

CO25: Laplace's Equation

T.P. Galligan
Brasenose College
Trinity Term 2017

1 Abstract

We use the method of successive over-relaxation to approximate a solution to a two-dimensional Laplace's equation with Dirichlet boundary conditions. The solution is compared to the analytical result, and the convergence process is examined for different regimes of the over-relaxation parameter α .

2 Introduction

Laplace's equation is a linear second order partial differential equation. It takes the form

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = 0 \quad (1)$$

Where $\phi = \phi(x, y) : \mathbb{R}^2 \rightarrow \mathbb{R}$.

Laplace's equation occurs frequently in mathematics and physics, for example in complex analysis [1], two-dimensional fluid mechanics [2], and electrostatics problems [?]. We use a method of over-relaxation to approximate the solution to eq. (1) with the (Dirichlet) boundary conditions.

$$\phi(0, y) = 0 \quad \forall y \in [0, 1] \quad (2)$$

$$\phi(x, 0) = 1 \quad \forall x \in [0, 1] \quad (3)$$

$$\phi(1, y) = \sin(1) \sinh(y) \quad \forall y \in [0, 1] \quad (4)$$

$$\phi(x, 1) = \sin(x) \sinh(1) \quad \forall x \in [0, 1] \quad (5)$$

3 Theory

Let the domain be rectangular and described by $x \in [x_0, x_1]$ and $y \in [y_0, y_1]$. We discretise the solution into a $p+1$ by $q+1$ rectangular grid given by $x_i = x_0 + i\delta x$, $y_j = y_0 + j\delta y$ where $i \in [0, p] \subset \mathbb{Z}$ and similarly $j \in [0, q] \subset \mathbb{Z}$. The mesh spacing is then $\delta x = (x_1 - x_0)/p$, $\delta y = (y_1 - y_0)/q$. We let $\phi_{i,j} \equiv \phi(x_i, y_j)$ be the exact solution at mesh point (i, j) , and $\psi_{i,j} \approx \phi_{i,j}$ be the numerical solution at that point.

We Taylor Expand ψ about some point (i, j) :

$$\psi_{i+1,j} = \psi_{i,j} + \delta x \left. \frac{\partial \psi}{\partial x} \right|_{i,j} + \frac{\delta x^2}{2} \left. \frac{\partial^2 \psi}{\partial x^2} \right|_{i,j} + \mathcal{O}(\delta x^3) \quad (6)$$

$$\psi_{i-1,j} = \psi_{i,j} - \delta x \left. \frac{\partial \psi}{\partial x} \right|_{i,j} + \frac{\delta x^2}{2} \left. \frac{\partial^2 \psi}{\partial x^2} \right|_{i,j} + \mathcal{O}(\delta x^3) \quad (7)$$

$$\psi_{i,j+1} = \psi_{i,j} + \delta y \left. \frac{\partial \psi}{\partial y} \right|_{i,j} + \frac{\delta y^2}{2} \left. \frac{\partial^2 \psi}{\partial y^2} \right|_{i,j} + \mathcal{O}(\delta y^3) \quad (8)$$

$$\psi_{i,j-1} = \psi_{i,j} - \delta y \left. \frac{\partial \psi}{\partial y} \right|_{i,j} + \frac{\delta y^2}{2} \left. \frac{\partial^2 \psi}{\partial y^2} \right|_{i,j} + \mathcal{O}(\delta y^3) \quad (9)$$

Then we can approximate eq. (1) to first order using the four adjacent mesh points. This gives us the approximation

$$\frac{\psi_{i+1,j} - 2\psi_{i,j} + \psi_{i-1,j}}{\delta x^2} + \frac{\psi_{i,j+1} - 2\psi_{i,j} + \psi_{i,j-1}}{\delta y^2} = 0 \quad (10)$$

We can then solve this equation using a variety of numerical approaches.

4 Methods

We use an iterative method described in [4]. We wish to solve (1) for the boundary conditions eqs. (2) - (5) for a uniform 7×7 grid. In this case, $p = q = 7$ and we have the iteration procedure

$$\psi_{i,j}^{m+1} = \psi_{i,j}^m + \frac{\alpha R_{i,j}^m}{4} \quad (11)$$

Where we define the residual

$$R_{i,j}^m := \psi_{i,j+1}^m + \psi_{i-1,j}^m + \psi_{i-1,j}^m + \psi_{i+1,j}^m - 4\psi_{i,j}^m. \quad (12)$$

and the over-relaxation parameter

$$\alpha := \frac{2}{1 + \sin \frac{\pi}{p}} \quad (13)$$

We set a maximum iteration number to stop the process in the case of divergent iterations, and we seek to minimise the residual matrix $R_{i,j}$. We break the iterations when the residual is less than 0.001. Given the function is no more than on the order of 1 in the domain, this is a reasonable condition (as will be shown upon comparison with the analytic solution). Full details of the method are given in the appendix.

5 Results

We examined the numerical solution and the convergence process for a variety of regimes of α .

5.1 $\alpha = 1.1$

Theory (see, for example, [3]) suggests that we should see convergent behaviour. This is indeed the case, and the numerical and analytical solutions are virtually indistinguishable. By analysing the sample values, we see a monotonic convergence towards the analytical solution (see Fig.2).

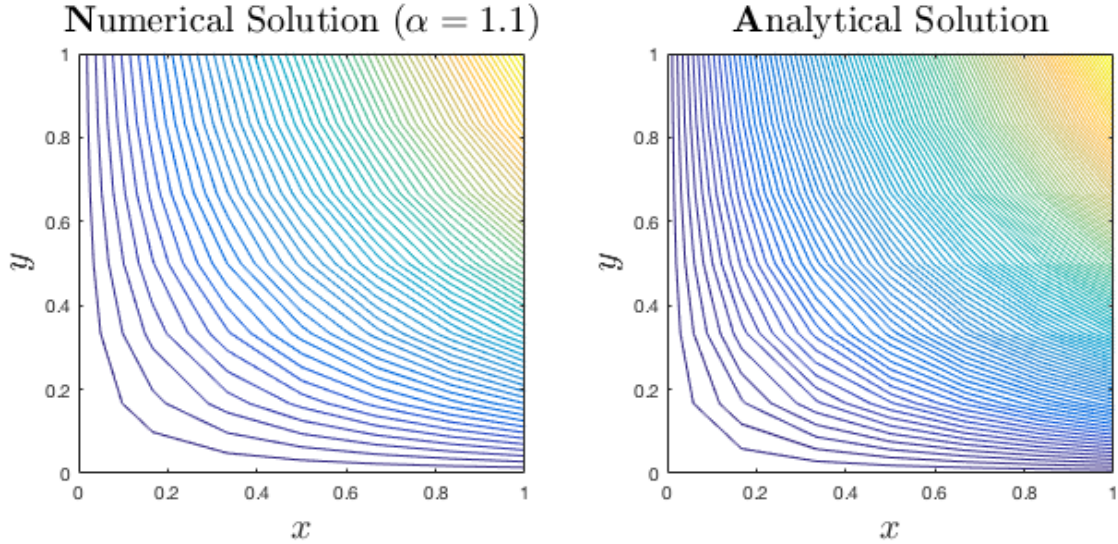


Figure 1: Comparison of numerical and analytical solutions for $\alpha = 1.1$.

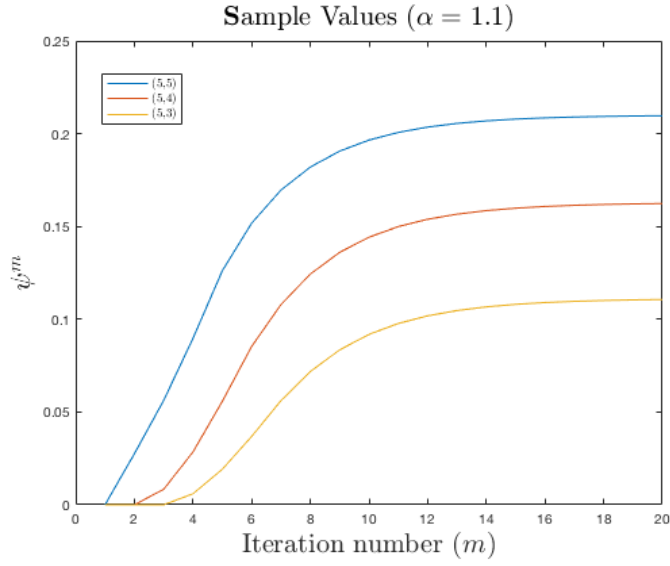


Figure 2: Convergence to the analytical solution for $\alpha = 1.1$

5.2 $\alpha = 1.25$

The analytical and numerical solutions were again very similar, as seen by the convergence of the sample points:

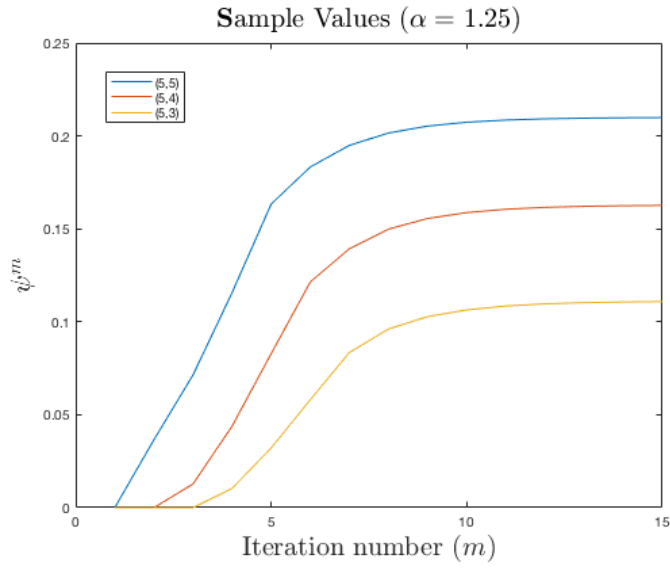


Figure 3: Convergence of sample values for $\alpha = 1.25$

5.3 $\alpha = 1.45$

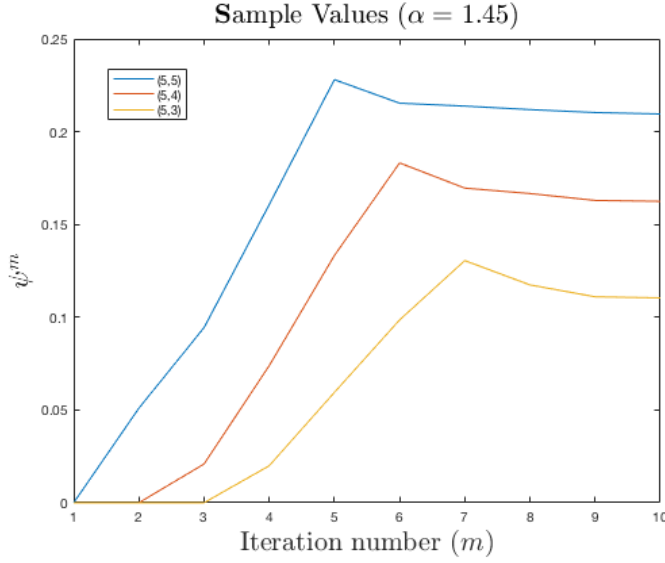


Figure 4: Results from a slit of width 10 at the centre

At $\alpha = 1.45$, the system still converges on the analytical solution, but does so in an oscillatory manner (seen by the peak at $m \approx 5$). We note that this is a much faster convergence than the previous values of α .

5.4 $\alpha = 2.1$

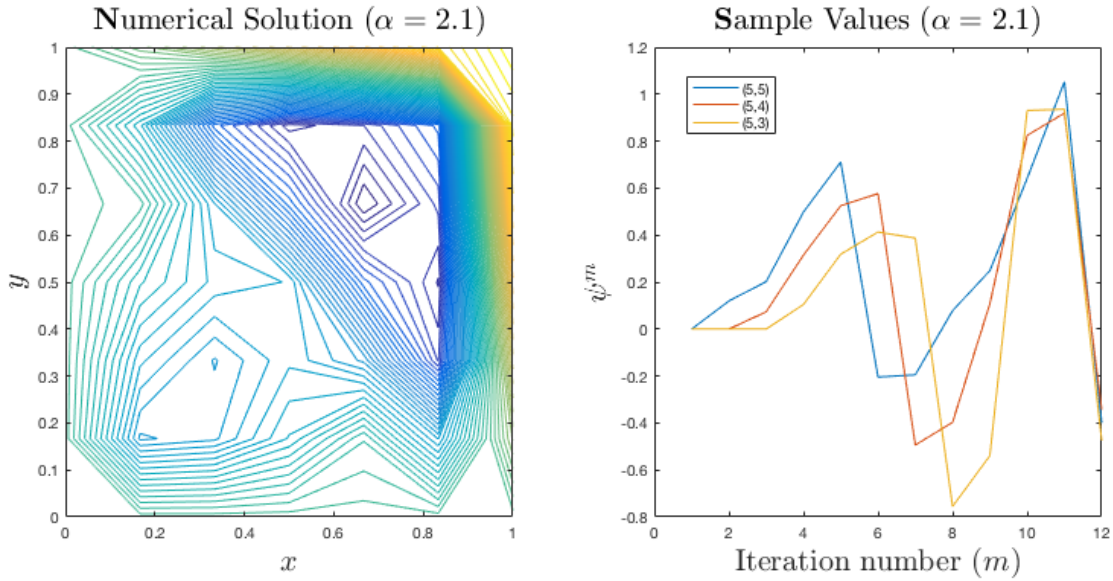


Figure 5: Divergence of the system for $\alpha = 2.1$

When the over-relaxation parameter is too large, [3] implies that the system will not reach the analytical solution, but instead will diverge with each iteration. As is clear from the second graph in Fig. 5, the behaviour of the system's divergence is oscillatory.

6 Conclusions

The *Successive Over-Relaxation* (SOR) technique has been shown to be effective in solving Laplace's equation in two dimensions. Results suggest any choice of $\alpha \in [1, 2)$ will give a solution, though the rate of convergence to that solution varies with α . Further research could investigate rates of convergence

against α , and this could be compared with theoretical predictions. We see that, as predicted by theory, the iteration process produces an oscillatory divergent solution for $\alpha > 2$. Indeed this is always the case when using SOR, and convergence is only reached when $\alpha \in (0, 2)$.

References

- [1] Graduate Course in Analytical Methods at UCL, Dr H. J. Winson.
<http://www.ucl.ac.uk/~ucahwhi/LTCC/sectionF-complex.pdf>
- [2] Notes from a course in Calculus from Queen Mary, University of London.
http://qplus.qmul.ac.uk/pluginfile.php/283383/mod_resource/content/1/Chapter%207%20-%20Laplace%2C%202010.pdf
- [3] Course on Numerical Methods, R.L. Burder & J.D. Faires, Dublin City University.
https://www.math.ust.hk/~mamu/courses/231/Slides/CH07_4A.pdf
- [4] CO25 Lab Script, Physics Practical Course, University of Oxford.
<http://www-teaching.physics.ox.ac.uk/>

7 Appendix

7.1 Code (MATLAB)

```
n = 7; %number of grid spacings
a = 2.1; %over-relaxation parameter (alpha)
psi=zeros(n,n); %initial 7x7 matrix of zeros, includes boundary
    conditions at x,y=0
x= [0:1/(n-1):1];
y = [0:1/(n-1):1];

for i=2:n %boundary conditions along y=1 and x=1
    psi(1,i) = sin(x(i))*sinh(1);
    psi(2:(n-1),n) = sin(1)*sinh(y((n-1):-1:2));
end

R=zeros(n); %initialize the value of the residual
iter=1; %initial iteration number
iterMAX=30; %max iterations
errorMAX=0.001; %condition for convergence
sample1 = zeros(1,iterMAX); %initialise vectors for sample convergence
sample2 = zeros(1,iterMAX);
sample3 = zeros(1,iterMAX);
%initialise vectors for sample convergence
while iter<iterMAX %main calculation with strict condition for stopping
    (max no. of iterations)
    for i=(n-1):-1:2 %sweeping right to left
        for j=2:(n-1) %sweeping up
            R(i,j) = psi(i,j+1) +psi(i,j-1) + psi(i+1,j) +psi(i-1,j)
                - 4*psi(i,j); %calculate residual
            psi(i,j) = psi(i,j) +a*R(i,j)/4; %perform iteration
        end
        sample1(iter)=psi(5,5); %allocate sample values to see the
            convergence
        sample2(iter)=psi(5,4);
        sample3(iter)=psi(5,3);
    end
end
```

```

end
iter = iter+1; %update iteration number
if abs(max(max(R))) < errorMAX %test for convergence condition. If
    satisfied then proceed to plotting.
    for i = 1:7 %aligns the matrix entries with their correct
        positions for the contour function (line 41)
        A(i,:) = psi(8-i,:);
    end
    for i=1:7 %produce analytic result
        for j=1:7
            Z(i,j)=sin(x(i))*sinh(y(j));
        end
    end
    end
A=transpose(A); %make sure plots are in same orientation.
sample1 = sample1(1:iter-1) %crop sample vectors
sample2 = sample2(1:iter-1)
sample3 = sample3(1:iter-1)

[X,Y] = meshgrid(x,y); %plot both the computed and the analytic
    solutions if convergence occurs.
%This happens if convergence condition satisfied, or max no. of
    steps reached
figure
subplot(1,2,1); contour(X,Y,A,60); xlabel('$x$', 'Interpreter', '
    latex', 'fontsize', 20); ylabel('$y$', 'Interpreter', 'latex', '
    fontsize', 20); title('\textbf Numerical Solution ($\alpha =
    2.1$)', 'Interpreter', 'latex', 'fontsize', 20);
subplot(1,2,2); plot(sample1) %plot samples to see convergence
hold on
plot(sample2)
hold on
plot(sample3)
title('\textbf Sample Values ($\alpha = 2.1$)', 'Interpreter', '
    latex', 'fontsize', 20)
xlabel('Iteration number ($m$)', 'Interpreter', 'latex', 'fontsize'
    ,20)
ylabel('$\psi^m$', 'Interpreter', 'latex', 'fontsize', 20)
hold off
solution=A %display the calculated solution
analytic_solution=Z %display the analytic solution
sprintf('Total number of iterations is %d', iter-1)%display the
    convergence iteration

break %end iteration loop

end

end

```