

I. CMPE 360 HOMEWORK IV ANSWERS

A. 2.7

```
function cp02_07 % growth factor in Gaussian elimination with partial pivoting

disp('(a) With partial pivoting, entries of U grow exponentially.')
disp('With complete pivoting, entries of U do not grow.')
n = 1;
condU = 1;
disp('(b)')
disp(' n resid_norm error_norm condA condU')
while condU < 1e+16
    n = n+1;
    A = eye(n,n)-tril(ones(n,n),-1);
    A(1:n,n) = ones(n,1);
    condA = cond(A);
    x_true = ones(n,1);
    b = A*x_true;
    [L, U] = lu(A);
    condU = cond(U);
    x = U\(L\b);
    r_norm = norm(b-A*x,inf);
    e_norm = norm(x-x_true,inf);
    fprintf('%2d %13.6e %13.6e %13.6e %13.6e\n', n, r_norm,e_norm, condA, condU)
end
```

B. 2.11

```
function cp02_11(n) % pivoting strategies for Gaussian elimination

A = rand(n,n); x_true = ones(n,1);
b = A*x_true;
x_np = GaussElim_np(A, b);
e_norm_np = norm(x_np-x_true,inf);
r_norm_np = norm(b-A*x_np,inf);
x_pp = GaussElim_pp(A,b);
e_norm_pp = norm(x_pp-x_true,inf);
r_norm_pp = norm(b-A*x_pp,inf);
x_cp = GaussElim_cp(A,b);
e_norm_cp = norm(x_cp-x_true,inf);
r_norm_cp = norm(b-A*x_cp,inf);
disp('e_norm_np r_norm_np e_norm_pp r_norm_pp e_norm_cp r_norm_cp')
fprintf('%9.4e %9.4e %9.4e %9.4e %9.4e %9.4e\n', ...
e_norm_np, r_norm_np, e_norm_pp, r_norm_pp, e_norm_cp, r_norm_cp)

function [x] = GaussElim_np(A, b) % Gaussian elimination without pivoting

n = size(A,1); x = zeros(n,1);
for i=1:n
    p = A(i,i);
    for j=(i+1):n
        A(j,(i+1):n) = A(j,(i+1):n)-A(i,(i+1):n)*(A(j,i)/p);
        b(j) = b(j)-b(i)*(A(j,i)/p); % forward substitution
    end;
end;
x(n) = b(n)/A(n,n);
for i = n-1:-1:1 % back-substitution
    x(i) = (b(i)-dot(x((i+1):n),A(i,(i+1):n)))/A(i,i);
end
```

```
function [x] = GaussElim_pp(A,b) % Gaussian elimination with partial pivoting
```

```
n = size(A,1);
x = zeros(n,1);
for i = 1:n
    [p, maxk] = max(abs(A(i:n,i))); % pivot search
    maxk = maxk+i-1; p = A(maxk,i);
    if i ~= maxk % row interchange
        tmp = A(i,i:n); A(i,i:n) = A(maxk, i:n); A(maxk, i:n) = tmp;
        z = b(i); b(i) = b(maxk); b(maxk) = z;
    end
    for j = (i+1):n % elimination
        A(j,(i+1):n) = A(j,(i+1):n)-A(i,(i+1):n)*(A(j,i)/p);
        b(j) = b(j)-b(i)*(A(j,i)/p); % forward substitution
    end;
end;
x(n) = b(n)/A(n,n);
for i = n-1:-1:1 % back-substitution
    x(i) = (b(i)-dot(x((i+1):n),A(i,(i+1):n)))/A(i,i);
end
```

```
function [x] = GaussElim_cp(A,b) % Gaussian elimination with complete pivoting
```

```
n = size(A,1);
x = zeros(n,1);
index = 1:n;
for i = 1:n
    [maxA, maxK] = max(A(i:n,i:n)); % pivot search
    [p, maxl] = max(maxA); maxk = maxK(maxl)+i-1;
    maxl = maxl+i-1; p = A(maxk,maxl);
    if i ~= maxk % row interchange
        tmp = A(i,:); A(i,:) = A(maxk,:); A(maxk,:) = tmp;
        z = b(i); b(i) = b(maxk); b(maxk) = z;
    end
    if i ~= maxl % column interchange
        tmp = A(:,i); A(:,i) = A(:,maxl); A(:,maxl) = tmp;
        ii = index(i); index(i) = index(maxl); index(maxl) = ii;
    end;
    for j = (i+1):n % elimination
        A(j,(i+1):n) = A(j,(i+1):n)-A(i,(i+1):n)*(A(j,i)/p);
        b(j) = b(j)-b(i)*(A(j,i)/p); % forward substitution
    end;
end; x(n) = b(n)/A(n,n);
for i = n-1:-1:1 % back-substitution
    x(i) = (b(i)-dot(x((i+1):n),A(i,(i+1):n)))/A(i,i);
end;
y = x;
for i = 1:n % reverse permutation
    x(i) = y(index(i));
end;
```