Implimentation of Francis QR Algorithm for Real Matrices

Haoyang Chen, School of Data Science, Fudan University
23307130004@m.fudan.edu.cn

1 Algorithm

给定一个实矩阵 A, 首先通过 Householder 镜像变换将其转换为上 Hessenberg 形式; 然后, 通过双重步位移隐式 QR 迭代将其转换为实 Schur 形式. 具体而言, 我们选择右下角 2×2 的子矩阵, 位移值通过子矩阵的特征值来确定, 为了避免可能的复数运算, 算法隐式地进行 QR 迭代; QR 迭代的过程中, 算法通过小规模的 Householder 镜像变换, 处理 "bulge". 一次迭代后, 从下副对角线的右下角向左上角检索收敛为零的元素, 将其置零，否则继续迭代. 每次检索到一个收敛为零的元素, 算法都会使得矩阵的规模减小, 递归到更小的子矩阵上执行 QR 迭代. 最终, 算法递归至 2×2 的子矩阵, 迭代结束. 通过双重步位移 QR 算法, 得到的矩阵 T 是拟上三角阵, 正交矩阵 Q 记录相似变换, 算法还会输出迭代次数和运行时间, 其主要结构如下所述.

Algorithm 1: Hessenberg Transformation

Input: Square matrix A of size $n \times n$

Output: Hessenberg matrix H, orthogonal matrix Q such that $Q^\top AQ = H$

$H = A$

$Q = \mathrm{I}_n$

for $k = 0$ to $n-3$ do

  $x = A[k+1:n, k]$

  $H = \mathrm{house}(x)$

  $A[k+1:n, k:n] = H \cdot A[k+1:n, k:n]$

  $A[0:n, k+1:n] = A[0:n, k+1:n] \cdot H$

  $Q[k+1:n, :] = H \cdot Q[k+1:n, :]$

end

return H, Q

算法利用 Householder 镜像变换将矩阵转换为上 Hessenberg 形式.

Algorithm 2: Francis Double Shift QR Algorithm for Hessenberg Matrices

Input: Hessenberg matrix H, accumulation matrix Q, iteration index i

Output: Updated Hessenberg matrix H, updated Q such that

$Q^\top \ Origin \ H \ Q = Updated \ H$

$n = i$, $m = n - 1$

$s = H[m-1, m-1] + H[n-1, n-1]$

$t = H[m-1, m-1] \cdot H[n-1, n-1] - H[m-1, n-1] \cdot H[n-1, m-1]$

$x = H[0,0] \cdot H[0,0] + H[0,1] \cdot H[1,0] - s \cdot H[0,0] + t$

$y = H[1,0] \cdot (H[0,0] + H[1,1] - s)$, $z = H[1,0] \cdot H[2,1]$

for $k = 0$ **to** $n-3$ **do**

    $House$ = house(np.array($([x, y, z])$))

    $H[k : k+3, :] = House \cdot H[k : k+3, :]$

    $H[:, k : k+3] = H[:, k : k+3] \cdot House$

    $Q[k : k+3, :] = House \cdot Q[k : k+3, :]$

    update x, y, z

end

$House$ = house(np.array($([x, y])$))

$H[n-2 : n, n-3 :] = House \cdot H[n-2 : n, n-3 :]$

$H[:, n-2 : n] = H[:, n-2 : n] \cdot House$

$Q[n-2 : n, :] = House \cdot Q[n-2 : n, :]$

return H, Q

算法如此隐式地进行 QR 迭代, 并处理”bulge”.

Algorithm 3: Hessenberg to Schur Form

Input: Hessenberg matrix H

Output: Real Schur form T, orthogonal matrix Q such that $Q^\top H Q = T$

$T = H$, $Q = \mathrm{I}_n$, $m = dimension$, iteration_time $= 0$

if $m \leq 2$ **then** trivial return;

while $m > 2$ **do**

    found_block = False

    for $i = m-1$ **to** 0 **step** -1 **do**

        if $|T[i, i-1]| < \mathrm{tol} \cdot (|T[i-1, i-1]| + |T[i, i]|)$ **then**

            $T[i, i-1] = 0$, $m = i$, found_block = True

            break

        end

    end

    if not found_block **then** T, Q = (Algorithm 2)francis_one(T, Q, m);

    iteration_time = iteration_time + 1

end

return T, Q, iteration_time

算法不断迭代, 直至矩阵规模减小至 2×2，迭代结束.

2 Code Structure

Francis QR Double Shift Part 01: Calculation Function

This section includes the core calculation functions used for Hessenberg transformation, Francis double-shift QR iteration, and Schur form computation.

1. house(x):
   - Input: a vector x to be transformed.
   - Output: a Householder transformation matrix H such that $Hx = \|x\|e_1$, where e_1 is the first standard basis vector.
   - Special Note: Used to reflect vectors during Hessenberg and QR transformation.
2. hessenberg(origin_A):
   - Input: a square matrix A.
   - Output: The Hessenberg form of A and the orthogonal transformation matrix Q such that $Q^T AQ = H$.
3. francis_one(H, Q_accum, i):
   - Input: a Hessenberg matrix H, the accumulated orthogonal matrix Q_accum, and the iteration index i.
   - Output: Updated Hessenberg matrix H and the accumulated orthogonal matrix Q_accum.
   - Special Note: Performs a single iteration of the Francis double-shift QR algorithm.
4. hessenberg_to_schur_form(H, max_iter=np.inf, tol=1e-16):
   - Input: a Hessenberg matrix H, maximum number of iterations, and tolerance for convergence.
   - Output: Schur form H, orthogonal transformation Q_accum, and the number of iterations performed.
5. zeros_hessenberg(H):
   - Input: a Hessenberg matrix H.
   - Output: Hessenberg matrix with lower subdiagonal entries explicitly set to zero.

6. generate_test_matrix(n, seed=None):
   - Input: Matrix dimension n and an optional random seed.
   - Output: a randomly generated square matrix of dimension n.
7. francis_double_shift_qr(A):
   - Input: a square matrix A.
   - Output: Schur form T, orthogonal transformation matrix Q, total iteration time, and execution time.
   - Special Note: Combines Hessenberg transformation and Francis double-shift QR algorithm to compute the Schur form.

Francis QR Double Shift Part 02: Plot Figure Function

This section includes functions for visualizing the performance metrics of the algorithm.

1. collect_information(dimensions, seed=None):
   - Input: a list of matrix dimensions and an optional random seed.
   - Output: a dictionary of performance information, including orthogonality loss, forward error, and running time and so on.
2. plot_orthogonality_loss(dimensions, orthogonality_losses):
3. plot_forward_abs_error(dimensions, forward_abs_errors):
4. plot_forward_rel_error(dimensions, forward_rel_errors):
5. plot_running_time(dimensions, running_times):
6. plot_iteration_time(dimensions, iteration_times):
7. plot_iterations_per_dimension(dimensions, iterations_per_dimension):
8. plot_iterations_per_time(dimensions, iterations_per_time):

Francis QR Double Shift Part 03: Demonstration

This section demonstrates the use of the above functions for testing and visualizing the algorithm's performance.

1. one_test(dimension):

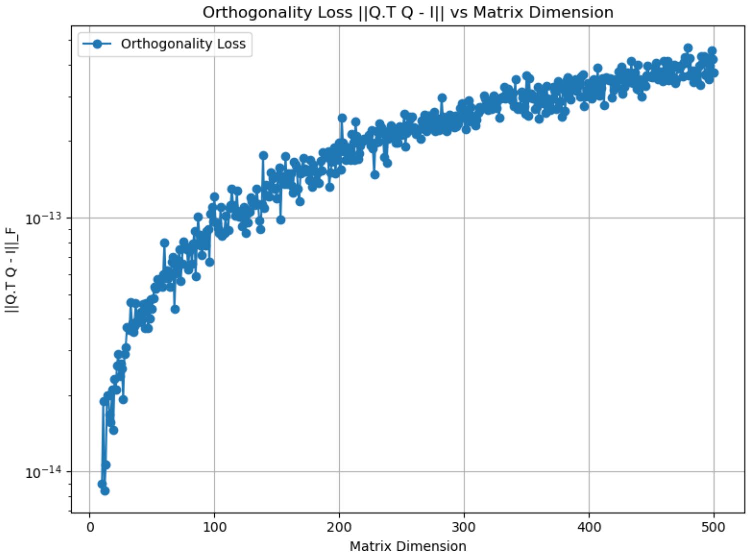
- Input: Dimension of the square matrix to test.
- Output: Various metrics, including orthogonality loss, forward error, and eigenvalues, written to a text file.

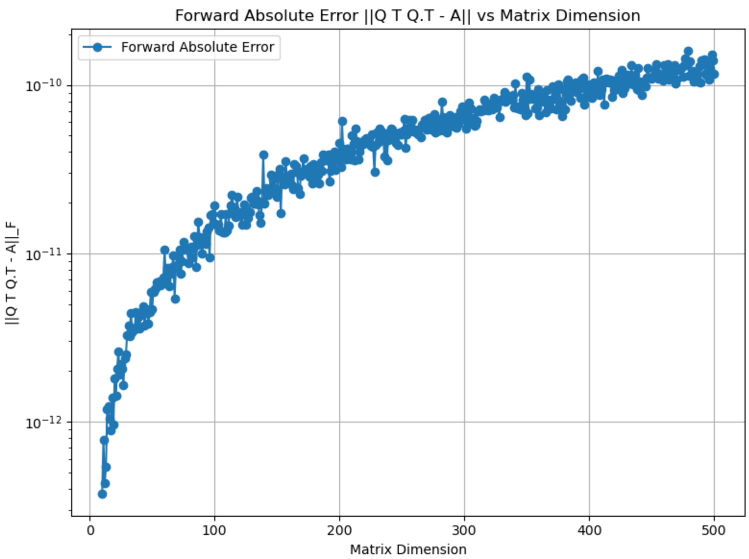
2. various_test():

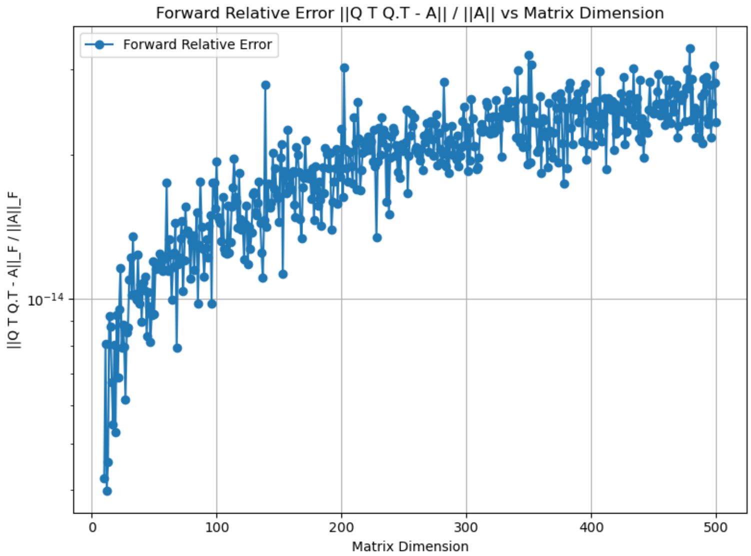
- Input: None (tests predefined dimensions).
- Output: Generates and saves performance plots for orthogonality loss, forward error, and iteration statistics.

3 Numerical Experiments

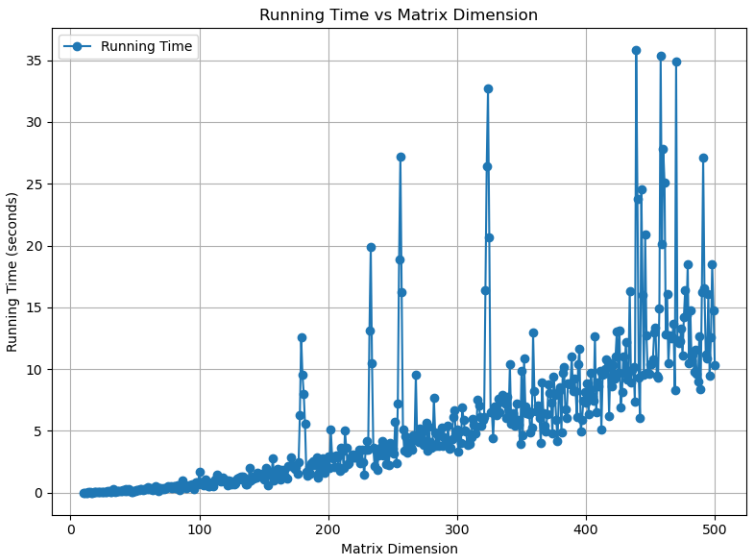
3.1 Orthogonality Loss



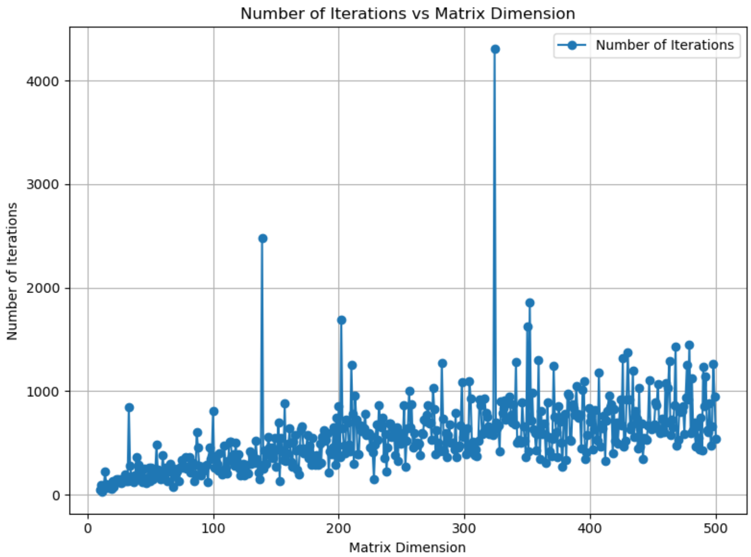


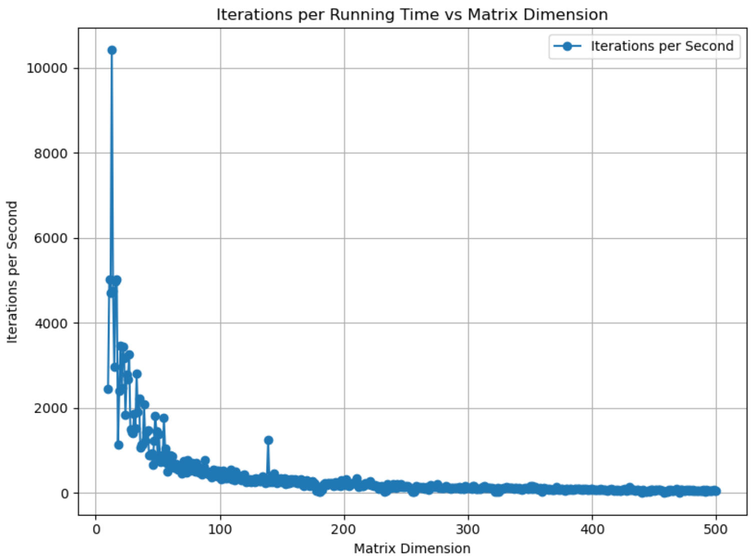


3.2 Execution Time



3.3 Convergence Speed





4 Acknowledgements

学生向授课教师邵美悦教授表达感谢, 感谢他在数值算法与案例分析课程中提供的指导和支持. 感谢本课程的助教们, 特别感谢助教叶爵达耐心批改我的作业, 提供反馈意见. 感谢舍友们在我探索路上的陪伴, 和他们的交流激励我不断进步.

5 References

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

[2] Lloyd N. Trefethen, David Bau III: 数值线性代数, 陆金甫, 关治译, 人民邮电出版社 2006 年版.