

CS3414. Homework problem set VI. Each problem is worth 10 points unless otherwise specified.

C&K = *Cheney and Kincaid* textbook.

1. C&K 7d, page 276.
2. C&K 9, page 293.
3. C&K 5, top of page 307.
4. C&K 18, page 336. May use Mathematica. Give U and L separately. If $\epsilon_{mach} = 10^{-16}$ how many correct digits you can expect in the solution of $A\vec{x} = \vec{b}$?

5. (30 points). In this exercise you will test and benchmark different codes for solving systems of linear equations on your computer. Use of Mathematica ONLY will result in ZERO points. Specify CPU, Memory, and Compiler you use. Justify your choice of a given method.

a). Define a “random” $n \times n$ matrix \mathbf{A} where each a_{ij} is a random number from 0.5 to 1.5. The size n will vary.

b). Solve $\mathbf{A}\vec{x} = \vec{b}$, $b_i = 1$, $i = 1, n$. Set $n = 10$ and use **two** different methods, only one of which may be *Mathematica*. The other one (or both) must come from either of the following: GSL, NetLib (“GAMS” portal may help you find the right algorithm) or *Numerical Recipes*.

c). Determine maximum possible n with which you can solve $\mathbf{A}\vec{x} = \vec{b}$ on your machine within 1 week. DO NOT actually run your machine for a week, but run the codes for $n = 2, 4, 8, 16, \dots$ and determine $Time(n)$, $MemoryUsed(n)$. Extrapolate to larger n , that is find the best polynomial fit (example: assume $Time = Kn^3$ [seconds], and get K . Your own case may be different). Plot the results. For fits and plots you MAY use Mathematica. Find out whether it is Memory or CPU that is the bottleneck on your system. Use same algorithms as in part b). Prepare a benchmark table similar to what you did in HW 3.

d) Same as c), but for a “random” *symmetric* ($a_{ij} = a_{ji}$) *tri-diagonal* matrix. At least one of the methods you use here must be specifically tailored for tri-diagonal systems. What is the bottle-neck now?