基于Francis QR算法的实Schur分解

李维杰

2024 年 11 月 14 日

目录

1 算法

实现实Schur分解的代码框架如下:

Algorithm 1 Real Schur Decomposition

function QR_ITERATION(M)
  $r \leftarrow n, l \leftarrow 1$
  $[A, Q] \leftarrow$ Householder_Hessenberg(M) ▷ Compute the Hessenberg form of M and orthogonal matrix Q
  while $r > l$ **do**
    Eliminate small off-diagonal elements
    $[isreal, eig_1, eig_2] \leftarrow$ eigen_shift($A(r-1:r, r-1:r)$) ▷ Compute the eigenvalues of the current submatrix
    if $isreal = 1$ **then**
      Choose the smaller eigenvalue min_eig
      $[A, Q] \leftarrow$ single_shift_iteration(A, l, r, min_eig, Q) ▷ Perform a single shift iteration
    else
      if $r > l + 1$ **then**
        $[A, Q] \leftarrow$ double_shift_iteration($A, l, r, eig_1 + eig_2, eig_1 \times eig_2, Q$) ▷ Perform a double shift iteration
      else
        Break ▷ Exit the loop if $r \leq l + 1$
      end if
    end if
  end while
  return A, Q ▷ Return the computed matrix A and orthogonal matrix Q
end function

具体来说,它由以下两步算法组成:

1.1 基于Householder变换的矩阵上Hessenberg化

对于任意给定的实矩阵$M \in \mathbb{R}^{n\times n}$, 我们采用Householder变换,第$k$步变换$HAH^T$将矩阵$A$的第$(k-1)$列的第$(k+1)$行及以下元素置零,从而将$A$阵变换为一个上Hessenberg阵. 这一部分算法的伪代码见附录A.

1.2 上Hessenberg矩阵的实Schur分解

对于上Hessenberg化的矩阵$A \in \mathbb{R}^{n\times n}$,将其分块为

$$A = \begin{bmatrix} T_{11} & T_{12} & T_{13} \\ & T_{22} & T_{23} \\ & & T_{33} \end{bmatrix},$$

其中T_{11}, T_{33}均是已完成收敛的拟上三角阵.这里T_{11}收敛为拟上三角阵是因为模数远大于其它特征值的特征值会提前收敛于左上角,T_{33}收敛为拟上三角阵是因为带位移的QR算法会不断把接近位移μ的特征值收敛于右下角. 约定本节之后所提到的A均指代$T_{22} \in \mathbb{R}^{(r-l+1)\times(r-l+1)}$.计算该矩阵右下角$2\times 2$子矩阵的特征值$\mu_1, \mu_2$.

(1) 当$\mu_1, \mu_2 \in \mathbb{R}$时,采用Wilkinson隐式位移.

取μ_1与μ_2中距离$A(r,r)$最近的一项作为位移量.在变换过程中,采用Givens变换逐一把下次对角线下方的bulge排出,从而完成一步迭代过程.特别地,由于隐式Q定理,第一步Givens变换矩阵G_1中的c, s应被确定为

$$\begin{cases} c = \frac{A(1,1)-\mu}{\sqrt{(A(1,1)-\mu)^2+A(2,1)^2}} \\ s = \frac{A(2,1)}{\sqrt{(A(1,1)-\mu)^2+A(2,1)^2}} \end{cases} \tag{1}$$

这一部分算法的伪代码见附录B.

(2) 当$\mu_1, \mu_2 \in \mathbb{C}$时,采用Francis隐式双位移.

使共轭复数λ_1与λ_2交替对矩阵A做位移.在变换过程中,采用Householder变换逐一把下次对角线下方的2×2bulge排出,从而完成一步迭代过程(最后一步由于bulge只剩一个元素,故可以使用Givens变换). 特别地,由于隐式Q定理,第一步Householder变换矩阵应当将e_1镜射为向量x,其中x是向量$col_1(A^2 - 2\Re(\mu)A + |\mu|^2 I)$的单位化结果.这一部分的伪代码见附录C.

2 实现细节

1. Francis QR算法可以被作用于任意实矩阵,算法过程中所有的计算均在实数域内进行.

2. 算法中的所有Householder变换均由$(I - 2\omega\omega^T)A = A - 2\omega(\omega^T A)$进行计算,从而绕过了求解Householder矩阵H的步骤.在这一优化下,所有Householder变换的效率均为$O(n^2)$.

3. 为了进一步减小误差,在求位移时若发现右下角2×2子矩阵的特征值为复数,则直接返回共轭特征值的和(即$\mathrm{tr}(A(n-1:n, n-1:n))$)以及特征值的模长的平方(即$\det(A(n-1:n, n-1:n))$).

4. 矩阵达到局部收敛的判断标准是:

$$A(i+1,i) < \mathbf{u}(|A(i+1,i+1)| + |(A(i,i))|),$$

在满足这一条件时,置$A(i+1,i) = 0$, 并缩小参与迭代的子矩阵规模.

3 数值表现

在对程序进行数值测试时,设定初始矩阵$M \in \mathbb{R}^{n\times n}$. 其中, M中的各元素在$(0,1)$中随机生成. 在3.1至3.3小节中, 让M的规模n从50开始增长到500,对期间每一规模下的矩阵M进行Francis QR算法的数值测试;

3.1 运行效率

Francis QR算法作用于各规模下的矩阵M时的运行效率表现如图1所示.可见运行效率-矩阵规模的对数图像近似线性,斜率对应于$O(n^3)$.

3.2 数值稳定性

考察正交性损失$\|Q^*Q-I\|_{\mathsf{F}}$和标准偏差$\frac{\|Q^*AQ-M\|_{\mathsf{F}}}{\|A\|_{\mathsf{F}}}$, 它们的数值表现如图2. 可见$\|Q^*Q-I\|_{\mathsf{F}}$的数量级大约为$10^{-13}$,$\frac{\|Q^*AQ-M\|_{\mathsf{F}}}{\|A\|_{\mathsf{F}}}$的数量级大约为$10^{-7}$,二者均为极小量, 故可以认为Francis QR算法具有良好的数值稳定性.

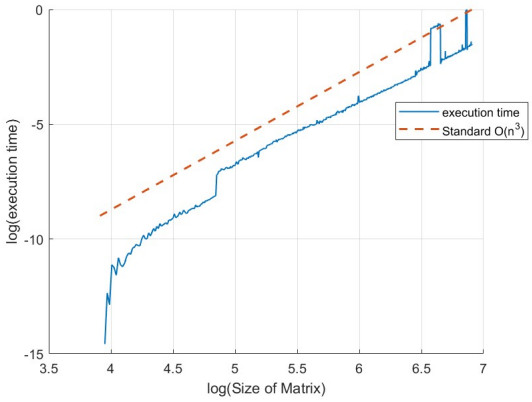

图 1: 运行时间

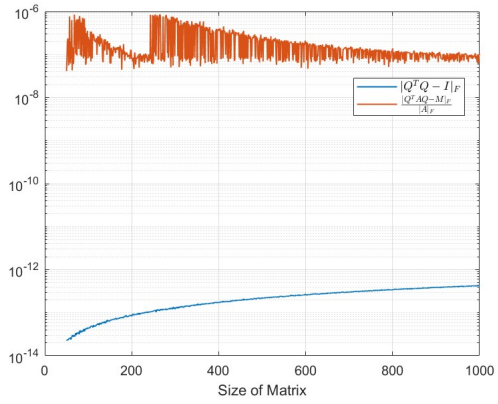

图 2: 数值稳定性

3.3 迭代次数

由图3可见平均迭代1.3次即可得到一个特征值,即该算法具有相当快的收敛速度.

3.4 收敛矩阵的结构

取规模为2500阶的初始矩阵M进行Francis QR算法, 最终得到的矩阵元素的热力图如图4, 可见这是一个拟上三角阵.

注: 为了避免对角线上的元素过大导致上三角中的其它元素在热力图上的表现近似于0, 在绘制热力图时对角线上的元素均被做了缩小一定倍数的处理.

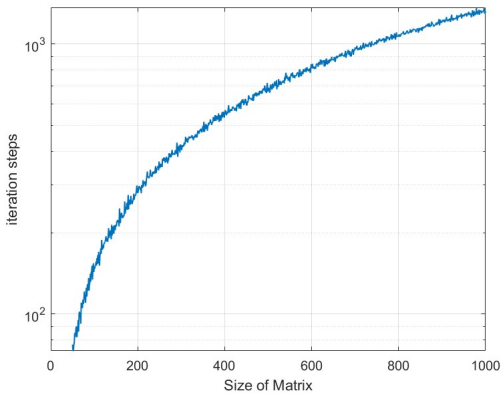

图 3: 迭代次数

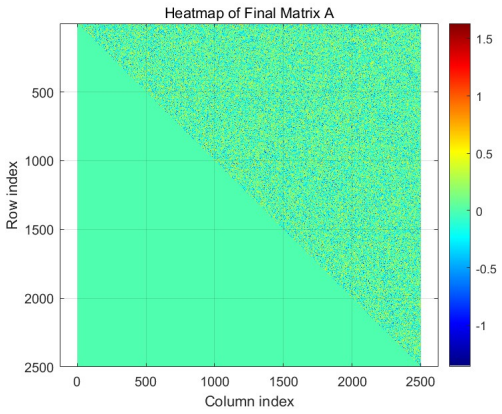

图 4: 收敛矩阵结构

4 声明

在完成这份报告时,作者在以下方面使用了ChatGPT辅助作业:

1. 辅助将写好的代码转换为LaTeX格式的伪代码.
2. 辅助撰写Matlab可视化数据的代码.

除了上面提到的部分,其余内容均没有借用ChatGPT或其它可能的LLMs.

参考文献

[1] 徐树方,高立,张平文(2013). ”数值线性代数(第二版)”. 北京:北京大学出版社.

[2] https://www.cs.cornell.edu/ bindel/class/cs6210-f12/notes/lec28.pdf

[3] Trefethen, L. N., & Bau, D. (1997). ”Numerical linear algebra”. Society for Industrial and Applied Mathematics.

[4] J.G.F. Francis, ”The QR Transformation, I”, The Computer Journal, 4(3), pages 265–271 (1961, received October 1959).

A 基于Householder变换的矩阵上Hessenberg化

Algorithm 2 Householder Transformation to Hessenberg Form

Require: M: input matrix of size $n \times n$

Ensure: Compute the Upper Hessenberg form of M using Householder transformations

function HOUSEHOLDER_HESSENBERG(A)
    $n \leftarrow$ number of rows (and columns) of A
    for $i = 1$ to $n - 2$ **do**
        $x \leftarrow$ subvector of A from row $i + 1$ to n, column i
        Compute the Householder matrix H that sets x to e_1
        $A[i + 1 : n, i : n] \leftarrow H \times A[i + 1 : n, i : n]$
        $A[1 : n, i + 1 : n] \leftarrow A[1 : n, i + 1 : n] \times H^T$
    end for
    return A
end function

B Wilkinson隐式位移

Algorithm 3 Single Shift Iteration with Givens Rotation

Require: A: input matrix of size $n \times n$, l: the left side of the submatrix, r: the right side of the submatrix, μ: shift parameter, Q: orthogonal matrix (initially identity matrix)

Ensure: Compute the QR factorization of A using Givens rotations with a single shift

function SINGLE_SHIFT_ITERATION(A, l, r, μ, Q)
    $[A, Q] \leftarrow$ INITIAL_GIVENS(A, l, r, μ, Q)
    $[A, Q] \leftarrow$ GIVENS_ROTATION(A, l, r, Q)
    return A, Q
end function

Algorithm 4 The first step of iteration

function INITIAL_GIVENS(A, l, r, μ, Q)
    Compute the Givens matrix G that satisfies the formula (1).
    $A(l : l+1, :) \leftarrow G \times A(l : l+1, :)$
    $A(1 : \min(l+2, r), l : l+1) \leftarrow A(1 : \min(l+2, r), l : l+1) \times G^T$
    $Q(l : l+1, :) \leftarrow G \times Q(l : l+1, :)$
    return A, Q
end function

Algorithm 5 The Givens rotations during the iteration

function GIVENS_ROTATION(A, l, r, Q)
    $n \leftarrow$ size of A
    for $i = l$ to $r-2$ **do**
        Compute the Givens matrix G that eliminates the bulge
        $A(i+1 : i+2, i : n) \leftarrow G \times A(i+1 : i+2, i : n)$
        $A(1 : \min(i+3, r), i+1 : i+2) \leftarrow A(1 : \min(i+3, r), i+1 : i+2) \times G^T$
        $Q(i+1 : i+2, :) \leftarrow G \times Q(i+1 : i+2, :)$
    end for
    return A, Q
end function

C Francis隐式双位移

Algorithm 6 Double Shift Iteration with Householder Reflection and Givens Rotation

Require: A: input matrix of size $n \times n$, l: the left side of the submatrix, r: the right side of the submatrix, m_{real}: twice the real part of the shift, m_{norm}: norm of the shift, Q: orthogonal matrix

Ensure: Compute the QR factorization of A using Householder reflections and Givens rotations with double shift

function DOUBLE_SHIFT_ITERATION($A, l, r, m_{\text{real}}, m_{\text{norm}}, Q$)
    $[A, Q] \leftarrow$ INITIAL_HOUSEHOLDER($A, l, r, m_{\text{real}}, m_{\text{norm}}, Q$)
    $[A, Q] \leftarrow$ HOUSEHOLDER_REFLECTION(A, l, r, Q)
    $[A, Q] \leftarrow$ GIVENS_ROTATION(A, l, r, Q)
    return A, Q
end function

Algorithm 7 The first step of the iteration

function INITIAL_HOUSEHOLDER($A, l, r, m_{\text{real}}, m_{\text{norm}}, Q$)
    $n \leftarrow$ the size of A
    $x \leftarrow$ the first column vector of $A^2 - 2\Re(\mu)A + |\mu|^2 I$
    Compute the Householder matrix P that sets e_1 to x
    $A(l : l+2, l : n) \leftarrow P' \cdot A(l : l+2, l : n)$
    $A(1 : \min(l+3, r), l : l+2) \leftarrow A(1 : \min(l+3, r), l : l+2) \cdot P$
    $Q(l : l+2, :) \leftarrow P' \cdot Q(l : l+2, :)$
    return A, Q
end function

Algorithm 8 The Householder reflections during the iteration

function HOUSEHOLDER_REFLECTION(A, l, r, Q)
    $n \leftarrow$ the size of A
    for $i = l$ to $r - 3$ **do**
        Compute the Householder matrix H that eliminates the bulge
        $A(i+1 : i+3, i : n) \leftarrow H \cdot A(i+1 : i+3, i : n)$
        $A(1 : \min(i+4, r), i+1 : i+3) \leftarrow A(1 : \min(i+4, r), i+1 : i+3) \cdot H'$
        $Q(i+1 : i+3, :) \leftarrow H \cdot Q(i+1 : i+3, :)$
    end for
    return A, Q
end function

Algorithm 9 The Givens rotation at the end of the iteration

function GIVENS_ROTATION(A, l, r, Q)
    $n \leftarrow$ the size of A
    Compute the Givens rotation G that eliminates the bulge
    $A(r-1 : r, r-2 : n) \leftarrow G \cdot A(r-1 : r, r-2 : n)$
    $A(1 : r, r-1 : r) \leftarrow A(1 : r, r-1 : r) \cdot G^T$
    $Q(r-1 : r, :) \leftarrow G \cdot Q(r-1 : r, :)$
    return A, Q
end function