May 2023 Ponder This Solution

By Daniel Copeland

Solution

$n = 2^{20}$: 52390.244

$n = 2^{30}$: 156032.269

Explanation

We factor the matrix A as $X^T X$, where $X : \mathbb{R}^n \to \mathbb{R}^5 \otimes \mathbb{R}^{32}$ is defined entrywise by

$$X^i_{j,k} = \frac{2^{j/2}}{\sqrt{31}} \delta_{k, f((i+1)(j+1))},$$

where $f : \mathbb{R} \to \{0, 1, \ldots, 31\}$ is the function $f(x) = \lfloor 32\big(\sin(x) - \lfloor \sin(x) \rfloor\big) \rfloor$.

Then the quantity we want to compute, $x^T A x$, can be written

$$x^T A x = \|Xx\|^2$$

Now Xx is a vector in $\mathbb{R}^5 \otimes \mathbb{R}^{32}$. We need to compute the sum of its components squared. For $0 \leq a \leq 4$ and $0 \leq l \leq 31$ we have

$$(Xx)_{a,l} = \frac{2^{a/2}}{\sqrt{31}} \sum_{i : f((i+1)(a+1)) = l} x_i.$$

Therefore

$$\|Xx\|^2 = \sum_{\substack{0 \leq a \leq 4 \\ 0 \leq l \leq 31}} \frac{2^a}{31} \left(\sum_{i : f((i+1)(a+1)) = l} x_i \right)^2.$$

This sum can be computed naively in time $O(n)$, which is what we did. Computing took ~27 seconds for $n = 2^{20}$ and ~8 hours for $n = 2^{30}$. We used Python and the NumPy package.

Acknowledgements

Thanks to Raphael Deem and Isaac Lawrence for fun conversations about this problem.

In [5]:
```
import numpy as np
n = 2**20
# n = 2**30

def f(x):
    return int(32*(np.sin(x) - np.floor(np.sin(x))))

def x(i):
    return -1 + 2 * i / (n-1)

def compute_inner_sums(n):
    inner_sums = [[0 for l in range(32)] for a in range(5)]
    for a in range(5):
        for i in range(n):
            inner_sums[a][f((i+1)*(a+1))] += x(i)
    return inner_sums

def compute_qform(n):
    tot = 0.0
    inner_sums = compute_inner_sums(n)
    for a in range(5):
        for l in range(32):
            tot += (inner_sums[a][l]**2)*2**a
    return tot/31

compute_qform(n)
```

Out[5]:
```
52390.27440217224
```

In [ ]:
```
```