CS205 Homework #7 Solutions

Problem 1

We have seen the application of the conjugate gradient algorithm on the solution of symmetric, positive definite systems. Now assume that in the system $\mathbf{Ax} = \mathbf{b}$, the $n \times n$ matrix $\mathbf{A}$ is symmetric positive semi-definite with a nullspace of dimension $p < n$. This problem illustrates that one can use a modified version of conjugate gradients to solve this system as well.

1. Prove that we can write $\mathbf{A}$ as
$$\mathbf{A} = \mathbf{M}\tilde{\mathbf{A}}\mathbf{M}^T$$
where $\mathbf{M}$ is an $n \times (n-p)$ matrix with orthonormal columns that form a basis for the column space of $\mathbf{A}$, while $\tilde{\mathbf{A}}$ is an $(n-p) \times (n-p)$ symmetric *positive* definite matrix (no nullspace) [Hint: Use the diagonal form of $\mathbf{A} = \mathbf{Q\Lambda Q}^T$]

2. Let the $n \times n$ matrix $\mathbf{P}$ be defined as $\mathbf{P} = \mathbf{MM}^T$. Explain (no formal proof required) why this is a projection matrix and onto what space it projects. How can we compute $\mathbf{P}$ without knowledge of the eigenvalues-eigenvectors of $\mathbf{A}$?

3. Show that, in order to have a solution to $\mathbf{Ax} = \mathbf{b}$, we must be able to write
$$\mathbf{b} = \mathbf{M}\tilde{\mathbf{b}}$$
for an appropriate vector $\tilde{\mathbf{b}} \in \mathbb{R}^{n-p}$

4. Let $\tilde{\mathbf{x}}$ be the solution to the system $\tilde{\mathbf{A}}\tilde{\mathbf{x}} = \tilde{\mathbf{b}}$ and explain why $\tilde{\mathbf{x}}$ is unique. Show that any solution to the original system $\mathbf{Ax} = \mathbf{b}$ can be written as $\mathbf{x} = \mathbf{M}\tilde{\mathbf{x}} + \mathbf{x}_0$ where $\mathbf{x}_0$ is in the nullspace of $\mathbf{A}$.

5. Consider the conjugate gradients algorithm for solving $\tilde{\mathbf{A}}\tilde{\mathbf{x}} = \tilde{\mathbf{b}}$

$$\begin{aligned}
&\tilde{\mathbf{x}}_0 = \text{initial guess}\\
&\tilde{\mathbf{s}}_0 = \tilde{\mathbf{r}}_0 = \tilde{\mathbf{b}} - \tilde{\mathbf{A}}\tilde{\mathbf{x}}_0\\
&\textbf{for } k = 0, 1, \ldots, 2\\
&\quad \tilde{\alpha}_k = \frac{\tilde{\mathbf{r}}_k^T \tilde{\mathbf{r}}_k}{\tilde{\mathbf{s}}_k^T \tilde{\mathbf{A}} \tilde{\mathbf{s}}_k}\\
&\quad \tilde{\mathbf{x}}_{k+1} = \tilde{\mathbf{x}}_k + \tilde{\alpha}_k \tilde{\mathbf{s}}_k\\
&\quad \tilde{\mathbf{r}}_{k+1} = \tilde{\mathbf{r}}_k - \tilde{\alpha}_k \tilde{\mathbf{A}} \tilde{\mathbf{s}}_k\\
&\quad \tilde{\mathbf{s}}_{k+1} = \tilde{\mathbf{r}}_{k+1} + \frac{\tilde{\mathbf{r}}_{k+1}^T \tilde{\mathbf{r}}_{k+1}}{\tilde{\mathbf{r}}_k^T \tilde{\mathbf{r}}_k} \tilde{\mathbf{s}}_k\\
&\textbf{end}
\end{aligned}$$

Show that we can compute a solution to the original system $\mathbf{Ax} = \mathbf{b}$ by using the following modification of the algorithm

$$
\begin{aligned}
&\mathbf{x}_0 = \text{initial guess} \\
&\mathbf{s}_0 = \mathbf{r}_0 = \mathbf{P}(\mathbf{b} - \mathbf{A}\mathbf{x}_0) \\
&\textbf{for } k = 0, 1, \ldots, 2 \\
&\quad \alpha_k = \frac{\mathbf{r}_k^T \mathbf{r}_k}{\mathbf{s}_k^T \mathbf{A} \mathbf{s}_k} \\
&\quad \mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{s}_k \\
&\quad \mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k \mathbf{PA}\mathbf{s}_k \\
&\quad \mathbf{s}_{k+1} = \mathbf{r}_{k+1} + \frac{\mathbf{r}_{k+1}^T \mathbf{r}_{k+1}}{\mathbf{r}_k^T \mathbf{r}_k} \mathbf{s}_k \\
&\textbf{end}
\end{aligned}
$$

[Hint: Show that $\mathbf{x}_k = \mathbf{M}\tilde{\mathbf{x}}_k, \mathbf{r}_k = \mathbf{M}\tilde{\mathbf{r}}_k, \mathbf{s}_k = \mathbf{M}\tilde{\mathbf{s}}_k, \tilde{\alpha}_k = \alpha_k$]

Solution

1. Since $\mathbf{A}$ is symmetric and positive definite it can be written as

$$
\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T = \begin{bmatrix} \mathbf{q}_1 & \mathbf{q}_2 & \cdots & \mathbf{q}_n \end{bmatrix} \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{bmatrix} \begin{bmatrix} \mathbf{q}_1^T \\ \mathbf{q}_2^T \\ \vdots \\ \mathbf{q}_n^T \end{bmatrix}
$$

Since $\mathbf{A}$ has a nullspace of dimension p, exactly $n-p$ of its eigenvalues, say $\lambda_1, \lambda_2, \ldots, \lambda_k$, are nonzero (and positive), while $\lambda_{k+1} = \lambda_{k+2} = \cdots = \lambda_n = 0$. Therefore

$$
\begin{aligned}
\mathbf{A} &= \begin{bmatrix} \mathbf{q}_1 & \cdots & \mathbf{q}_k & \mathbf{q}_{k+1} & \cdots & \mathbf{q}_n \end{bmatrix} \begin{bmatrix} \lambda_1 & & & & & \\ & \ddots & & & & \\ & & \lambda_k & & & \\ & & & 0 & & \\ & & & & \ddots & \\ & & & & & 0 \end{bmatrix} \begin{bmatrix} \mathbf{q}_1^T \\ \vdots \\ \mathbf{q}_k^T \\ \mathbf{q}_{k+1}^T \\ \vdots \\ \mathbf{q}_n^T \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{q}_1 & \mathbf{q}_2 & \cdots & \mathbf{q}_k \end{bmatrix} \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_k \end{bmatrix} \begin{bmatrix} \mathbf{q}_1^T \\ \mathbf{q}_2^T \\ \vdots \\ \mathbf{q}_k^T \end{bmatrix} = \mathbf{M}\tilde{\mathbf{A}}\mathbf{M^T}^T
\end{aligned}
$$

where the columns of the $n \times (n-p)$ matrix $\mathbf{M}$ form an orthonormal basis for the column space of $\mathbf{A}$ (see homework 3, problem 3.5) and $\tilde{\mathbf{A}}$ is symmetric and positive definite since it is diagonal and its diagonal contains only the positive eigenvalues of $\mathbf{A}$.

2. Since the columns of $\mathbf{M}$ form an orthonormal basis for the column space of $\mathbf{A}$, the matrix $\mathbf{P} = \mathbf{M}\mathbf{M}^T$ is the projection matrix onto the column space of $\mathbf{A}$. From homework 2, problem 2.1 (check the solutions) we know that if we have the $\mathbf{QR}$ decomposition of $\mathbf{A}$, we can get the projection matrix onto the column space of $\mathbf{A}$ as $\mathbf{P} = \mathbf{Q}\mathbf{Q}^T$. The $\mathbf{QR}$ decomposition can be computed using Gram-Schmidt, without any need to solve for the eigenvalues and eigenvectors of $\mathbf{A}$. Note that the columns of this $\mathbf{Q}$ *are not* the eigenvectors of $\mathbf{A}$, nevertheless the resulting projection matrix is exactly the same.

3. For any value of $\mathbf{x}$ the vector $\mathbf{Ax}$ lies in the column space of $\mathbf{A}$ (it's a linear combination of its columns with coefficients given by the individual elements of $\mathbf{b}$). Therefore, in order for $\mathbf{Ax} = \mathbf{b}$ to have a solution, $\mathbf{b}$ has to be in the column space of $\mathbf{A}$ as well. Another way to see this is

$$\mathbf{Ax} = \mathbf{b} \Rightarrow \mathbf{M}\tilde{\mathbf{A}}\mathbf{M}^T\mathbf{x} = \mathbf{b} \Rightarrow \mathbf{b} = \mathbf{M}(\tilde{\mathbf{A}}\mathbf{M}^T\mathbf{x}) = \mathbf{M}\tilde{\mathbf{b}}$$

4. The matrix $\tilde{\mathbf{A}}$ is positive definite and thus nonsingular, therefore the solution $\tilde{\mathbf{x}}$ to the system $\tilde{\mathbf{A}}\tilde{\mathbf{x}} = \tilde{\mathbf{b}}$ is unique. We know that any solution $\mathbf{x}$ to the original system $\mathbf{Ax} = \mathbf{b}$ can be written as $\mathbf{x} = \mathbf{x}_{CS} + \mathbf{x}_0$ where $\mathbf{x}_{CS}$ is in the column space of $\mathbf{A}$ and $\mathbf{x}_0$ is in the nullspace (see review session notes). We know that $\mathbf{x}_{CS}$ is unique and since it is in the column space it can be written as $\mathbf{x}_{CS} = \mathbf{M}\tilde{\mathbf{x}}$ where $\tilde{\mathbf{x}} \in \mathbb{R}^{n-p}$. Therefore we have

$$\mathbf{x} = \mathbf{M}\tilde{\mathbf{x}} + \mathbf{x}_0 \Rightarrow \mathbf{Ax} = \mathbf{AM}\tilde{\mathbf{x}} \Rightarrow \mathbf{b} = \mathbf{M}\tilde{\mathbf{A}}\mathbf{M}^T\mathbf{M}\tilde{\mathbf{x}} \Rightarrow \mathbf{M}\tilde{\mathbf{b}} = \mathbf{M}\tilde{\mathbf{A}}\mathbf{M}^T\mathbf{M}\tilde{\mathbf{x}} \Rightarrow$$
$$\Rightarrow \mathbf{M}^T\mathbf{M}\tilde{\mathbf{b}} = \mathbf{M}^T\mathbf{M}\tilde{\mathbf{A}}\mathbf{M}^T\mathbf{M}\tilde{\mathbf{x}} \Rightarrow \tilde{\mathbf{b}} = \tilde{\mathbf{A}}\tilde{\mathbf{x}}$$

5. We will show that each part of the proposed algorithm for solving $\tilde{\mathbf{A}}\tilde{\mathbf{x}} = \tilde{\mathbf{b}}$ translates to the corresponding part of the proposed modified algorithm

   - $\mathbf{x_0}$ =initial guess
     $\mathbf{s}_0 = \mathbf{r}_0 = \mathbf{P}(\mathbf{b} - \mathbf{A}\mathbf{x}_0)$

     Since in order to have a solution we must have $\mathbf{b} = \mathbf{M}\tilde{\mathbf{b}}$

     $\mathbf{s}_0 = \mathbf{r}_0 = \mathbf{P}(\mathbf{b} - \mathbf{A}\mathbf{x}_0) = \mathbf{M}\mathbf{M}^T(\mathbf{M}\tilde{\mathbf{b}} - \mathbf{M}\tilde{\mathbf{A}}\mathbf{M}^T\mathbf{x}_0) = \mathbf{M}(\tilde{\mathbf{b}} - \tilde{\mathbf{A}}\tilde{\mathbf{x}}_0) = \mathbf{M}\tilde{\mathbf{s}}_0 = \mathbf{M}\tilde{\mathbf{r}}_0$

     where $\tilde{\mathbf{x}}_0 = \mathbf{M}^T\mathbf{x}_0$ is the initial guess used in the conjugate gradient algorithm for $\tilde{\mathbf{A}}\tilde{\mathbf{x}} = \tilde{\mathbf{b}}$. We can also write the initial guess $\mathbf{x}_0 = \mathbf{M}\tilde{\mathbf{x}}_0 + \mathbf{x}_{NS}$ where $\mathbf{x}_{NS}$ is in the nullspace of $\mathbf{A}$.
     To continue with induction, assume that for $i = 0, 1, \ldots, k$ we have

$$\mathbf{x}_i = \mathbf{M}\tilde{\mathbf{x}}_i + \mathbf{x}_{NS}, \mathbf{r}_i = \mathbf{M}\tilde{\mathbf{r}}_i, \mathbf{s}_i = \mathbf{M}\tilde{\mathbf{s}}_i$$

   - For α_k we have

$$\alpha_k = \frac{\mathbf{r}_k^T\mathbf{r}_k}{\mathbf{s}_k^T\mathbf{A}\mathbf{s}_k} = \frac{\tilde{\mathbf{r}}_k^T\mathbf{M}^T\mathbf{M}\tilde{\mathbf{r}}_k}{\tilde{\mathbf{s}}_k^T\mathbf{M}^T\mathbf{A}\mathbf{M}\tilde{\mathbf{s}}_k} = \frac{\tilde{\mathbf{r}}_k^T\tilde{\mathbf{r}}_k}{\tilde{\mathbf{s}}_k^T\tilde{\mathbf{A}}\tilde{\mathbf{s}}_k} = \tilde{\alpha}_k$$

- For $\mathbf{x}_{k+1}$ we have

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{s}_k = \mathbf{M}\tilde{\mathbf{x}}_k + \mathbf{x}_{NS} + \tilde{\alpha}_k \mathbf{M}\tilde{\mathbf{s}}_k = \mathbf{M}(\tilde{\mathbf{x}}_k + \tilde{\alpha}_k \tilde{\mathbf{s}}_k) + \mathbf{x}_{NS} = \mathbf{M}\tilde{\mathbf{x}}_{k+1} + \mathbf{x}_{NS}$$

- For $\mathbf{r}_{k+1}$ we have

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k \mathbf{P}\mathbf{A}\mathbf{s}_k = \mathbf{M}\tilde{\mathbf{r}}_k - \alpha_k \mathbf{M}\mathbf{M}^T\mathbf{A}\mathbf{M}\tilde{\mathbf{s}}_k = \mathbf{M}\tilde{\mathbf{r}}_k - \tilde{\alpha}_k \mathbf{M}\tilde{\mathbf{A}}\tilde{\mathbf{s}}_k = \mathbf{M}\tilde{\mathbf{r}}_{k+1}$$

- For $\mathbf{s}_{k+1}$ we have

$$\mathbf{s}_{k+1} = \mathbf{r}_{k+1} + \frac{\mathbf{r}_{k+1}^T \mathbf{r}_{k+1}}{\mathbf{r}_k^T \mathbf{r}_k}\mathbf{s}_k = \mathbf{M}\tilde{\mathbf{r}}_{k+1} + \frac{\tilde{\mathbf{r}}_{k+1}^T \mathbf{M}^T \tilde{\mathbf{M}}\mathbf{r}_{k+1}}{\tilde{\mathbf{r}}_k^T \mathbf{M}^T \mathbf{M}\tilde{\mathbf{r}}_k}\mathbf{M}\tilde{\mathbf{s}}_k = \mathbf{M}\tilde{\mathbf{r}}_{k+1} + \frac{\tilde{\mathbf{r}}_{k+1}^T \tilde{\mathbf{r}}_{k+1}}{\tilde{\mathbf{r}}_k^T \tilde{\mathbf{r}}_k}\mathbf{M}\tilde{\mathbf{s}}_k = \mathbf{M}\tilde{\mathbf{s}}_{k+1}$$

Therefore our modified algorithm "translates" every step of conjugate gradients for $\tilde{\mathbf{A}}\tilde{\mathbf{x}} = \tilde{\mathbf{b}}$ into an equivalent step for the original system $\mathbf{A}\mathbf{x} = \mathbf{b}$

Problem 2

Consider a real function $f(x)$ that is differentiable on an interval $[a, b]$.

1. Find a quadratic polynomial $g(x)$ that approximates $f(x)$ on $[a, b]$ in that $f'(a) = g'(a)$, $f'(b) = g'(b)$ and $f\left(\frac{a+b}{2}\right) = g\left(\frac{a+b}{2}\right)$ [Hint: Consider expressing $g(x)$ as a quadratic polynomial of $(x - \frac{a+b}{2})$].

2. Define a numerical quadrature rule for $\int_a^b f(x)\,dx$ by integrating the interpolant $g(x)$ on $[a, b]$.

3. Prove that this integration scheme has degree of accuracy equal to 3.

4. Define the corresponding composite quadrature rule for $\int_a^b f(x)\,dx$ we obtain by sub-dividing $[a, b]$ into the n sub-intervals $\left[a + k\frac{b-a}{n}, a + (k+1)\frac{b-a}{n}\right]$

Solution

1. Let $g(x) = c_2\left(x - \frac{a+b}{2}\right)^2 + c_1\left(x - \frac{a+b}{2}\right) + c_0$. Using the given constraints we have

$$\left\{\begin{array}{c} g'(a) = f'(a) \\ g'(b) = f'(b) \\ g(\frac{a+b}{2}) = fa + b2 \end{array}\right\} \Rightarrow \left\{\begin{array}{c} c_2(a-b) + c_1 = f'(a) \\ c_2(b-a) + c_1 = f'(b) \\ c_0 = f(\frac{a+b}{2}) \end{array}\right\} \Rightarrow \left\{\begin{array}{c} c_2 = \frac{f'(b)-f'(a)}{2(b-a)} \\ c_1 = \frac{f'(a)+f'(b)}{2} \\ c_0 = f(\frac{a+b}{2}) \end{array}\right\}$$

Thus

$$g(x) = \frac{f'(b) - f'(a)}{2(b-a)}\left(x - \frac{a+b}{2}\right)^2 + \frac{f'(a) + f'(b)}{2}\left(x - \frac{a+b}{2}\right) + f\left(\frac{a+b}{2}\right)$$

2. We have

$$\begin{aligned}\int_a^b f(x)\,dx &\approx \int_a^b g(x)\,dx = \int_a^b \left[c_2\left(x-\frac{a+b}{2}\right)^2 + c_1\left(x-\frac{a+b}{2}\right)+c_0\right]dx \\ &= c_2\frac{(b-a)^3}{12}+c_0(b-a) \\ &\Rightarrow \int_a^b f(x)\,dx \approx (b-a)f\left(\frac{a+b}{2}\right)+\frac{(b-a)^2}{24}[f'(b)-f'(a)]\end{aligned}$$

3. The interpolant used approximates exactly polynomials of degree up to 2, thus the degree of accuracy is at least 2. We also have

$$\begin{aligned}\int_a^b x^3\,dx &= (b-a)\left(\frac{a+b}{2}\right)^3+\frac{(b-a)^2}{24}[3b^2-3a^2] = \frac{(b-a)(a+b)^3}{8}+\frac{(b-a)^3(a+b)}{8} \\ &= \frac{(b-a)(a+b)}{8}[(a+b)^2+(a-b)^2] = \frac{b^2-a^2}{4}(b^2+a^2) = \frac{b^4-a^4}{4}\end{aligned}$$

which is the exact result. To show that the degree of accuracy is exactly 3, we give the counterexample $f(x)=x^4$ on the interval $[-a,a]$

$$\int_{-a}^a x^4\,dx = 2a(0)^4+\frac{(2a)^2}{24}[4a^3+4a^3] = \frac{4}{3}a^5$$

which is not the exact result $2/5a^5$. Thus the method is third order accurate.

4. The compositie rule is

$$\int_a^b f(x)\,dx = \sum_{k=0} n-1\int_{a+k\frac{b-a}{n}}^{a+(k+1)\frac{b-a}{n}} f(x)\,dx$$

which is approximately

$$\sum_{k=0}^{n-1}\left\{\frac{b-a}{n}f\left(a+(2k+1)\frac{b-a}{2n}\right)+\frac{(b-a)^2}{24n^2}\left[f'\left(a+(k+1)\frac{b-a}{n}\right)-f'\left(a+k\frac{b-a}{n}\right)\right]\right\}$$

which is

$$\left\{\frac{b-a}{n}\sum_{k=0}^{n-1} f\left(a+(2k+1)\frac{b-a}{2n}\right)\right\}+\frac{(b-a)^2}{24n^2}[f'(b)-f'(a)]$$

5. If we know the *exact* value of $f'(a)$ and $f'(b)$ the rule we proved in 4 is third order accurate while only slightly more complex than the midpoint rule and should be prefered. Note that this wouldn't work if we tried to approximate $f'(a)$ and $f'(b)$ from nearby

values of f, since this approximation would have $O(h)$ error leading to an $O(h^3)$ error in the integration formula (same as the midpoint rule).

If we dont know $f'(a)$ and $f'(b)$ and third order accuracy is desired, Simpson's rule is the only option. Nevertheless, if first order accuracy is sufficient (for example if f is very smooth or if the discretization step h is already very small) the midpoint rule is simpler and requires much fewer floating point operations.

Problem 3

The first order divided difference is given by

$$f[x_0, x_1] = \frac{f(x_1) - f(x_0)}{x_1 - x_0}.$$

When x_0 is close to x_1 we have the approximation

$$f[x_0, x_1] \approx f'\left(\frac{x_0 + x_1}{2}\right)$$

Now let $z = (x_0 + x_1)/2$, $h = (x_1 - x_0)/2$ then the error is given as

$$E = f[x_0, x_1] - f'\left(\frac{x_0 + x_1}{2}\right) = \frac{f(z+h) - f(z-h)}{2h} - f'(z)$$

Prove that the error is

$$E = \frac{h^2}{6} f'''(z) + O(h^3)$$

Solution

Expanding $f(z-h)$ and $f(z+h)$ about z by using Taylor's theorem. The taylor expansion about z is

$$f(x) = f(z) + f'(z)(x-z) + \frac{1}{2} f''(z)(x-z)^2 + \frac{1}{6} f'''(z)(x-z)^3 + O((x-z)^4)$$

so we get

$$\begin{aligned} f(z+h) &= f(z) + hf'(z) + \frac{h^2}{2} f''(z) + \frac{h^3}{6} f'''(z) + O(h^4) \\ f(z-h) &= f(z) - hf'(z) + \frac{h^2}{2} f''(z) - \frac{h^3}{6} f'''(z) + O(h^4) \end{aligned}$$

Subtracting the second equation from the first gives

$$f(z+h) - f(z-h) = 2hf'(z) + \frac{1}{3} h^3 + O(h^4)$$

Dividing through by $2h$ and rearranging gives

$$\frac{f(z+h)-f(z-h)}{2h} - f'(z) = \frac{h^2}{6} f'''(z) + O(h^3)$$