CS/SE 4X03 — Assignment 2

Ned Nedialkov

31 January 2020

Due date: 11 February, in class.

> **A hardcopy must be submitted before 1:20pm.**
>
> **Between 1:20pm and 1:20pm on February 14th, an assignment will be accepted only with an MSAF.**

Problem 1 (8 points) Implement in Matlab the following functions (2 points each). In the first three functions, you are not allowed to use the **lu** function and the backslash operator.

```
function [L, U, P] = lu_pp(A)
%[L,U,P] = lu_pp(A) computes a unit lower triangular matrix L,
%an upper triangular matrix U, and a permutation matrix P such
   that
%P*A = L*U.
%The LU factorization is computed using partial pivoting.
```

```
function [L, U, P] = lu_spp(A)
%[L,U,P] = lu_spp(A) computes a unit lower triangular matrix L,
%an upper triangular matrix U, and a permutation matrix P such
   that
%P*A = L*U.
%The LU factorization is computed using scaled partial pivoting
   .
```

```
function x = lu_solve(L,U,b)
%Solves LUx=b by solving Ly=b and then Ux=y.
%L is unit lower triangular, U is upper triangular
```

```
function [xm, xpp, xspp] = solve_all(A,b)
%Solves a linear system A*x = b using Matlab's lu and
%lu_pp and lu_spp.
%Returns
%xm = A\b
%xpp solution computed using lu_pp and lu_solve
%xspp solution computed using lu_spp and lu_sovle
```

For this problem, we will investigate the errors in the computed solutions by `solve_all`. As a reference solution, we use a vector of ones. That is, for a given matrix A, if $x = (1, 1, \ldots, 1)^T$, we compute $b = A * x$ and then solve $Ax = b$. If $\widetilde{x}$ is the computed solution, the error is $\|x - \widetilde{x}\|$ and the residual is $\|b - A\widetilde{x}\|$.

To produce numerical results run the script `main_linear.m` (see Avenue).
(10 points) Discuss the accuracy of the computed solutions. In particular:

- How does it relate to the condition number of the matrix and the residual?
- How do your results (with partial and scaled partial pivoting) compare to Matlab's?
- Gauss elimination with partial pivoting usually produces small residuals. If some residuals are not very small, can you explain why?
- Can scaled partial pivoting improve the accuracy of the computed solution?

Submit also

- Avenue: the files `lu_pp.m`, `lu_spp.m`, `lu_solve.m`, `solve_all.m` containing the corresponding functions.
- Hardcopy: `lu_pp.m`, `lu_spp.m`, `lu_solve.m`, `solve_all.m`, `linearsolve.dat` and your discussion.

Problem 2 (5 points) The **lu** function of Matlab does partial pivoting, but we don't know if it does scaled partial pivoting. Can you construct a numerical experiment from which you can conclude if it does such pivoting?

Problem 3 (4 points) Consider the system $Ax = b$, where

$$A = \begin{bmatrix} 0.1 & 0.2 & 0.3 \\ 0.4 & 0.5 & 0.6 \\ 0.7 & 0.8 & 0.9 \end{bmatrix}$$

and $b = [0.1, 0.3, 0.5]^T$.

(a) (1 point) Show that A is singular.

(b) (1 point) If we were to use Gaussian elimination with partial pivoting to solve this system using exact arithmetic, show where the process fails.

(c) (2 points) Although A is singular, if we solve $Ax = b$ in Matlab using the backslash operator, `A\b`, we still find a solution. Explain how this is possible. How accurate is it?

Problem 4 (4 points) Given the data points

x	0	2	3	4
y	7	11	28	63

write the interpolating polynomials using (a) Lagrange and (b) Newton basis.

Problem 5 (4 points)

(a) (2 points) Prove by induction that the nth, $n \geq 1$, derivative of $\sqrt{x}$ is

$$(\sqrt{x})^{(n)} = (-1)^{n-1}\frac{1}{2^n}(2n-3)!! \cdot x^{-n+1/2}, \tag{1}$$

where $(2n-3)!! = 1$ if $n = 0, 1$ and $(2n-3)!! = 1 \cdot 3 \cdot 5 \cdot (2n-3)$ otherwise.

Let $y_i = \sqrt{x_i}$, where $x_i = 1 + 0.25i$, $i = 0, 1, 2, 3, 4$. Suppose you interpolate $\{(x_i, y_i)\}_{i=0}^{4}$ by a polynomial of degree 4. Denote this polynomial by $p_4(x)$.

(b) (1 point) Derive a bound for the error $|p_4(x) - \sqrt{x}|$ in this interpolation.

(c) (1 point) In Matlab, use **semilogy** to plot $|p_4(x) - \sqrt{x}|$ and this bound versus $x \in [1, 2]$. You can use e.g. 100 evenly spaced points to evaluate $|p_4(x) - \sqrt{x}|$.

Submit your plot.

Problem 6 (4 points) Assume that your computer (or calculator) cannot compute the cosine function, and that you need to approximate $\cos(\pi/6)$.

(a) (1 point) We know that $\cos(0) = 1$, $\cos(\pi/4) = \sqrt{2}/2$, and $\cos(\pi/2) = 0$. Using this information, calculate an approximation for $\cos(\pi/6)$. Do not use Matlab's **polyfit**.

(b) (2 points) Derive a bound for the error in this approximation. How does it compare to the actual error?

(c) (1 point) Without calculating the value for $\cos(\pi/6)$, determine how accurate your approximation is.

Problem 7 (4 points) Consider $f(x) = \max(0, \sin(x))$ on $[-\pi, \pi]$.

(a) (1 point) Interpolate $f(x)$ at 21 evenly spaced points x_i in $[-\pi, \pi]$, $i = 0, \ldots 20$. You can use the **polyfit** function. Denote the resulting interpolation polynomial by $p(x)$. Plot on the same plot $f(x)$ and $p(x)$ at 200 evenly spaced points in $[-\pi, \pi]$.

(b) (1 point) Instead of **polyfit** use **spline** and produce a plot as in (a).

(c) (1 point) Instead of the above points x_i, now use 21 Chebyshev points. Use **polyfit** and produce a plot as in (a).

(d) (1 point) Explain the differences in the plots in (a) and (c).

For reference, my plots for $x^2/(1+x^2)$ function are shown in Figure 1. Produce similar plots.

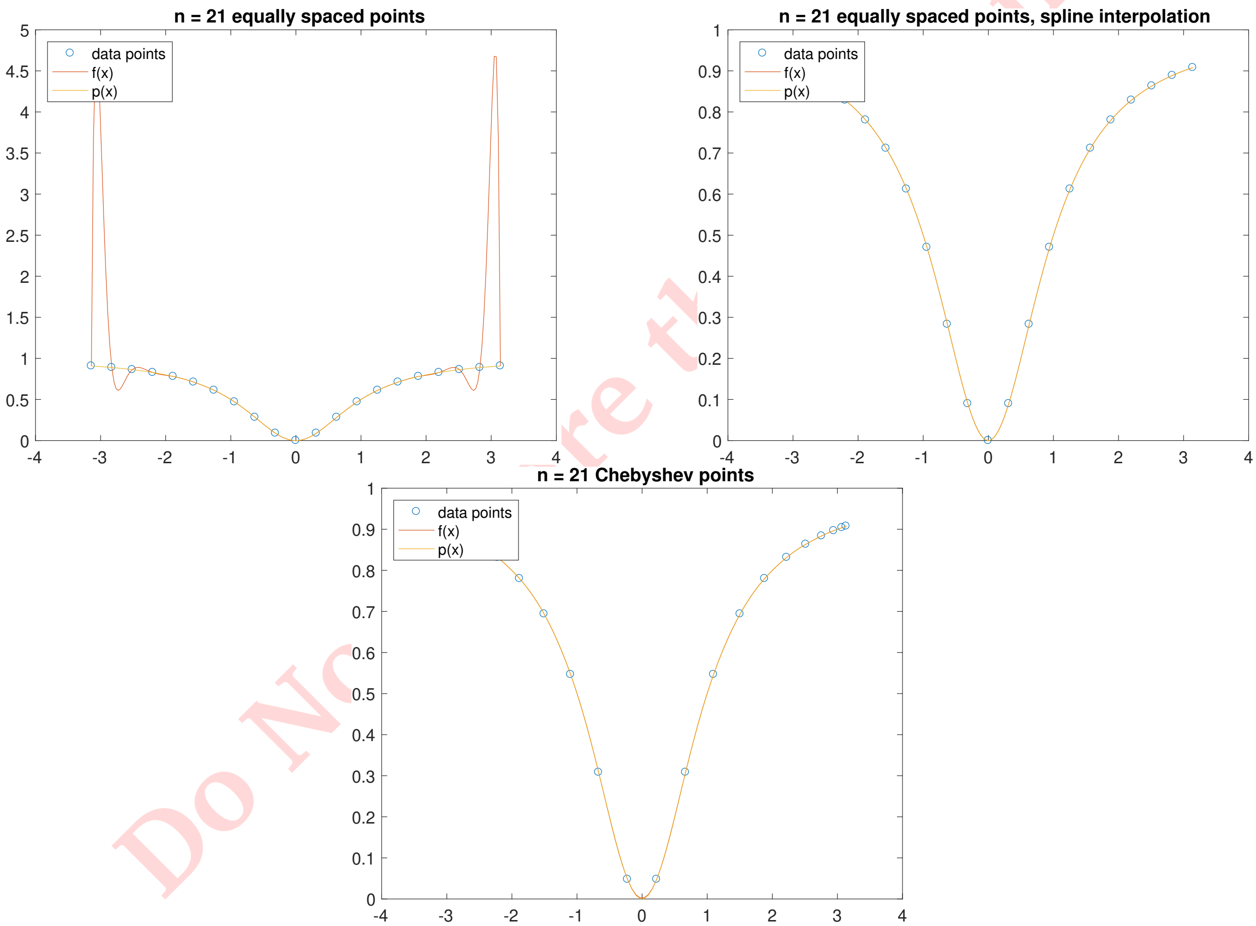

Figure 1: Interpolation plots.

.