

Math 304 (Spring 2010)

Study Guide for Weeks 2-3

Before attempting homework 2 please make sure that you have understood the topics listed below. This homework concerns Gaussian elimination, LU factorization and Gaussian elimination with partial (or row) pivoting.

- Augmented matrix (Fausett - 3.1, page 99; Watkins - 1.7, page 71)
- Three row operations and the fact that they preserve the solution set of a linear system when applied to the associated augmented matrix (Watkins - 1.7, pages 70-71)
- The row-reduction of the coefficient matrix into an upper triangular form (this process is called *Gaussian elimination*, Fausett 3.1.1, pages 98-106; Watkins - 1.7, pages 71-76)
- Computation of an LU factorization assuming no row-interchange is needed during the row-reduction process (Fausett 4.1.1 pages 138-139; Watkins - 1.7, pages 76-83)
- How to use the LU factorization to solve a linear system (see question 3 below, Fausett 4.1.3, pages 150-151, Watkins - 1.7, page 79)
- Gaussian elimination with partial (or row) pivoting (Fausett 3.1.2, pages 107-111; Watkins - 1.8, pages 93-97)

Homework 2 (due on March 12th, Friday by 4pm)

In Matlab question (questions 3) attach the Matlab outputs and print-outs of the Matlab routines that you implemented.

1. Consider the linear systems

$$\underbrace{\begin{bmatrix} -2 & -2 & -2 \\ -2 & -4 & -4 \\ -2 & -4 & -6 \end{bmatrix}}_{A_1} x = \underbrace{\begin{bmatrix} 0 \\ 4 \\ 6 \end{bmatrix}}_{b_1} \quad \text{and} \quad \underbrace{\begin{bmatrix} 2 & 3 & 0 \\ -4 & -4 & 3 \\ 0 & -4 & -4 \end{bmatrix}}_{A_2} x = \underbrace{\begin{bmatrix} -5 \\ 7 \\ 8 \end{bmatrix}}_{b_2}.$$

- Compute the LU factorizations of A_1 and A_2 by hand.
- Exploiting the LU factorizations from part (a) solve the linear systems $A_1x = b_1$ and $A_2x = b_2$. (Note : you should only solve triangular system; more precisely first a lower triangular system by forward substitution followed by an upper triangular system by back substitution. See question 2 below for details.)
- Compute the determinants of A_1 and A_2 by exploiting their LU factorizations.

2. Your doctor recommends that your ideal daily fat, energy and protein intakes should be 60 grams, 2.2 Kcal (kilocalories) and 100 grams, respectively. Your diet consists of three

types of food; broccoli, kidney beans, sirloin steak. The amounts of fat, energy and protein per serving for each type of food are provided in the table below.

Food	Fat (grams/serving)	Energy(Kcal/serving)	Protein(grams/serving)
broccoli	1	0.02	5
kidney beans	3	0.06	10
sirloin steak	50	2	60

- (a) Set up a system of 3 linear equations in 3 three unknowns. Your variables are x_1 , x_2 , x_3 denoting the number of daily servings of broccoli, kidney beans and sirloin steak, respectively.
- (b) Solve the system in part (a) by Gaussian elimination with partial pivoting by hand. (Note : You cannot use the row-scaling operation.)

3. Given a non-singular $n \times n$ dense matrix A (dense in this context means most of the entries, for instance more than half of the entries, of A are nonzero). The standard approach to solve the linear system $Ax = b$ is as follows.

1. Compute the LU factorization $A = LU$.
2. Let $\hat{x} := Ux$. Solve the lower triangular system $L\hat{x} = b$ by forward substitution.
3. Solve the upper triangular system $Ux = \hat{x}$ by back substitution.

- (a) Write a Matlab function to compute the LU factorization of A . Assume that A can be reduced into an upper triangular matrix U by only applying row-replacement operations. Your function must take a matrix A as input and return a lower triangular matrix L and an upper triangular matrix U . Your Matlab code should look like

```
function [L,U] = lu_factor(A)

[n,n1] = size(A);

if n ~= n1
    error('A must be square');
end

L = eye(n);
U = zeros(n,n);

....

return;
```

In the above code you need to fill in the part in between the statements $U = \text{zeros}(n,n)$; and return ; The statement $U = \text{zeros}(n,n)$; initializes U to the $n \times n$ zero matrix.

You only need to modify the entries of U above and on the main diagonal, that is $U(i, j)$, $i \leq j$. The statement `L = eye(n)`; sets L to the $n \times n$ identity matrix initially. You need to modify only the portion of this matrix below the main diagonal, that is $L(i, j)$, $i > j$.

- (b) Write another Matlab function to solve the system $Ax = b$ following steps 1, 2, 3 described above. Here is how your function should look like

```
function x = linear_sys_solver(A,b)

% compute the LU factorization of A
% Solve Ly = b for y by forward substitution
% Solve Ux = y by back substitution

return;
```

You should use the routines from homework 1 for back substitution and forward substitution.

- (c) In homework 1 - question 4. you solved either (**Exercise 1.2.19**) or (**Exercise 1.2.20**) from Watkin's book. For one of these exercises, whichever one you worked on previously, solve the resulting linear system in part (b) of the exercise by typing `x = linear_sys_solver(A,b)`; in Matlab. Compare this solution with the solution returned by the built-in linear system solver, which you can obtain by typing `x = A\b`.
- (d) Implement a Matlab routine `matinverse` to compute the inverse of an $n \times n$ matrix A . You should use the routines `lu_factor` (that you implemented in part (a)), `forwardsubstitute` and `backsubstitute` (from the previous homework). Suppose $X = A^{-1}$ so that $AX = I_n$. You should exploit the fact that

$$Ax_i = e_i \quad i = 1, \dots, n$$

where x_i is the i th column of the inverse X and e_i the i th column of the $n \times n$ matrix I_n . (Note : Perhaps you should compute the LU factorization of A only once.)

- (e) Compute the inverses of A_1 and A_2 given in question 1 by using the matlab routine `matinverse` that you implemented in part (d).