ACM106a - Homework 2 Solutions

prepared by Svitlana Vyetrenko

October 17, 2006

1. **Chapter 21, problem 21.2 (solution adapted from Golub, Van Loan, pp.152-154):**

   For the proof we will use the fact that if $A \in \mathbb{C}^{m\times m}$ has an LU-factorization $A = LU$ and has an upper bandwidth q and a lower bandwidth p, then U has an upper bandwidth q and L has a lower bandwidth p (see Golub, Van Loan, p.152, theorem 4.3.1).

   Now let $PA = LU$ be the factorization computed by Gaussian elimination with partial pivoting. $P = P_{n-1}\dots P_1$ - a product of permutation matrices. $P_T = [e_{s_1}\dots e_{s_n}]$, where $s_1, s_2 \dots s_n$ is a permutation of $1, 2, \dots n$. If $s_i > i + p$, then it follows that the leading $i \times i$ principal submatrix of PA is singular, since $(PA)_{ij} = a_{s_i,j}$ for $j = 1 \dots s_i - p - 1$ and $s_i - p - 1 \geq i$. This implies that U and A are singular, thus, we reach a contradiction. Therefore, $s_i \leq i + p$ for $i = 1 \dots n$, thus, PA has an upper bandwidth $p+p = 2p$. Thus, by the above mentioned fact we see that U has an upper bandwidth $2p$.

   Note that pivoting destroys the band structure in a sense that U ends up having a higher bandwidth, while nothing at all can be said about the band structure of L (in fact, since $L = P(L_{n-1}P_{n-1}\dots L_1P_1)^{-1}$ the only thing we can say is that L contains at most $p + 1$ nonzero elements per column).

2. **Chapter 21, problem 21.6:**

   For the first step of Gaussian elimination with partial pivoting, the entry of maximum modulus is a_{11}, therefore, no row interchange is necessary.

   Let $A^{(1)}$ be the matrix obtained after the first step: $A^{(1)} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ 0 & a_{22}^{(1)} & \dots & a_{2n}^{(1)} \\ \dots & \dots & \dots & \dots \\ 0 & a_{n2}^{(1)} & \dots & a_{nn}^{(1)} \end{pmatrix}$

   Let's show that $A^{(1)}$ is again column diagonally dominant:

$$\begin{aligned} \sum_{i=2,i\neq j}^{n} |a_{ij}^{(1)}| &= \sum_{i=2,i\neq j}^{n} |a_{ij} - \frac{a_{1j}a_{i1}}{a_{11}}| \\ &\leq \sum_{i=2,i\neq j}^{n} |a_{ij}| + \sum_{i=2,i\neq j}^{n} |\frac{a_{1j}a_{i1}}{a_{11}}| \\ &< (|a_{jj}| - |a_{1j}|) + \frac{|a_{1j}|(|a_{11}| - |a_{j1}|)}{|a_{11}|} \\ &= |a_{jj} - \frac{a_{j1}a_{1j}}{a_{11}}| = |a_{jj}^{(1)}|, \end{aligned}$$

   where the second inequality and the fact that $|a_{jj}||a_{11}| > |a_{1j}||a_{j1}|$ follow from the diagonal dominance of A.

Thus, the pivot for the second step is $a_{22}^{(1)}$ and, consequently, no row interchange is necessary. Similarly, this process can be continued to show that each step of Gaussian elimination results in a strictly column diagonally dominant matrix, therefore, the element with the maximum absolute value will be on the diagonal and no row interchanges will take place.

3. **Chapter 23, problem 23.3:**

   We want to see how long it takes to solve $Ax = b$ (i.e. which solver is being used to solve the system) depending on the structure of A (for example, if A is symmetric, symmetric positive definite, triangular, etc.) You can find a good description of what backslash actually does depending on the structure of A at `http://www.mathworks.com`.

   (a) Note that A is symmetric positive definite matrix (as a remark, A is invertible with probability 1). We expect that the system is going to be solved using Cholesky factorization (the flop count is $\approx \frac{m^3}{3}$).

   (b) Run it just to make sure that no operating system effects are incorporated into the estimated time for part (a).

   (c) A_2 is not symmetric. We expect that it is going to be solved using LU-factorization. Our guess is confirmed by time estimate (recall that Gaussian elimination with partial pivoting takes $\approx \frac{2m^3}{3}$ flops), which is twice as large as the time count in part (a).

   (d) A_3 is symmetric positive definite (note that if λ is the eigenvalue of B, then $\lambda - \sigma$ is the eigenvalue of $B - \sigma I$). Therefore, Cholesky factorization is used to solve the system. It costs roughly $\frac{m^3}{3}$ flops, which agrees with part (a).

   (e) A_4 is still a symmetric matrix. However, it is not positive definite since it has at least one eigenvalue which is less than zero. According to the mldivide algorithm documentation available at `http://www.mathworks.com`, we can assume that because of the symmetry of A_4 MATLAB attempts to run Cholesky factorization first, which subsequently fails and then proceeds with the indefinite symmetric factorization.

   (f) A_5 is upper triangular, thus the system is solved using back substitution - roughly m^2 flops.

   (g) A_6 is not symmetric again. The system is solved using Gaussian elimination with partial pivoting, which results in $\approx \frac{2m^3}{3}$ flops.

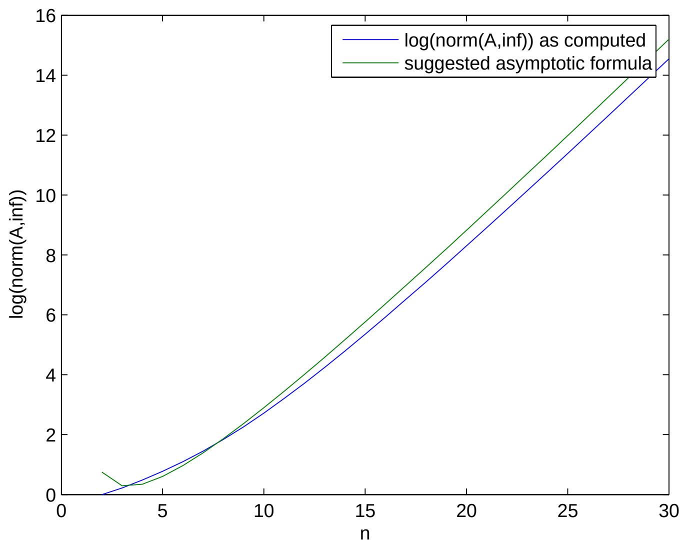
4. **Chapter 12, problem 12.2:**

   (a) Given n distinct points $x_1, x_2 \dots x_n$ and n corresponding function values $f(x_1), f(x_2) \dots f(x_n)$ there exists a unique polynomial of degree $n-1$ $P_n(x)$ such that $P_n(x_i) = f(x_i)$ (it is called Lagrange interpolating polynomial). Its explicit representation is: $P_n(x) = \sum_{i=1}^{n} f(x_i) \prod_{j=1, j\neq i}^{n} \frac{x - x_j}{x_i - x_j}$.

   Therefore, the entries of A are: $\prod_{j=1, j\neq i}^{n} \frac{y_k - x_j}{x_i - x_j}$, $k = 1 \dots m$, $i = 1 \dots n$.

   (b)
```
function pr122b;
for n=2:30
   m=2*n-1;
```

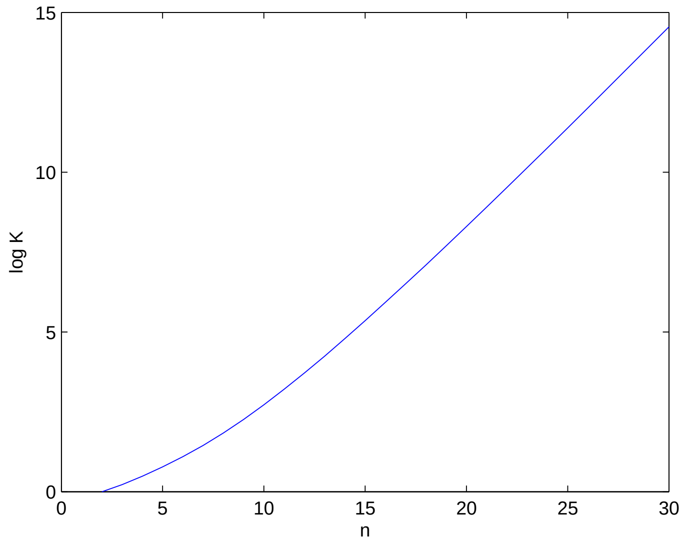
```
    f=1:n;
    x=-1+2*(f-1)/(n-1);
    f=1:m;
    y=-1+2*(f-1)/(m-1);

    A=ones(m,n);
    for i=1:m
    for j=1:n
        for f=1:n
           if (f~=j) A(i,j)=A(i,j)*(y(i)-x(f))/(x(j)-x(f));
           end;
        end;
    end;
    end;

    array(n,1)=n;
    array(n,2)=log(norm(A,inf));
    array(n,3)=n*log(2)-log(exp(1)*(n-1)*log(n));
end;

plot(array(2:30,1), array(2:30,2), array(2:30,1), array(2:30,3));
```



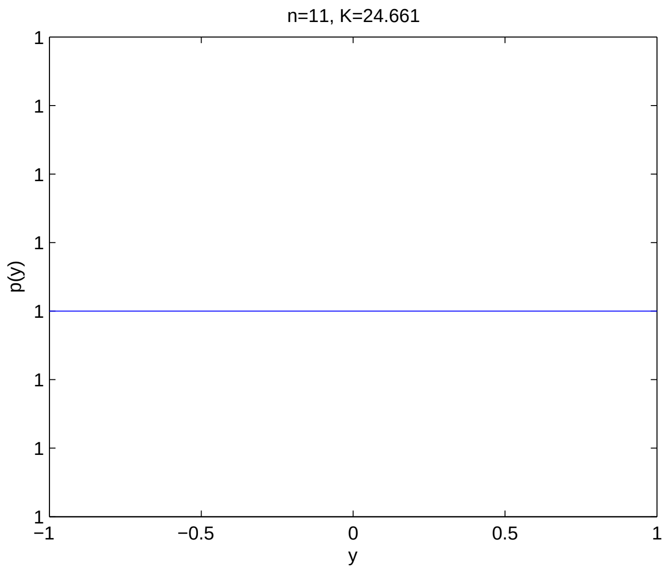
(c) Note that for our problem (12.6) is equivalent to (12.9).

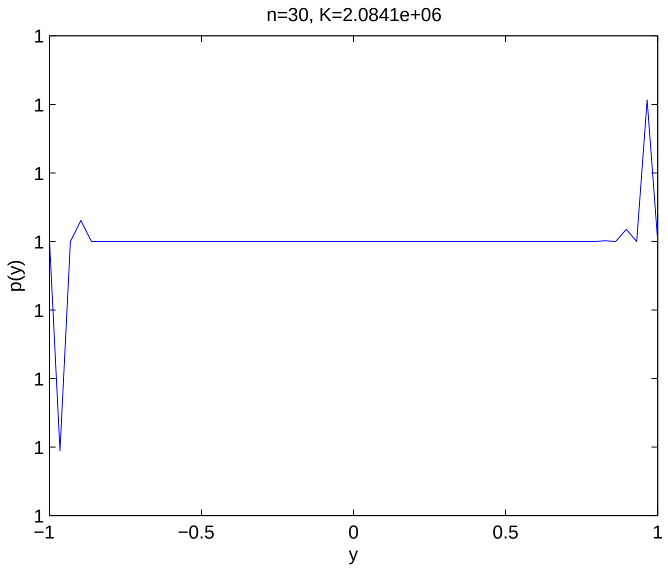
```
function pr122c;
for n=2:30
    m=2*n-1;

    f=1:n;
    x=-1+2*(f-1)/(n-1);
    f=1:m;
    y=-1+2*(f-1)/(m-1);
    val=ones(n,1);

    A=ones(m,n);
```

```
    for i=1:m
    for j=1:n
        for f=1:n
           if (f~=j) A(i,j)=A(i,j)*(y(i)-x(f))/(x(j)-x(f));
           end;
        end;
    end;
    end;

    array(n,1)=n;
    array(n,2)=(norm(A,inf)/norm(A*val, inf));
end;

plot(array(2:30,1),log( array(2:30,2)));
```



(d) Now let's use the code of the previous problem to sketch the computed polynomial interpolation for $n = 11$ and $n = 30$:





5. **Problem 5:**

   (a) We know that $(I + uv^T)x = b$, i.e. $x + uv^T x = b$.
   Multiply the last identity by v^T:

$$v^T x = \frac{v^T b}{1 + v^T u},$$

therefore

$$x + \frac{uv^T b}{1 + v^T u} = b,$$

thus

$$x = (I - \frac{uv^T}{1 + v^T u})b$$

and

$$(I + uv^T)^{-1} = I - \frac{uv^T}{1 + v^T u}$$

if $v^T u \neq -1$.

Note that when $v^T u = -1$ we have

$$(I + uv^T)u = u + u(v^T u) = u - u = 0,$$

hence 0 is the eigenvalue of $I + uv^T$, therefore, $I + uv^T$ is singular.

(b) Using the result of part (a) we have

$$\begin{aligned}(A + uv^T)^{-1} &= (I + A^{-1}uv^T)^{-1}A^{-1} \\ &= (I - \frac{1}{1 + v^T A^{-1}u}A^{-1}uv^T)A^{-1} \\ &= A^{-1} - \frac{1}{1 + v^T A^{-1}u}A^{-1}uv^T A^{-1}.\end{aligned}$$

(c) Having a fast solver for $Ax = b$ means that we can easily compute $x = A^{-1}b$.
If $Hx = b$, we have

$$x = H^{-1}b = (A + uv^T)b = A^{-1}b - \frac{1}{1 + v^T A^{-1}u}A^{-1}uv^T A^{-1}b.$$

Now since we know how to compute $A^{-1}b$ and $A^{-1}u$, we can obtain the solution to the system.

(d) Note that if A is orthonormal, $A^{-1} = A^T$ and therefore

$$x = A^T b - \frac{1}{1 + v^T A^T u}A^T uv^T A^T b.$$

```
function x=problem5(A,u,v,b)
pr1=A'*u;
pr2=A'*b;
x=pr2-1/(1+v'*pr1)*pr1*v'*pr2;
```