S1336 - Project 6

Erik Weilow

December 11, 2018

10.10 Numerical solution of the potential within a rectangular region

Subtask a)

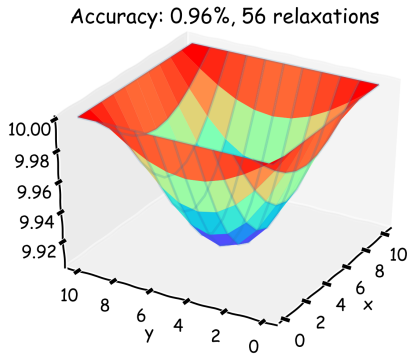
> Determine the potential $V(x, y)$ in a square region with linear dimension $L = 10$. The boundary of the square is at a potential $V = 10$. Choose the grid size $\Delta x = \Delta y = 1$

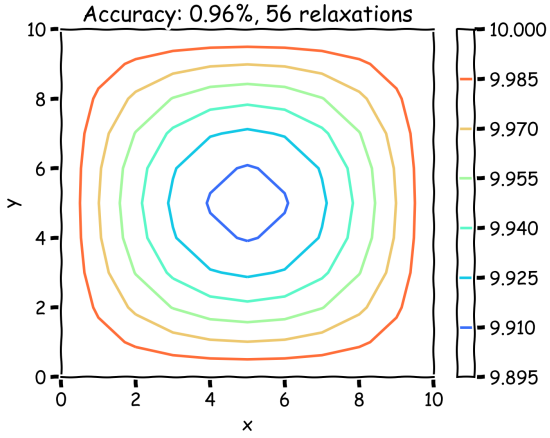
Guess the exact form of $V(x, y)$

A guess is $V(x, y) = 10$. This is reasonable as all the boundaries are $V = 10$ and the algorithm algorithm makes each point the average of the points around it. For a small grid ($\Delta x = \Delta y = 5$), the center point would be instantly equal to 10.

How many iterations are necessary to achieve 1% accuracy?

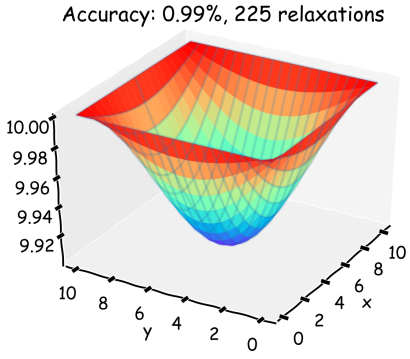
It takes about 56 relaxations to achieve 1% accuracy, producing the shape seen in the below plots.





Decrease the grid size by a factor of two, and determine the number of iterations that are now necessary to achieve 1% accuracy.

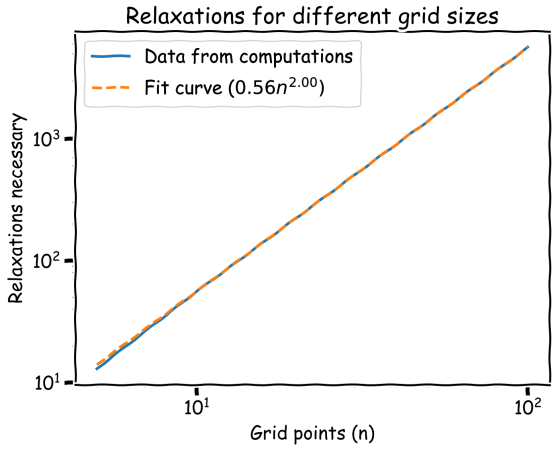
With a grid size $\Delta x = \Delta y = 0.5$, it takes instead closer to 225 relaxations to achieve the same accuracy.



For these two simple cases, we can see that the amount of relaxations could potentially follow

$$n_{relaxations} \propto \frac{1}{\Delta x \Delta y} \propto (n_{gridpoints})^2$$

This can be shown via the following loglog diagram showing the necessary relaxations for different grid sizes, where the curve $0.56n^2$ can be fit to the computed data.

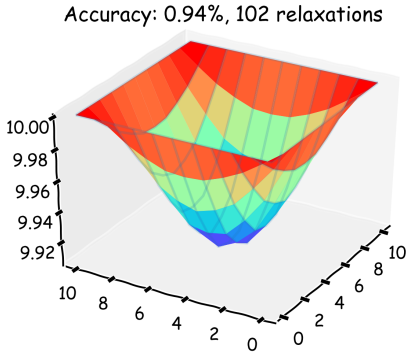


Subtask b)

> Consider the same geometry as in part (a), but set the initial potential at the interior sites equal to zero except for the center site whose potential is set equal to four.

Does the potential distribution evolve to the same values as in part (a)?

It does evolve to the same values as in part (a), but does so much slower with around 102 relaxations necessary.



What is the effect of a poor initial guess?

The result is really only that the convergence is slower.

Are the final results independent of your initial guess?

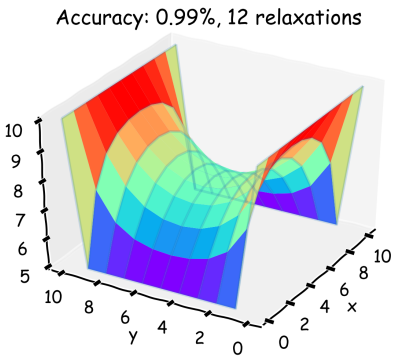
The final results should be independent of the guess. Since each point in every relaxation step is set to the average of the 4 surrounding points, there will only be no change if all points in the grid fulfill $V(x, y) = 10$.

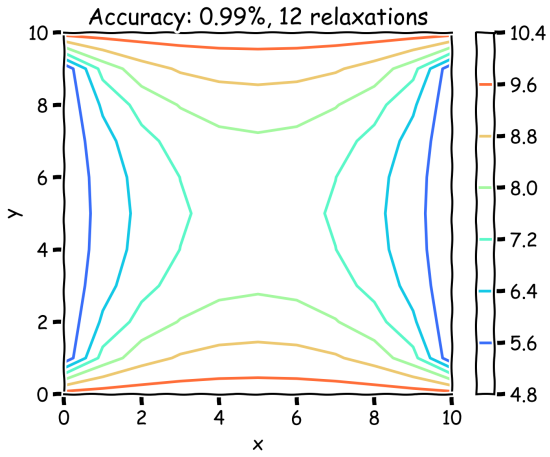
Subtask c)

> Now modify the boundary conditions so that the value of the potential at the four sides is 5, 10, 5, and 10, respectively. (Start with a reasonable guess for the initial values of the potential at the interior sites and iterate until 1% accuracy is obtained.)

Sketch the equipotential surfaces.

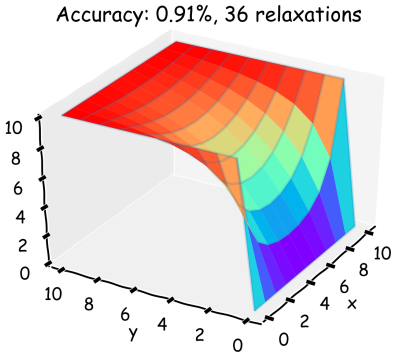
By these boundary conditions, we get a saddle as seen in the below figure (and equipotential surfaces plot).

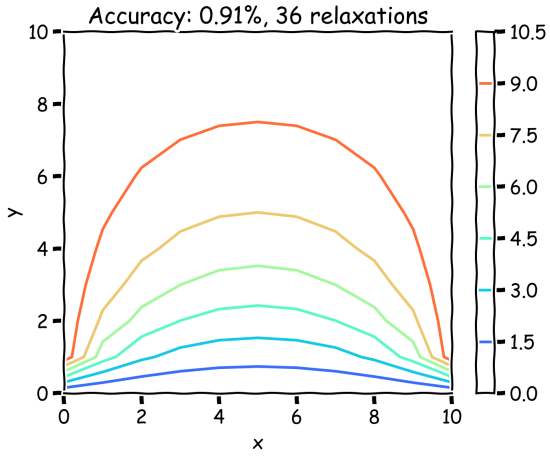




What happens if the potential is 10 on three sides and 0 on the fourth?

Now we get a sort of rounded slope, that tends toward $V = 0$ for $y = 0$.





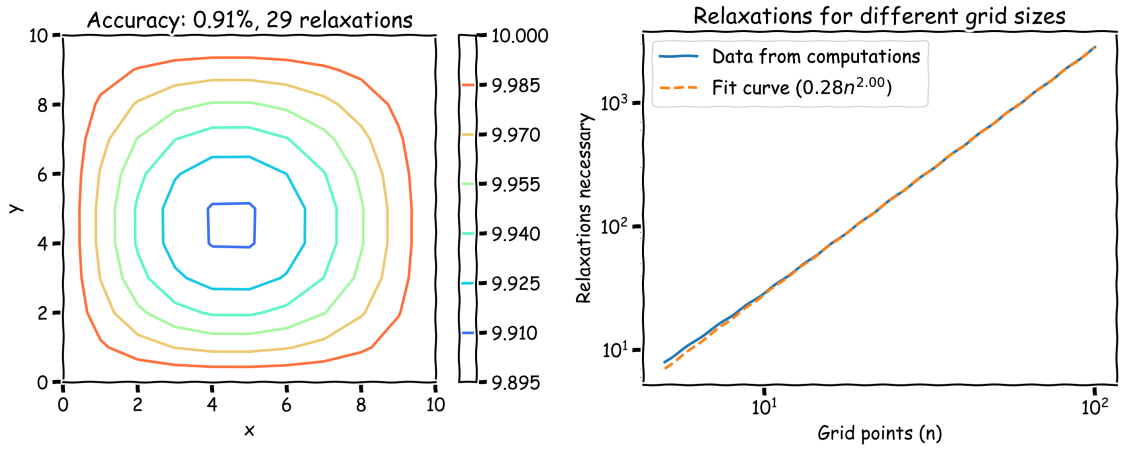
10.11 Gauss-Seidel relaxation

Subtask a)

> Use a modification of the program used in 10.10 so that the potential at each site is updated sequentially. In this way the new potential of the next site is computed using the most recently computed values of its nearest neighbor potentials.

Are your results better, worse, or about the same as for the simple relaxation method?

The convergence is about twice as fast, but produces asymmetrical equipotential surfaces as seen in the left figure. For the relaxations needed for different grid point counts, the constant is about half however.



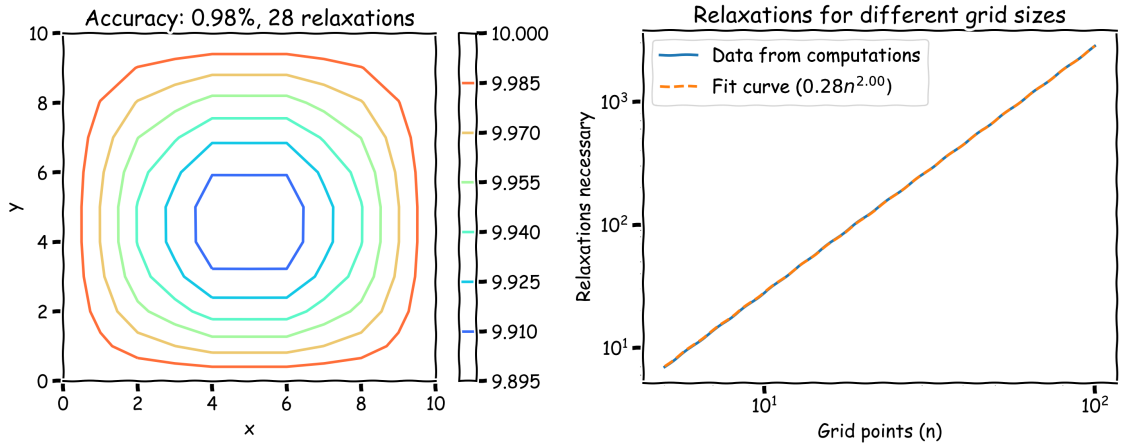
The tradeoff is that the results might be slightly less accurate, with the bonus being increased speed. For larger grids, this method also only have to store a single grid - a plus for the memory usage.

Subtask b)

> Imagine coloring the alternate sites of a grid red and black, so that the grid resembles a checkerboard. Modify the program so that all the red sites are updated first, and then all the black sites are updated.

Do your results converge any more quickly than in part (a)?

For smaller grids, this method requires perhaps a single iteration less. For larger grids, the constant is still more or less the same.



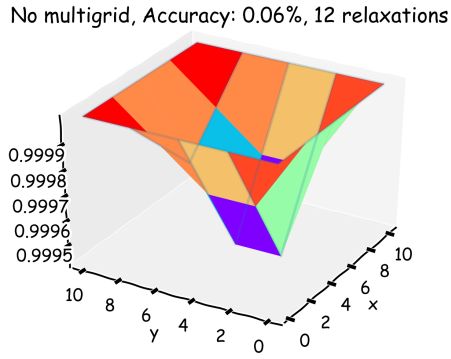
It's worth noting that the loglog plot for the checkered method has data from computations matching the fit curve exactly, while for the method in (a) the computed data trends above for small grids while still asymptotically matching the curve for larger grids.

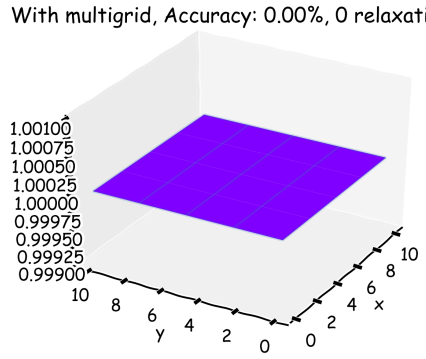
10.26 The multigrid method

Subtask a)

> Write a program that implements the multigrid method using Gauss-Seidel relaxation on a checkerboard lattice. Test your program on a 4x4 grid whose boundary sites are all equal to unity, and whose initial internal sites are set to zero.

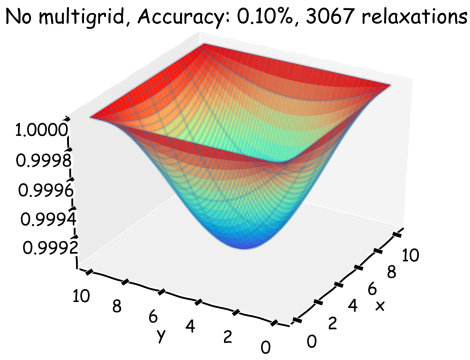
While the results are fairly boring, here are some illustrative plots:

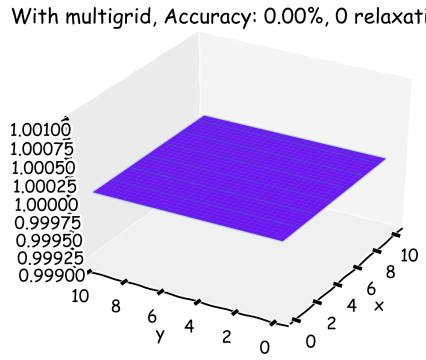




The result is that the rightmost plot, set up using the multigrid method, requires no further relaxation. It was run with a single relaxation step on each progressive prolongation up to a grid size of 2^4, with no relaxation on the restrictions down to a grid size of 2^2. In effect, we only need 4 relaxation steps to produce the rightmost grid, which really isn't comparable to the leftmost plot that requires 12 relaxations and still isn't close to the exact result.

This effect is further amplified if we look at a larger grid (2^6):





In this case, the multigrid method was set up to run a single relaxation on each prolongation up to a grid size of 2^8, with no relaxation on the restrictions down to a grid size of 2^6. The result is 8 relaxations necessary to produce an *exact* result, rather than 3067 relaxations to produce an approximate result.

Subtask b)

> The exact solution for part (a) gives a potential of unity at each point.

How many relaxation steps does it take to reach unity within 0.1% at every site by simply using the 4x4 grid?

As answered in (a), it takes about 12 relaxation steps.

How many steps does it take if you use one coarse grid and continue until the coarse grid values are within 0.1% of unity?

This one is interesting, because the course grid is a single point surrounded by our boundary conditions. On the first relaxation, it will be set to the average of the surrounding points ($V = 1$), which results in the entire grid taking on unity.

Is it necessary to carry out any fine grid relaxation steps to reach the desired accuracy on the fine grid?

If we start with a grid size of 2x2, with a single non-boundary point in the middle, during all prolongations all the new points will take on the average of the neighbours (from the previous, smaller size). This means that only the first course grid needs a relaxation step, as subsequent finer grids will become unity[1].

Next start with the coarsest scale, which is just one site. How many relaxation steps does it take now?

In this case, no relaxation steps should be necessary, according to the same reasoning as above.

Subtask c)

> Repeat part (b), but change the boundary so that one side of the boundary is held at a potential of 0.5.

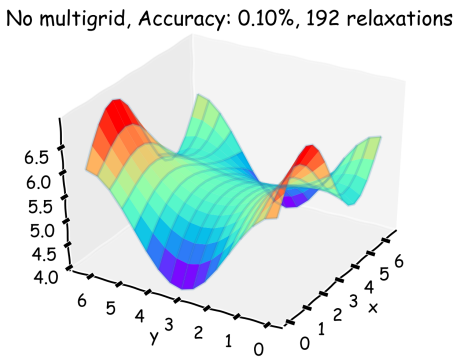
Experiment with different sequences of prolongation, restriction, and relaxation.

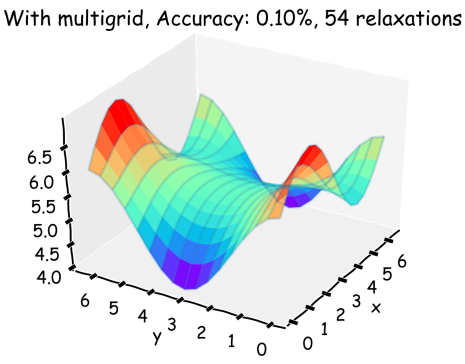
I decided that the boundary from this task is boring and defined my own boundary

$$V_{boundary}(x, y) = 5 + cos(x)sin(y)$$

with a grid spanning $x \in [0, 2\pi], y \in [0, 2\pi]$.

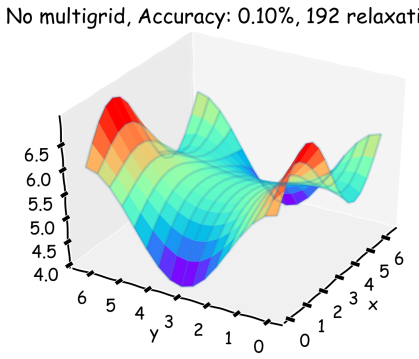
The first sequence we can look at is one that does 5 relaxations on every prolongation, up to a grid size of 2^4:

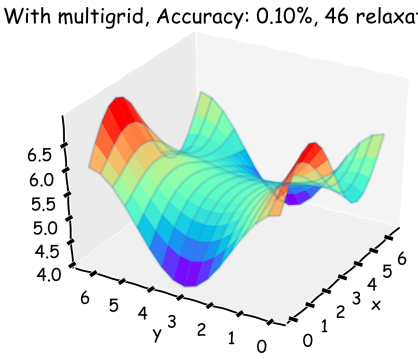




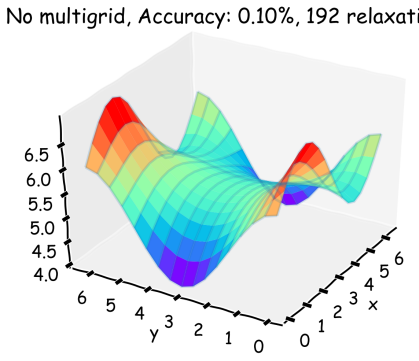
[1] resistance is futile

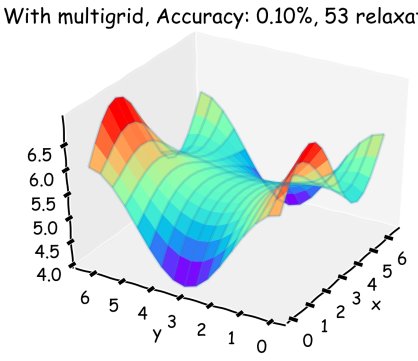
Here the multigrid method does a total of $54+5\cdot 4 = 74$ relaxations and the simple iterative method does 192 relaxations, to produce the same accuracy.
We can then look at a sequence that applies 5 relaxations on every prolongation, up to a grid size of 2^8, and then applies 5 relaxations on every restriction down to a grid size of 2^4:





Here the multigrid method does a total of $46 + 8 \cdot 4 + 4 \cdot 4 = 94$ relaxations, still less than 192. The final sequence is similar to the previous one, but one that doesn't relax the grid on restrictions:





Here the multigrid method does a total of $53 + 8 \cdot 4 = 85$ relaxations, still less than 192. It seems to be better not to apply relaxation when restricting the grid size, as the effect of that is to smooth the data rather than to just supersample it.

All in all, the multigrid method seems to be much more effective than just the standard Gauss-Seidel relaxation.