Query about Solving Ax=b

Let’s start with smaller grid , taking n = 3, the grid looks like this

- is Outside, # is Boundary and . is Domain point

With this, we will fill the matrix A, with rows describing equation of Biharmonic Stencil at each Domain point.

```
-  -  -  -  -  -  -
-  #  #  #  #  #  -
-  #  .  .  .  #  -
-  #  .  .  .  #  -
-  #  .  .  .  #  -
-  #  #  #  #  #  -
-  -  -  -  -  -  -
```

```
 0  1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31 32 33 34
35 36 37 38 39 40 41
42 43 44 45 46 47 48
```

To do that we start by taking this matrix and get an indexing for x vector :

as i, j value if grid correspond to $i*(n+4)+j\ th$ value in x

```
i,j
2,2 = 0 0 1 0 0 0 0 0 2 -8 2 0 0 0 1 -8 20 -8 1 0 0 0 2 -8 2 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
2,3 = 0 0 0 1 0 0 0 0 0 2 -8 2 0 0 0 1 -8 20 -8 1 0 0 0 2 -8 2 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
2,4 = 0 0 0 0 1 0 0 0 0 0 2 -8 2 0 0 0 1 -8 20 -8 1 0 0 0 2 -8 2 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
```

defining matrix this way is easier as it doesn’t have to calculated for each point, and if we even encounter any point other than Domain, i.e. Initial points, Boundary points, Outside, their eqn. can simply be substituted.

the problem comes that if we create the x vector , we get it as follows :

```
- - - - - - - - # # # # # - - # . . . # - - # . . . # - - # . . . # - - # # # # # - - - - - - - -
```

And this layout will cause come problems while calculating because, we do not have any equation at the Outside points shown as - ,

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} \right), \quad i = 1, 2, \ldots, n.$$

the iterative approach requires us to have a value at a_{ii} should be non zero i.e. all entries along the main diagonal should be non zero,

my approach to get around this was to make $a_{0,i} = -1$ i.e. and check if first entry in row vector is -1 i would just skip that iteration,

I tried implementing this, but this did not work and the were a lot of randomness, still present

Another way to do it might be that we clamp all the points in groups as :

```
Domain             Boundary                        Outside
. . . . . . . . . # # # # # # # # # # # # # # # # - - - - - - - - - - - - - - - - - - - - - - - -
```

```
   .   #  -
.
#
```

and We produce the A matrix as follows , we only create A for only the (n+2)*(n+2) points in the domain and boundary, not on the outside points here we will not be needing to solve for the outside points but, they will always remain zero and not be updated are not solving them we will not be able to update.

```
0  0  0  0  0  0  0
0  0  0  0  0  0  0
0  0  .  .  .  0  0
0  0  .  .  .  0  0
0  0  .  .  .  0  0
0  0  0  0  0  0  0
0  0  0  0  0  0  0
```

If we do it this way and we go on to solve it with keeping the outside points as zero, then can’t we just approach this as a fixing them at zero and running the iteration from there.

like having a 2 layer thick boundary of 0’s zeroes