The University of Texas
at Austin

CS383C Numerical Analysis

Codes for QR Factorization

Edited by LaTeX

Department of Computer Science

STUDENT

Jimmy Lin
xl5224

COURSE COORDINATOR

Robert A. van de Geijn

UNIQUE NUMBER

53180

RELEASE DATE

Sep. 25 2014

DUE DATE

Oct. 02 2014

TIME SPENT

2 hours

September 25, 2014

Exercise 1. Classical Gram-Schmidt (CGS)

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Project 01: CGS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Copyright 2014 The University of Texas at Austin
%
% For licensing information see
%                http://www.cs.utexas.edu/users/flame/license.html
%
% Programmed by: Jimmy Lin
%                jimmylin@utexas.edu

function [ A_out, R_out ] = CGS_unb( A, R )

  [ AL, AR ] = FLA_Part_1x2( A, ...
                             0, 'FLA_LEFT' );

  [ RTL, RTR, ...
    RBL, RBR ] = FLA_Part_2x2( R, ...
                               0, 0, 'FLA_TL' );

  while ( size( AL, 2 ) < size( A, 2 ) )

    [ A0, a1, A2 ]= FLA_Repart_1x2_to_1x3( AL, AR, ...
                                           1, 'FLA_RIGHT' );

    [ R00,  r01,   R02,  ...
      r10t, rho11, r12t, ...
      R20,  r21,   R22 ] = FLA_Repart_2x2_to_3x3( RTL, RTR, ...
                                                  RBL, RBR, ...
                                                  1, 1, 'FLA_BR' );

    %------------------------------------------------------------%

    r01 = A0' * a1;
    a1 = a1 - A0 * r01;
    rho11 = norm(a1, 2);
    a1 = a1 / rho11;

    %------------------------------------------------------------%

    [ AL, AR ] = FLA_Cont_with_1x3_to_1x2( A0, a1, A2, ...
                                           'FLA_LEFT' );

    [ RTL, RTR, ...
      RBL, RBR ] = FLA_Cont_with_3x3_to_2x2( R00,  r01,   R02,  ...
                                             r10t, rho11, r12t, ...
                                             R20,  r21,   R22, ...
                                             'FLA_TL' );

  end

  A_out = [ AL, AR ];

  R_out = [ RTL, RTR
            RBL, RBR ];

return
```

Exercise 2. Modified Gram-Schmidt (MGS)

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Project 01: MGS (Alternative)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Copyright 2014 The University of Texas at Austin
%
% For licensing information see
%                http://www.cs.utexas.edu/users/flame/license.html
%
% Programmed by: Jimmy Lin
%                jimmylin@utexas.edu

function [ A_out, R_out ] = MGS_unb( A, R )

  [ AL, AR ] = FLA_Part_1x2( A, ...
                             0, 'FLA_LEFT' );

  [ RTL, RTR, ...
    RBL, RBR ] = FLA_Part_2x2( R, ...
                             0, 0, 'FLA_TL' );

  while ( size( AL, 2 ) < size( A, 2 ) )

    [ A0, a1, A2 ]= FLA_Repart_1x2_to_1x3( AL, AR, ...
                                       1, 'FLA_RIGHT' );

    [ R00,  r01,   R02,  ...
      r10t, rho11, r12t, ...
      R20,  r21,   R22 ] = FLA_Repart_2x2_to_3x3( RTL, RTR, ...
                                                  RBL, RBR, ...
                                                  1, 1, 'FLA_BR' );

    %------------------------------------------------------------%

    rho11 = norm(a1);
    a1 = a1 / rho11;
    r12t = a1' * A2;
    A2 = A2 - a1 * r12t;

    %------------------------------------------------------------%

    [ AL, AR ] = FLA_Cont_with_1x3_to_1x2( A0, a1, A2, ...
                                       'FLA_LEFT' );

    [ RTL, RTR, ...
      RBL, RBR ] = FLA_Cont_with_3x3_to_2x2( R00,  r01,   R02,  ...
                                             r10t, rho11, r12t, ...
                                             R20,  r21,   R22, ...
                                             'FLA_TL' );

  end

  A_out = [ AL, AR ];

  R_out = [ RTL, RTR
            RBL, RBR ];

return
```

Exercise 3. Householder QR Transformation

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Project 01: Householder QR Transformation
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Copyright 2014 The University of Texas at Austin
%
% For licensing information see
%                http://www.cs.utexas.edu/users/flame/license.html
%
% Programmed by: Jimmy Lin
%                jimmylin@utexas.edu

function [ A_out, T_out ] = HQR_unb ( A, T )

  [ ATL, ATR, ...
    ABL, ABR ] = FLA_Part_2x2( A, ...
                               0, 0, 'FLA_TL' );

  [ TT, ...
    TB ] = FLA_Part_2x1( T, ...
                         0, 'FLA_TOP' );

  while ( size( ATL, 1 ) < size( A, 1 ) )

    [ A00,  a01,     A02,  ...
      a10t, alpha11, a12t, ...
      A20,  a21,     A22 ] = FLA_Repart_2x2_to_3x3( ATL, ATR, ...
                                                    ABL, ABR, ...
                                                    1, 1, 'FLA_BR' );
    [ T0, ...
      t1t, ...
      T2 ] = FLA_Repart_2x1_to_3x1( TT, ...
                                    TB, ...
                                    1, 'FLA_BOTTOM' );
    %----------------------------------------------------------%

    [ alpha11, a21, t1t ] = Housev( alpha11, a21 );
    w12t = (a12t + a21' * A22) / t1t;
    a12t = a12t - w12t;
    A22 = A22 - a21 * w12t;

    %----------------------------------------------------------%

    [ ATL, ATR, ...
      ABL, ABR ] = FLA_Cont_with_3x3_to_2x2( A00,  a01,     A02,  ...
                                             a10t, alpha11, a12t, ...
                                             A20,  a21,     A22, ...
                                             'FLA_TL' );
    [ TT, ...
      TB ] = FLA_Cont_with_3x1_to_2x1( T0, ...
                                       t1t, ...
                                       T2, ...
                                       'FLA_TOP' );
  end

  A_out = [ ATL, ATR
            ABL, ABR ];
  T_out = [ TT
            TB ];
return
```

Exercise 4. FormQ algorithm

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Project 01: FormQ_unb algorithm
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Copyright 2014 The University of Texas at Austin
%
% For licensing information see
%                http://www.cs.utexas.edu/users/flame/license.html
%
% Programmed by: Jimmy Lin
%                jimmylin@utexas.edu

function [ A_out, T_out ] = FORMQ_unb ( A, T )

  [ ATL, ATR, ...
    ABL, ABR ] = FLA_Part_2x2( A, ...
                               0, 0, 'FLA_BR' );

  [ TT, ...
    TB ] = FLA_Part_2x1( T, ...
                         0, 'FLA_BOTTOM' );

  while ( size( ABR, 1 ) < size( A, 1 ) )
    [ A00,  a01,     A02,  ...
      a10t, alpha11, a12t, ...
      A20,  a21,     A22 ] = FLA_Repart_2x2_to_3x3( ATL, ATR, ...
                                                    ABL, ABR, ...
                                                    1, 1, 'FLA_TL' );
    [ T0, ...
      t1t, ...
      T2 ] = FLA_Repart_2x1_to_3x1( TT, ...
                                    TB, ...
                                    1, 'FLA_TOP' );
    %------------------------------------------------------------%

    alpha11 = 1 - 1 / t1t;
    a12t = - (a21' * A22) / t1t;
    A22 = A22 + a21 * a12t;
    a21 = - a21 / t1t;

    %------------------------------------------------------------%

    [ ATL, ATR, ...
      ABL, ABR ] = FLA_Cont_with_3x3_to_2x2( A00,  a01,     A02,  ...
                                             a10t, alpha11, a12t, ...
                                             A20,  a21,     A22, ...
                                             'FLA_BR' );

    [ TT, ...
      TB ] = FLA_Cont_with_3x1_to_2x1( T0, ...
                                       t1t, ...
                                       T2, ...
                                       'FLA_BOTTOM' );
  end

  A_out = [ ATL, ATR
            ABL, ABR ];
  T_out = [ TT
            TB ];

return
```