}$
  
  $j = 1, 2, 3, \ldots$ (nested subspaces)

- $\mathcal{K}_j(A, q)$ are “determined by $q$”.

- $p(A)\mathcal{K}_j(A, q) = \mathcal{K}_j(A, p(A)q)$

- ...because $p(A)A = Ap(A)$

- Conclusion: Power method induces nested subspace iterations on Krylov subspaces.
power method: \[ q \rightarrow p(A)^k q \]
- power method: \( q \rightarrow p(A)^k q \)
- nested subspace iterations:

\[
p(A)^k K_j(A, q) = K_j(A, p(A)^k q) \quad j = 1, 2, 3, \ldots
\]
- power method: \( q \rightarrow p(A)^k q \)
- nested subspace iterations:
  \[
p(A)^k \mathcal{K}_j(A, q) = \mathcal{K}_j(A, p(A)^k q) \quad j = 1, 2, 3, \ldots
\]
- convergence rates:
  \[
  \left| p(\lambda_{j+1})/p(\lambda_j) \right|, \quad j = 1, 2, 3, \ldots, n - 1
  \]
Krylov Subspaces ...
Krylov Subspaces . . .
...and Hessenberg matrices . . .
Krylov Subspaces ... 
...and Hessenberg matrices ...

... go hand in hand.
Krylov Subspaces . . .
... and Hessenberg matrices . . .

- ... go hand in hand.
- $A$ properly upper Hessenberg $\Rightarrow$

$$K_j(A, e_1) = \text{span}\{e_1, \ldots, e_j\}.$$
Krylov Subspaces . . .
...and Hessenberg matrices . . .

- ... go hand in hand.
- A properly upper Hessenberg \( \Rightarrow \)

\[ \mathcal{K}_j(A, e_1) = \text{span}\{e_1, \ldots, e_j\}. \]

- More generally . . .
Krylov-Hessenberg Relationship
Krylov-Hessenberg Relationship

If $\hat{A} = Q^{-1}AQ$, 
Krylov-Hessenberg Relationship

- If $\hat{A} = Q^{-1}AQ$, 
- and $\hat{A}$ is properly upper Hessenberg,
Krylov-Hessenberg Relationship

- If $\hat{A} = Q^{-1}AQ$, 
- and $\hat{A}$ is properly upper Hessenberg, 
- then for $j = 1, 2, 3, \ldots$, 

Francis's Algorithm – p. 2
Krylov-Hessenberg Relationship

- If \( \hat{A} = Q^{-1}AQ \),
- and \( \hat{A} \) is properly upper Hessenberg,
- then for \( j = 1, 2, 3, \ldots \),

\[
\text{span}\{q_1, \ldots, q_j\} = \mathcal{K}_j(A, q_1).
\]
Application to Francis’s Iteration
Application to Francis’s Iteration

\[ \hat{A} = Q^{-1}AQ \quad \text{where} \quad q_1 = \alpha p(A)e_1. \]
Application to Francis’s Iteration

\[ \hat{A} = Q^{-1}AQ \quad \text{where} \quad q_1 = \alpha p(A)e_1. \]

- power method with a change of coordinate system.
Application to Francis’s Iteration

\[ \hat{A} = Q^{-1}AQ \quad \text{where} \quad q_1 = \alpha p(A)e_1. \]

- power method with a change of coordinate system. Moreover …
Application to Francis’s Iteration

\[ \hat{A} = Q^{-1}AQ \quad \text{where} \quad q_1 = \alpha p(A)e_1. \]

- power method with a change of coordinate system. Moreover ...

- \( p(A)K_j(A, e_1) = K_j(A, p(A)e_1) \)
Application to Francis’s Iteration

\[ \hat{A} = Q^{-1} A Q \quad \text{where} \quad q_1 = \alpha p(A)e_1. \]

- power method with a change of coordinate system. Moreover ...
- \( p(A)K_j(A, e_1) = K_j(A, p(A)e_1) \)
- i.e. \( p(A)\text{span}\{e_1, \ldots, e_j\} = \text{span}\{q_1, \ldots, q_j\} \)
Application to Francis’s Iteration

\[ \hat{A} = Q^{-1}AQ \quad \text{where} \quad q_1 = \alpha p(A)e_1. \]

- power method with a change of coordinate system. Moreover . . .
- \[ p(A)K_j(A, e_1) = K_j(A, p(A)e_1) \]
- i.e. \[ p(A)\text{span}\{e_1, \ldots, e_j\} = \text{span}\{q_1, \ldots, q_j\} \]
- subspace iteration with a change of coordinate system
Application to Francis’s Iteration

\[ \hat{A} = Q^{-1} A Q \quad \text{where} \quad q_1 = \alpha p(A)e_1. \]

- power method with a change of coordinate system. Moreover …
- \( p(A)K_j(A, e_1) = K_j(A, p(A)e_1) \)
- i.e. \( p(A)\text{span}\{e_1, \ldots , e_j\} = \text{span}\{q_1, \ldots , q_j\} \)
- subspace iteration with a change of coordinate system for \( j = 1, 2, 3, \ldots , n - 1 \)
Application to Francis’s Iteration

\[ \hat{A} = Q^{-1}AQ \quad \text{where} \quad q_1 = \alpha p(A)e_1. \]

- power method with a change of coordinate system. Moreover ...
- \[ p(A)K_j(A, e_1) = K_j(A, p(A)e_1) \]
- i.e. \[ p(A)\text{span}\{e_1, \ldots, e_j\} = \text{span}\{q_1, \ldots, q_j\} \]
- subspace iteration with a change of coordinate system for \( j = 1, 2, 3, \ldots, n - 1 \)
- \[ |p(\lambda_{j+1})/p(\lambda_j)| \quad j = 1, 2, 3, \ldots, n - 1 \]
Details

- choice of shifts
Details

- choice of shifts
- We change the shifts at each step.
Details

- choice of shifts
- We change the shifts at each step.
- \(\Rightarrow\) quadratic or cubic convergence
Details

- choice of shifts
- We change the shifts at each step.
- \( \Rightarrow \) quadratic or cubic convergence
Details

- choice of shifts
- We change the shifts at each step.
- \( \Rightarrow \) quadratic or cubic convergence
Where is John Francis?
Where is John Francis?

- question asked frequently by Gene Golub
Where is John Francis?

- question asked frequently by Gene Golub
- inquiries by Golub and Uhlig
Where is John Francis?

- question asked frequently by Gene Golub
- inquiries by Golub and Uhlig
- Francis is alive and well, retired in the South of England.
Where is John Francis?

- question asked frequently by Gene Golub
- inquiries by Golub and Uhlig
- Francis is alive and well, retired in the South of England.
- was unaware of the impact of his algorithm
Where is John Francis?

- question asked frequently by Gene Golub
- inquiries by Golub and Uhlig
- Francis is alive and well, retired in the South of England.
- was unaware of the impact of his algorithm
- appearance at the Biennial Numerical Analysis Conference in Glasgow in June of 2009
John Francis speaking in Glasgow
A Portion of the Audience
Afterwards

Photos courtesy of Frank Uhlig