HW #2 Solutions: M552 Spring 2006

1. (**3.1**-*Trefethen & Bau*) Prove that if W is an arbitrary nonsingular matrix, the function $\|\cdot\|_W$ defined by $\|x\|_W = \|Wx\|$ is a vector norm.

   ANS: We need to show

   (i) $\|x\|_W \geq 0$, $\|x\|_W = 0 \;\Leftrightarrow\; x = 0$

   Given any x, $\|x\|_W = \|Wx\| \geq 0$ since $\|\cdot\| \geq 0$. Let $y = Wx$, and note since W is nonsingular $y = 0 \Leftrightarrow x = 0$. Then $\|x\|_W = \|y\| = 0 \Leftrightarrow y = 0 \Leftrightarrow x = 0$.

   (ii) $\|\alpha x\|_W = |\alpha|\|x\|_W$

   $\|\alpha x\|_W = \|W(\alpha x)\| = \|\alpha W x\| = |\alpha|\|Wx\|$, since this is true for the $\|\cdot\|$ norm. Thus, $\|\alpha x\|_W = |\alpha|\|Wx\| = |\alpha|\|x\|_W$.

   (iii) $\|x + y\|_W \leq \|x\|_W + \|y\|_W$

   $\|x + y\|_W = \|W(x + y)\| = \|Wx + Wy\| \leq \|Wx\| + \|Wy\|$, since $\|u + v\| \leq \|u\| + \|v\|$ for any u, v. Thus, $\|x + y\|_W \leq \|Wx\| + \|Wy\| = \|x\|_W + \|y\|_W$.

2. (**3.2**-*Trefethen & Bau*) Let $\|\cdot\|$ denote any norm on $\mathbb{C}^m$ and also the induced matrix norm on $\mathbb{C}^{m\times m}$. Show that $\rho(A) \leq \|A\|$, where $\rho(A)$ is the *spectral radius* of A, i.e., the largest absolute value $|\lambda|$ of an eigenvalue λ of A.

   ANS: First note that for any x, $\|Ax\| \leq \|A\|\|x\|$ since this is a property of any induced matrix norm. Now, let (λ, x) be any e-pair of A, i.e., $Ax = \lambda x$.Then

$$|\lambda|\|x\| = \|\lambda x\| = \|Ax\| \leq \|A\|\|x\| \quad \Rightarrow \quad |\lambda| \leq \|A\|.$$

   Note we divided by $\|x\| \neq 0$ since x is and e-vector, so we must have $x \neq 0$. Thus $|\lambda| \leq \|A\|$, and taking the sup over all λ on the left-handside (in fact a max since their are only a finite number of e-values), and noting the the right-handside is independent of λ gives the result.

3. (**3.3**-*Trefethen & Bau*) Vector and matrix p-norms are related by various inequalities, often involving the dimensions m or n. For each of the following, verify the inequality and give an example of a nonzero vector or matrix (for general m,n) for which equality is achieved. In this problem x is a m-vector and A is a $m \times n$ matrix.

   (a) $\|x\|_\infty \leq \|x\|_2$,

   (b) $\|x\|_2 \leq \sqrt{m}\|x\|_\infty$,

   (c) $\|A\|_\infty \leq \sqrt{n}\|A\|_2$,

   (d) $\|A\|_2 \leq \sqrt{m}\|A\|_\infty$.

<u>**ANS**</u>:

(a) $\|x\|_\infty = \max_i |x_i| = \max_i (|x_i|^2)^{1/2} \leq \left(\sum_{i=1}^m |x_i|^2\right)^{1/2} = \|x\|_2$. Equality is achieved for $x = e_k$, $k = 1, \ldots, m$, where e_k is one of the standard basis vectors for $\mathbb{C}^m$.

(b) $\|x\|_2 = \left(\sum_{i=1}^m |x_i|^2\right)^{1/2} \leq \left(\sum_{i=1}^m \max_i |x_i|^2\right)^{1/2} = \left(m\|x\|_\infty^2\right)^{1/2} = \sqrt{m}\|x\|_\infty$. For equality take $x = (1, 1, \ldots, 1)^T \in \mathbb{C}^m$. Then $\|x\|_\infty = 1$ and $\|x\|_2 = \sqrt{m}$.

(c) We have

$$\begin{aligned} \|A\|_\infty &= \sup_{x\neq 0} \frac{\|Ax\|_\infty}{\|x\|_\infty} \\ &\leq \sup_{x\neq 0} \frac{\|Ax\|_2}{\|x\|_\infty} && \text{using (a) in the numerator} \\ &\leq \sup_{x\neq 0} \frac{\|Ax\|_2}{\|x\|_2/\sqrt{n}} && \text{using (b) in the denominator, and noting } x \in \mathbb{C}^n \\ &= \sqrt{n}\|A\|_2 \end{aligned}$$

For equality, let $A \in \mathbb{C}^{m\times n}$ be the matrix whose first row is all ones, and zeros elsewhere. Clearly $\|A\|_\infty = n$. Now, A^*A is an $n \times n$ matrix whose entries are all equal to one (check!) and its rank is 1. So 0 is an e-value of A^*A of multiplicity $n-1$. What is the remaining e-value? It is $\lambda_n = n$, which is easily seen to be the case since $x = (1, 1, \ldots, 1)^T \in \mathbb{C}^n$ is a corresponding e-vector (check!). Thus $\|A\|_2 = \sqrt{\rho(A^*A)} = \sqrt{n}$. So we have $\|A\|_\infty = n = \sqrt{n}\sqrt{n} = \sqrt{n}\|A\|_2$.

(d) We have

$$\begin{aligned} \|A\|_2 &= \sup_{x\neq 0} \frac{\|Ax\|_2}{\|x\|_2} \\ &\leq \sup_{x\neq 0} \frac{\sqrt{m}\|Ax\|_\infty}{\|x\|_2} && \text{using (b) in the numerator, and noting } Ax \in \mathbb{C}^m \\ &\leq \sup_{x\neq 0} \frac{\sqrt{m}\|Ax\|_\infty}{\|x\|_\infty} && \text{using (a) in the denominator} \\ &= \sqrt{m}\|A\|_\infty \end{aligned}$$

For equality, let $A \in \mathbb{C}^{m\times n}$ be the matrix whose first column is all ones, and zeros elsewhere. Clearly $\|A\|_\infty = 1$. Now, A^*A is an $n \times n$ diagonal matrix whose entries are all equal to zero, except for the $(1, 1)$ entry, which is equal to m (check!). Thus, $\|A\|_2 = \sqrt{\rho(A^*A)} = \sqrt{m}$. So we have $\|A\|_2 = \sqrt{m} = \sqrt{m} * 1 = \sqrt{m}\|A\|_\infty$.

4. Prove that given a vector norm $\|x\|$, the formula $\|A\| = \sup_{x\neq 0} \frac{\|Ax\|}{\|x\|}$ defines a matrix norm for a square matrix A. Recall, this is referred to as the *induced matrix norm.*

ANS: Suppose $A \in \mathbb{C}^{m\times m}$. We need to show:

(i) $\|A\| \geq 0$, and $\|A\| = 0$ only if $A = 0$.

$\|A\| \geq 0$ since it is the supremum of the ratio of $\|x\|, \|Ax\| \geq 0$. Now suppose $\|A\| = 0$ but $A \neq 0$. Since $A \neq 0$ there exists an $x \neq 0$ such that $Ax = y \neq 0$. For example, if $a_k \neq 0$ where a_k is the k^{th} column of A (and there must be one since $A \neq 0$) let $x = e_k$. Then $y = a_k$ and $\|Ax\|/\|x\| = \|y\|/\|x\| > 0$, hence so is the suprmum over all $x \neq 0$. Contradiction. So $A = 0$.

(ii) $\|A + B\| \leq \|A\| + \|B\|$.

We have $\|x + y\| \leq \|x\| + \|y\|$ for any $x, y \in \mathbb{C}^m$. Then

$$\begin{aligned} \|A+B\| &= \sup_{x\neq 0} \frac{\|(A+B)x\|}{\|x\|} = \sup_{x\neq 0} \frac{\|Ax+Bx\|}{\|x\|} \\ &\leq \sup_{x\neq 0} \frac{\|Ax\|+\|Bx\|}{\|x\|} \\ &= \sup_{x\neq 0} \frac{\|Ax\|}{\|x\|} + \sup_{x\neq 0} \frac{\|Bx\|}{\|x\|} \\ &= \|A\| + \|B\|. \end{aligned}$$

(iii) $\|\alpha A\| = |\alpha|\|A\|$.

We have $\|\alpha x\| = |\alpha|\|x\|$ for any $x \in \mathbb{C}^m$, $\alpha \in \mathbb{C}$. Then

$$\begin{aligned} \|\alpha A\| &= \sup_{x\neq 0} \frac{\|\alpha Ax\|}{\|x\|} = \sup_{x\neq 0} \frac{|\alpha|\|Ax\|}{\|x\|} \\ &= \sup_{x\neq 0} \frac{\alpha\|Ax\|}{\|x\|} = \alpha \sup_{x\neq 0} \frac{\|Ax\|}{\|x\|} \\ &= |\alpha|\|A\|. \end{aligned}$$

5. Prove that $\|A\|_\infty = \max_i \sum_j |a_{ij}|$, the maximum absolute row sum of the matrix A.

ANS: Assume $A \in \mathbb{C}^{m\times n}$ and $x \in \mathbb{C}^n$. Then

$$\|Ax\|_\infty = \max_i \; |(Ax)_i| = \max_i |\sum_j a_{ij}x_j| = \max_i \sum_j |a_{ij}||x_j| \leq \|x\|_\infty \max_i \sum_j |a_{ij}|.$$

So $\dfrac{\|Ax\|_\infty}{\|x\|_\infty} \leq \max_i \sum_j |a_{ij}|$ for all $x \neq 0$. Now let k be such that $\max_i \sum_j |a_{ij}| = \sum_j |a_{kj}|$. If there is more than one such k choose the minimum. We can then write $a_{kj} = r_j e^{i\theta_j}$ for some real number $r_j \geq 0$ and $0 \leq \theta_j < 2\pi$. Note $|a_{kj}| = |r_j e^{i\theta_j}| = |r_j||e^{i\theta_j}| = |r_j| = r_j$. Define $\tilde{x} \in \mathbb{C}^n$ by $\tilde{x}_j = e^{-i\theta_j}$ for $j = 1, \ldots, n$. Notice that is A were a real matrix then we would have $\tilde{x}_j = \pm 1 = e^{-i(0,\pi)}$. Then $\|\tilde{x}\|_\infty = 1$ and

$$|\sum_j a_{kj}\tilde{x}_j| = |\sum_j r_j e^{i\theta_j} e^{-i\theta_j}| = |\sum_j r_j| = \sum_j r_j = \sum_j |a_{kj}|.$$

Finally,

$$\|Ax\|_\infty = \sup_{x\neq 0} \frac{\|Ax\|_\infty}{\|x\|_\infty} \leq \max_i \sum_j |a_{ij}| = \sum_j |a_{kj}| = \frac{\|A\tilde{x}\|_\infty}{\|\tilde{x}\|_\infty} \leq \sup_{x\neq 0} \frac{\|Ax\|_\infty}{\|x\|_\infty} = \|Ax\|_\infty.$$

Thus, $\|Ax\|_\infty = \max_i \sum_j |a_{ij}|$, the maximum absolute row sum of A.

6. Consider the 2-point BVP

$$\begin{cases} -y'' + (4x^2+2)y = 2x(1+2x^2) \\ y(0) = 1, \ y(1) = 1+e \end{cases}$$

The exact solution is $y(x) = x + e^{x^2}$. Write a MATLAB function M-file to solve the problem using the **4th** order centered compact FD scheme

$$-D_+D_-\Big(1 - \frac{h^2}{12}c_i\Big)u_i + c_iu_i = \Big(1 + \frac{h^2}{12}D_+D_-\Big)f_i.$$

Use meshsize $h = 1/2^p$, where p is a positive integer. Your code should use your M-files **trilu** and **trilu_solve**. For $p = 1:4$, plot the exact solution ($y(x)$ vs. x) and the numerical solution (u_i vs. x_i), including the boundary points. The 4 plots should appear separately in one figure, with axes labeled and a title for each indicating p. Investigate **subplot** in MATLAB for how to have multiple plots in a single figure window. For $p = 1:20$ present a table with the following data - column 1: h; column 2: $\|u_h - y_h\|_\infty$; column 3: $\|u_h - y_h\|_\infty/h^4$; column 4: cpu time; column 5: (cpu time)/m, where $h = 1/(m+1)$. Discuss the trends in each column. Also, compare the accuracy for each h with the results of the 2nd order code from HW #1. How do the computational times compare? Include a copy of your code.

ANS: Writing out the discretization gives

$$\frac{-(1-\frac{h^2}{12}c_{i-1})u_{i-1} + (2+\frac{5h^2}{6}c_i)u_i - (1-\frac{h^2}{12}c_{i+1})u_{i+1}}{h^2} = f_i + \frac{(f_{i-1} - 2f_i + f_{i+1})}{12}.$$

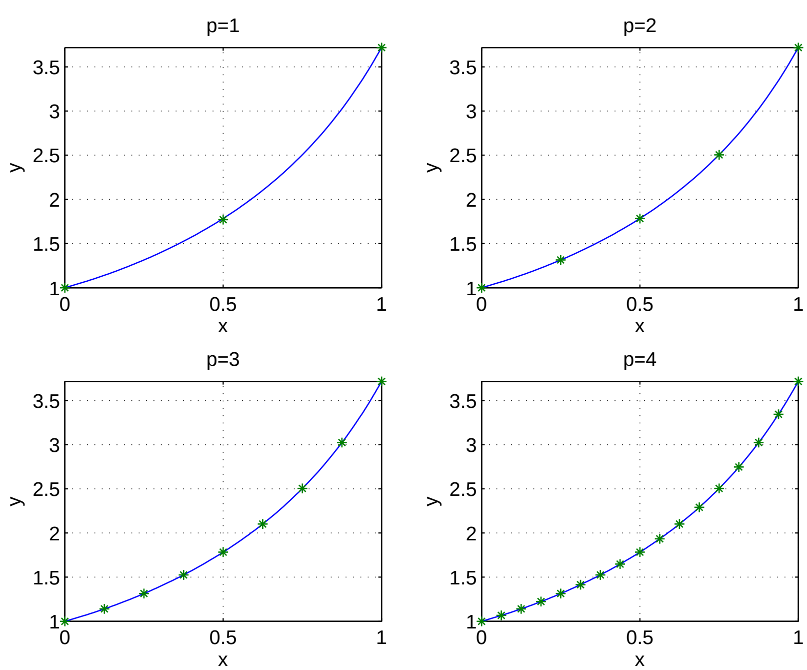
Here is the code for the first part of the problem, followed by a listing of the M-file function *bvp_solve4*. The latter requires the M-files *trilu* and *trilu_solve* from problem 7 (hw #1).

```
xx=0:0.01:1;xx=xx';
yy=xx+exp(xx.^2);
c='4.*x.^2+2'; f='2*x.*(1+2*x.^2)';
clf;
for p=1:4
    [x,u]=bvp_solve4(2^p-1,0,1,1,1+exp(1),c,f);
    max(abs(u-(x+exp(x.^2))))
    subplot(2,2,p),plot(xx,yy,x,u,'*'),grid
    axis('tight'),xlabel('x'),ylabel('y'),title(['p=',num2str(p)])
end
```

```
function [xv,uv,stime]=bvp_solve4(m,a,b,ya,yb,c,f)
%
h=(b-a)/(m+1);
xv=a:h:b; xv=xv';
fv=zeros(m+2,1);
cv=zeros(m+2,1);
```

```
%
x=xv;
cfv=eval(c);
fv=eval(f);
%
% setup tridiagonal matrix and rhs
%
av=(2+(5/6)*h^2*cfv(2:m+1))/(h^2);
bv=zeros(m,1); bv(2:m)  =-(1-(h^2/12)*cfv(2:m  ))/(h^2); bv(1)=0;
cv=zeros(m,1); cv(1:m-1)=-(1-(h^2/12)*cfv(3:m+1))/(h^2); cv(m)=0;
fv=fv(2:m+1)+(1/12)*(fv(1:m)-2*fv(2:m+1)+fv(3:m+2));
%
% BC adjustment
%
fv(1)=fv(1)+(1-(h^2/12)*cfv(1)  )*ya/(h*h);
fv(m)=fv(m)+(1-(h^2/12)*cfv(m+2))*yb/(h*h);
%
% solve tridiagonal system; collect cpu time
%
tic;
[alpha,beta]=trilu(av,bv,cv);
uv=trilu_solve(alpha,beta,cv,fv);
stime=toc;
uv=[ya;uv;yb];
```

Here is the graph for $p = 1, 2, 3$ and 4.



Next we solve the BVP with $m = 2^p - 1$ for $p = 1, \ldots, 20$. Here is the code:

```
c='4*x.^2+2'; f='2*x.*(1+2*x.^2)';
clf;
h=zeros(20,1); m=zeros(20,1);
times=zeros(20,1);
err_inf=zeros(20,1);
for p=1:20
    [x,u,stime]=bvp_solve4(2^p-1,0,1,1,1+exp(1),c,f);
    h(p)=1/(2^p);
    m(p)=2^p-1;
    times(p)=stime;
    y=x+exp(x.^2);
    err_inf(p)=max(abs(u-y));
end
format short e
disp(' ')
disp('       h        inf_err      err/h^4      cputime     cputime/m  ')
disp('  ---------------------------------------------------------------')
disp(' ')
disp([h err_inf err_inf./h.^4 times times./m])
```

The results are:

```
       h        inf_err      err/h^4      cputime     cputime/m
  ---------------------------------------------------------------

   5.0000e-01   1.2852e-02   2.0563e-01   9.2100e-04   9.2100e-04
   2.5000e-01   9.7605e-04   2.4987e-01   1.9500e-04   6.5000e-05
   1.2500e-01   6.3927e-05   2.6185e-01   1.3600e-04   1.9429e-05
   6.2500e-02   4.0888e-06   2.6796e-01   1.3100e-04   8.7333e-06
   3.1250e-02   2.5615e-07   2.6859e-01   1.5300e-04   4.9355e-06
   1.5625e-02   1.6019e-08   2.6875e-01   2.1600e-04   3.4286e-06
   7.8125e-03   1.0013e-09   2.6878e-01   2.5800e-03   2.0315e-05
   3.9062e-03   6.1728e-11   2.6512e-01   6.3200e-04   2.4784e-06
   1.9531e-03   1.0814e-12   7.4310e-02   1.0260e-03   2.0078e-06
   9.7656e-04   1.3562e-11   1.4911e+01   2.2850e-03   2.2336e-06
   4.8828e-04   5.5336e-11   9.7347e+02   4.4010e-03   2.1500e-06
   2.4414e-04   2.1187e-10   5.9635e+04   8.8300e-03   2.1563e-06
   1.2207e-04   8.7924e-10   3.9597e+06   1.6935e-02   2.0675e-06
   6.1035e-05   1.2362e-09   8.9078e+07   3.5043e-02   2.1390e-06
   3.0518e-05   2.3403e-09   2.6982e+09   6.8585e-02   2.0931e-06
   1.5259e-05   1.3832e-09   2.5516e+10   1.3560e-01   2.0692e-06
   7.6294e-06   1.3652e-08   4.0293e+12   2.6766e-01   2.0421e-06
   3.8147e-06   4.3051e-07   2.0330e+15   5.3457e-01   2.0392e-06
```

1.9073e-06	1.3555e-07	1.0242e+16	1.0638e+00	2.0291e-06
9.5367e-07	7.0937e-07	8.5757e+17	2.1305e+00	2.0318e-06

We can see from the err/h^4 column that the expected $O(h^4)$ error is observed until $h \approx 3.9062e-3$ since err/h^4 rapidly approaches a constant. But then we lose accuracy. Why? Roundoff error begins to dominate, as well as the conditioning of the matrix! Note, however, that the cputime stills scales linearly with m as evidenced by the $cputime/m$ column.

Comparing with the second order method of HW #1 we see that the cpu timeS for the solvers are close. However, for moderate p the fourth order solution is clearly superior. For example, the error is $6.1728e-1$ for $h = 3.9062e-03$, while for the second order method with the same h the error is $5.4521e-06$.

Note: I have only timed the linear solver portion of the code.