

Homework 11

Please answer the following questions in complete sentences in a clearly prepared manuscript and submit the solution by the due date on Blackboard (around Sunday, December 2nd, 2018.) Note that there will be a full-points extension opportunity.

Remember that this is a graduate class. There may be elements of the problem statements that require you to fill in appropriate assumptions. You are also responsible for determining what evidence to include. An answer alone is rarely sufficient, but neither is an overly verbose description required. Use your judgement to focus your discussion on the most interesting pieces. The answer to “should I include ‘something’ in my solution?” will almost always be: Yes, if you think it helps support your answer.

Problem 0: Homework checklist

- Please identify anyone, whether or not they are in the class, with whom you discussed your homework. This problem is worth 1 point, but on a multiplicative scale.
- Make sure you have included your source-code and prepared your solution according to the most recent Piazza note on homework submissions.

Problem 1: Using GMRES with Preconditioners

For this problem, you must use Julia.

It’s based on an assignment from Michael Saunders’ iterative methods class.

Download `poisson2D-data.csv` and `poisson2D-rhs.csv` and make sure you can load them via the following script. These files were originally MAT files from <https://sparse.tamu.edu/mat/FEMLAB/>

```
using DelimitedFiles
using SparseArrays
data = readdlm("poisson2D-data.csv",',')
A = sparse(Int.(data[:,1]), Int.(data[:,2]),(data[:,3]))
A = (A + A') ./ 2
b = vec(readdlm("poisson2D-rhs.csv"))
```

The matrix A is symmetric-positive definite. But we are going to look at solving $(A - \sigma I)\mathbf{x} = \mathbf{b}$ where $\sigma = 1.7 \times 10^{-2}$. We’ll declare a method converged if the relative residual has 2-norm less than 10^{-6} .

1. Plot the relative residuals for 100 iterations of the MINRES (`minres` from `IterativeSolvers.jl`) method and the GMRES (`gmres` from `IterativeSolvers.jl`) method restarted every 30 iterations on a semilogy

scale. Describe what you observe, in particular, do the methods converge?
For convenience, here is some code to help you get started:

```
using IterativeSolvers
x, hist_min = minres(A,b,tol = mytol, maxiter = mymaxiter, log=true)
x, hist_gm = gmres(A,b,restart = myrestart, tol = mytol, maxiter = mymaxiter, log=true)
# look at hist_min.data[:resnorm]
```

2. If you use an incomplete cholesky `ichol` or incomplete LU `ilu` preconditioner with GMRES, it will converge. How many iterations does it take? Does MINRES benefit from an incomplete Cholesky preconditioner? We provide you with the `ichol` and `ilu` code. Download the file `hw11_functions.jl` from here http://www.cs.purdue.edu/homes/dgleich/cs515-2018/homeworks/hw11_functions.jl. The `IterativeSolvers` package still has some rough edges. We will help you set that up. Here is the code to help you get started

```
using IterativeSolvers
include("hw11_functions.jl")
Lchol = ichol(A)
M1 = LowerPrecond(ichol(A))
x, hist_pgm = gmres(A,b,restart = myrestart, tol = mytol, maxiter = mymaxiter,P1=M, log=true)
```

For MINRES, it turns out that the `minres` function in `IterativeSolvers` still doesn't accept preconditioners. We use another package to test `minres`.

```
using Krylov # ] clone https://github.com/JuliaSmoothOptimizers/Krylov.jl.git
using LinearOperators
P1 = opInverse(LowerTriangular(ichol(A)))
Mprecond = P1'*P1;
x, hist_pmin = Krylov.minres(A,b,M=Mprecond,itmax=mymaxiter, rtol=mytol)
```

3. Show the eigenvalues of the matrix before and after preconditioning. Do you observe any clustering?

Problem 2: Using Multigrid to solve Poisson's equation

For this problem, if you don't use Julia, you will have to convert a series of multigrid codes to your own language. This isn't too hard as it's only about 100 lines of code. However, in one step, you will have to use a sparse direct solver. This is simple to do in Julia. If you know how to do it in other languages, then this should be easy. However, you are on your own!

Download the codes http://www.cs.purdue.edu/homes/dgleich/cs515-2018/homeworks/multigrid_functions.jl

1. Using these codes, you can solve Poisson's equation as follows:

```
X = solve_poisson_direct(poisson_setup(nx,ny, (x,y) -> 1))
```

and plot a solution via

```
using Plots
pyplot()
surface(X)
```

(which is what a previous homework asked you to do!)

For this problem, we will always have $n = nx = ny$. How long does it take to solve this for $n = 31, 63, 127, 255, 511, 1023$. (This worked great on my computer in terms of memory usage. But your mileage may vary, if you run out of memory, show the largest problem you can solve!)

Plot the times on a log-log scale. If you double the problem size, about how much longer does the computation need?

2. Explain what the `apply_poisson_jacobi` function does. Where is the matrix?
3. Use this method to implement the Jacobi method to solve a linear system? For a problem with $n = 31$, estimate how much decrease in error occurs at each iteration of Jacobi. That is, if $\mathbf{x}^{(k)}$ and $\mathbf{x}^{(k+1)}$ are successive iterates with error $e_k = \|\mathbf{x}^{(k)} - \mathbf{x}\|/\|\mathbf{x}\|$ and e_{k+1} , respectively, then what does the trend in e_{k+1}/e_k show? (Note, you probably have to run a few hundred to a few thousand iterations to see these values start to converge.)
4. Write a function `apply_poisson_gauss_seidel` to implement the Gauss-Seidel without building the matrix. Show the same error trend for Gauss-Seidel compared with Jacobi.
5. Estimate the same rate for Jacobi and Gauss Seidel for a problem with $n = 63$. Given that each iteration of Jacobi and Gauss-Seidel takes $O(n^2)$ time, give a rough estimate of the number of iterations needed to reduce the error to a fixed value of ε .
6. Now we get to play around with Multigrid! The function `interpolate` takes a $n = 31$ vector and returns a $n = 63$ vector. Use `interpolate` to evaluate the error that results from solving a $n = 31$ Poisson problem and interpolating the solution to a $n = 63$ solution. Show a mesh-plot of the absolute value of the error and the norm of the error.
7. What happens when you run a Jacobi iteration after interpolating the solution? Describe how the error changes in terms of the mesh-plot as well as the the norm of the error.
8. Explain what the `simple_multigrid` function does. Estimate the decrease in error for one iteration of the loop for $n = 31$ and $n = 63$.
9. Explain what the `apply_poisson_multigrid` function does and how `poisson_multigrid_error` compares with `simple_multigrid`. Estimate the decrease in error for one iteration of the loop for the same values of n .
10. In practice, we can't use the error as we won't have the true solution. Use the `poisson_multigrid_residual` method to solve Poissons equation with 25 iterations for $n = 31, 63, 127, 255, 511, 1023$. Show the times on a log-log scale along with the times from the direct method in part 1. Can you solve a 10,000-by-10,000 problem? (This is a big problem, you'll need O(16GB) of RAM!)

Problem 3: Eigenvalue estimates.

1. Use the Lanczos method to estimate the top 5 singular values of the eigenfaces matrix (Homework 7, problem 3). How many iterations do you need to get a good estimate. Compare this with the power method for the dominant vector. (Note that you do not need to be concerned with the vectors for this part.)
2. Does the loss of orthogonality of the Lanczos vectors impact the accuracy of the eigenvectors? Describe your investigation here. (A rough guide, you should present evidence that shows you are familiar with the ideas in class. This could be 1 paragraph up to around a page, and could include figures.)