

ECE 133A HW 6

Lawrence Liu

November 19, 2022

Exercise A9.4

Since $A = T(B)$, $D_v = T(E)$, and $D_h = T(E^T)$, and letting b , e , and f being the column major ordering of B , E and E^T . We have that since W is symmetric, \tilde{W} is symmetric. and thus we have that

$$A^T = \frac{1}{n^2} \tilde{W} \mathbf{diag}(\tilde{W}b) \tilde{W}^H$$

And the same for D_v^T and D_h^T . Thus we have that Thus we get that

$$A^T A = \tilde{W} \left(\mathbf{diag}(\tilde{W}b) \right)^2 \tilde{W}^H x$$

And likewise for $D_v^T D_v$ and $D_h^T D_h$. Thus we have that

$$(A^T A + \lambda D_v^T D_v + \lambda D_h^T D_h) x = A^T y$$

becomes:

$$\begin{aligned} \tilde{W} \left(\left(\mathbf{diag}(\tilde{W}b) \right)^2 + \lambda \left(\mathbf{diag}(\tilde{W}e) \right)^2 + \lambda \left(\mathbf{diag}(\tilde{W}f) \right)^2 \right) \tilde{W}^H x &= \frac{1}{n^2} \tilde{W} \mathbf{diag}(\tilde{W}b) \tilde{W}^H y \\ \left(\left(\mathbf{diag}(\tilde{W}b) \right)^2 + \lambda \left(\mathbf{diag}(\tilde{W}e) \right)^2 + \lambda \left(\mathbf{diag}(\tilde{W}f) \right)^2 \right) \tilde{W}^H x &= \frac{1}{n^2} \mathbf{diag}(\tilde{W}b) \tilde{W}^H y \end{aligned}$$

let $z = \tilde{W}^H x$ Then we get

$$\left(\left(\mathbf{diag}(\tilde{W}b) \right)^2 + \lambda \left(\mathbf{diag}(\tilde{W}e) \right)^2 + \lambda \left(\mathbf{diag}(\tilde{W}f) \right)^2 \right) z = \frac{1}{n^2} \mathbf{diag}(\tilde{W}b) \tilde{W}^H y$$

Then solving for $\tilde{W}b$, $\tilde{W}e$, and $\tilde{W}f$ and $\frac{1}{n^2} \tilde{W}^H y$ will cost $n^2 \log(n)$ flops each,

and then solving for $\left(\left(\mathbf{diag}(\tilde{W}b) \right)^2 + \lambda \left(\mathbf{diag}(\tilde{W}e) \right)^2 + \lambda \left(\mathbf{diag}(\tilde{W}f) \right)^2 \right)$

will cost us $7n$ flops. Likewise solving for $\frac{1}{n^2} \mathbf{diag}(\tilde{W}b) \tilde{W}^H y$ is just element wise multiplication of $\frac{1}{n^2} \tilde{W}^H y$ and $\tilde{W}b$ which is n flops. Then we can solve for

z by dividing $\frac{1}{n^2} \mathbf{diag}(\tilde{W}b) \tilde{W}^H y$ by $\left(\left(\mathbf{diag}(\tilde{W}b) \right)^2 + \lambda \left(\mathbf{diag}(\tilde{W}e) \right)^2 + \lambda \left(\mathbf{diag}(\tilde{W}f) \right)^2 \right)$

element wise which will cost us n flops. Then we can solve for x by just multiplying z with $\frac{1}{n^2} \tilde{W}$ or in other words, doing the FFT on z and then multiplying each value by $\frac{1}{n^2}$, which will cost us $n^2 \log(n) + n + 1$ flops. Therefore in total our algorithm will cost us $5n^2 \log(n) + 10n + 1$ flops.

Exercise A10.1

(a)

Let $u = [u(0), u(1), \dots, u(N-1)]^T$, then we have that $x = u$ we want to minimize the energy or

$$\|u\|^2$$

Furthermore we can express

$$s_1(N) = 0.1u(N-2) + (0.95+1) \cdot 0.1u(N-3) + (0.95^2+0.95+1) \cdot 0.1u(N-4) + \dots + \left(\sum_{i=0}^{N-2} 0.95^i \right) \cdot 0.1u(0)$$

$$s_2(N) = 0.1u(N-1) + (0.95) \cdot 0.1u(N-2) + (0.95^2) \cdot 0.1u(N-3) + \dots + (0.95^{N-1}) \cdot 0.1u(0)$$

So therefore we have that

$$C = \begin{bmatrix} 0.1 \sum_{i=0}^{N-2} 0.95^i & \dots & 0.1(0.95+1) & 0.1 & 0 \\ 0.1(0.95^{N-1}) & \dots & 0.1(0.95^2) & 0.1(0.95) & 0.1 \end{bmatrix}$$

And

$$d = \begin{bmatrix} 10 \\ 0 \end{bmatrix}$$

(b)

using PyPlot

```
function create_C(N)
    C = zeros(2,N)

    #make the first row of C
    for i=1:N-1
        for j=1:N-i
            C[1,i]+=0.1*(0.95^(j-1))
        end
    end
    # C[1,N-1]=0.1

    #make the second row of C
    for i=1:N
        C[2,i] = 0.1*(0.95^(N-i))
    end
    return C
end

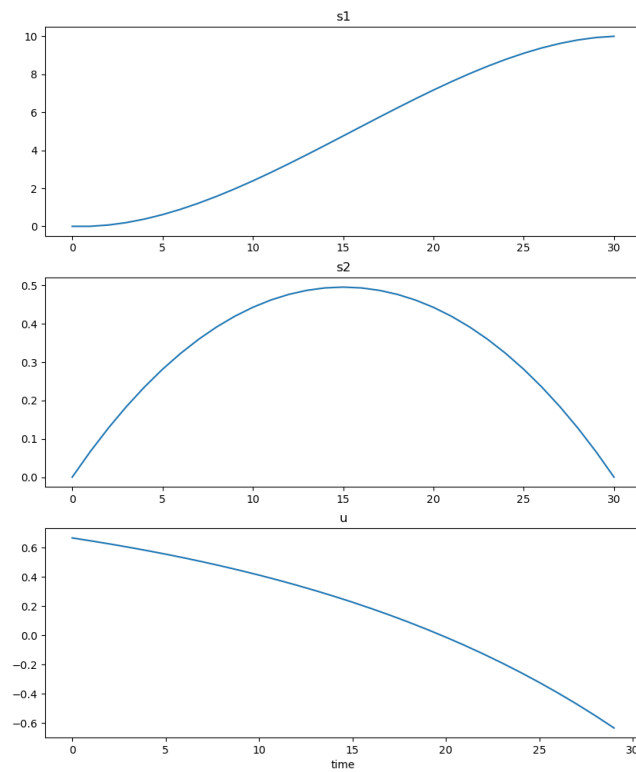
# calculate u
N=30
C = create_C(N)
d=[10,0]
u = C\d
s1=zeros(N+1)
s2=zeros(N+1)
for i=1:N
    s1[i+1]=s1[i]+s2[i]
    s2[i+1]=0.95*s2[i]+0.1*u[i]
```

end

```
fig,axs=subplots(3,1,figsize=(10,12))
axs[1].plot(s1)
axs[1][:set_title]("s1")
axs[2][:plot](s2)
axs[2][:set_title]("s2")
axs[3][:plot](u)
axs[3][:set_title]("u")
axs[3][:set_xlabel]("time")

savefig("problem2a.png")
close()
```

We get the following plot



(c)

With the following code:

using PyPlot

```
function create_C(N)
    C = zeros(2,N)
```

```
    #make the first row of C
```

```

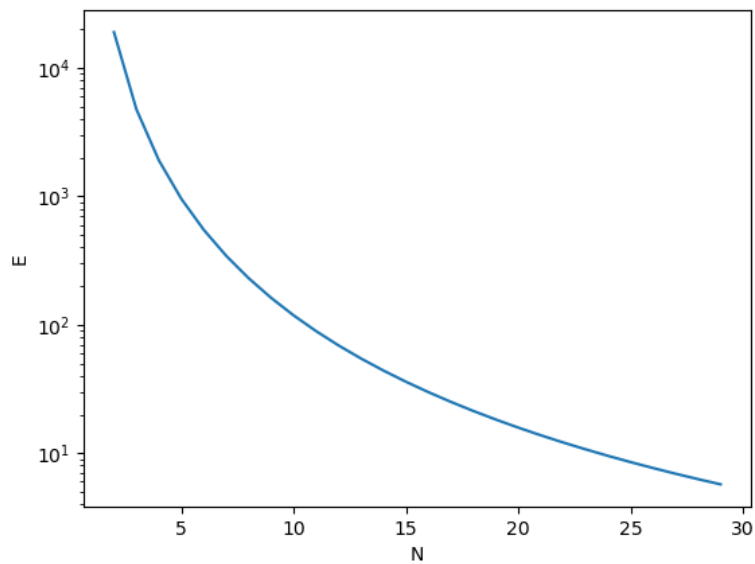
        for i=1:N-1
            for j=1:N-i
                C[1,i]+=0.1*(0.95^(j-1))
            end
        end
        # C[1,N-1]=0.1

        #make the second row of C
        for i=1:N
            C[2,i] = 0.1*(0.95^(N-i))
        end
        return C
    end

N=2:29
println(N)
E=zeros(length(N))
for i=1:length(N)
    C = create_C(N[i])
    d=[10,0]
    u = C\d
    E[i]=sum(u.^2)
end
plot(N,E)
xlabel("N")
ylabel("E")
yscale("log")
savefig("problem2b.png")
close()

```

We get the following plot



Exercise A10.9

(a)

We use Lagrange multipliers to solve this problem.

$$L(x) = \|Ax - b\|^2 + \lambda e_i^T x$$

$$\begin{aligned}
\nabla L(x) &= 0 \\
2A^T(Ax - b) + \lambda e_i &= 0 \\
2A^T Ax &= 2A^T b - \lambda e_i \\
A^T Ax &= A^T b - \frac{\lambda}{2} e_i \\
x &= (A^T A)^{-1} A^T b - \frac{\lambda}{2} (A^T A)^{-1} e_i \\
x &= \hat{x} - \frac{\lambda}{2} (A^T A)^{-1} e_i
\end{aligned}$$

substituting this back into the constraint we get

$$\begin{aligned}
e_i^T x &= 0 \\
e_i^T \left(\hat{x} - \frac{\lambda}{2} (A^T A)^{-1} e_i \right) &= 0 \\
\hat{x}_i - \frac{\lambda}{2} (A^T A)_{ii}^{-1} &= 0 \\
\hat{x}_i &= \frac{\lambda}{2} (A^T A)_{ii}^{-1} \lambda = \frac{2\hat{x}_i}{(A^T A)_{ii}^{-1}}
\end{aligned}$$

And thus we get that

$$x = \hat{x} - \frac{\hat{x}_i}{(A^T A)_{ii}^{-1}} (A^T A)^{-1} e_i$$

(b)

Calculating the QR factorization of A costs $2mn^2$ flops. And solving \hat{x} costs an additional $2mn + n^2$ flops. Then we can solve for $(A^T A)^{-1} e_i$ using backwards and forwards substitution which costs $2n^2$ flops, and then finding $(A^T A)_{ii}^{-1}$ is just finding the value for the i th index in the vector $(A^T A)^{-1} e_i$ which costs 0 flops. This is the same for \hat{x}_i . Then calculating $\frac{\hat{x}_i}{(A^T A)_{ii}^{-1}}$ costs 1 flop, and then multiplying that to every value of $(A^T A)^{-1} e_i$ costs n flops. And then subtracting the resulting vector from \hat{x} will cost n flops. So the total cost is $\boxed{2mn^2 + 2mn + 3n^2 + 2n + 1}$ flops.