CAAM 520 HW 1

Jacob Engel

January 29, 2019

1 Time for Large N Analysis

For a mesh size of 10, it takes the computer 0.003858 seconds. For a mesh size of 100, it takes the computer 25.662533 seconds. For a mesh size of 1000, it was hung. If it scaled the same it would take 2955 hours to run. It took the computer 6911 times longer to run for n = 100 than n = 10. When N = 1000, the largest array is the order of 10^6

The approximate storage in bytes is 3.2*10^7.

Not only would the code take thousands of hours to run, but it would also take up an unreasonable amount of memory.

2 Code Analysis

The function for total operartions is

$$\emptyset(N) = \ 35N^2 + 32N + 23$$

Sample code for the index-based Weighted Jacobian method [1]

```
for (int j = 1; j<N+1; j++){
  for (int i = 1; i<N+1; i++){
        int pos = i + j*(N+2);
        int lpos = i-1 + j*(N+2);
        int rpos = i+1 + j*(N+2);
        int bpos = i + (j-1)*(N+2);
        int upos = i + (j+1)*(N+2);
```

[1] https://en.wikipedia.org/wiki/Jacobi$_m$*ethod*

```
        double sum = 0;
        sum += (-1/pow(h, 2))*(u[lpos] + u[rpos] + u[bpos] + u[upos]);
        u1[pos] = (w*0.25*(pow(h,2)))*(b[pos] - sum) + (1-w)*u[pos];
  }
}
```

Finding the Error Vector

```
for (int j = 1; j<N+1; j++){
  for (int i = 1; i<N+1; i++){
        int pos = i + j*(N+2);
        int lpos = i-1 + j*(N+2);
        int rpos = i+1 + j*(N+2);
        int bpos = i + (j-1)*(N+2);
        int upos = i + (j+1)*(N+2);
        ue[pos] = fabs((1/(h*h))*(-1*(u[lpos]+u[rpos]+u[bpos]+u[upos]
) + 4*u[pos])-b[pos]);
  }
}
```

Nodes	Iterationss	Operations
10	776	$3.843*10^3$
20	2838	$1.4663*10^5$
30	6188	$3.2483*10^5$