Numerical PDE II HW 1 : Elliptic Equations on the Square

Dhruv Balwada

February 6, 2015

1 Statement of Problem

In this problem we will explore the Gauss-Seidel, SOR and Conjugate Gradient methods for the solution of the Dirichlet problem for the PDE

$$u_{xx} + u_{yy} = f; x \in \omega \tag{1}$$

$$u(x) = g(x); x \in \partial\omega \tag{2}$$

for $\omega = [0, 1] \times [0, 1]$. The laplacian is approximated by the standard second order five point stencil approximation.

2 Description of the Mathematics

We are required to solve an elliptic equation on the square. Here we will approximate the laplacian operator using a second order difference approximate

$$\nabla_h U = (\delta_x^+ \delta_x^- + \delta_y^+ \delta_y^-)U = F_{ij} \tag{3}$$

where the δ operators are the standard notation. This is a 5 point stencil. We show that this approximation is second order on attached hand written notes.

To solve this system on a square grid we need to use an iterative solver. In this exercise we will use the SOR, Gauss Seidel(special case of SOR) and conjugate gradient methods.

The SOR method involves solving the following

$$U_{ij}^{k+1} = U_{ij}^k - \omega h^2/4 R_{ij}^k \tag{4}$$

where the residual is given by the following.

$$R_{ij}^k = F_{ij} - \frac{U_{i+1,j}^k + U_{i-1,j}^{k+1} + U_{i,j+1}^k + U_{i,j-1}^{k+1} - 4U_{i,j}^k}{h^2} \tag{5}$$

ω is a parameter that can be tuned for faster convergence.

(a)

Figure 1: Growth factor as a function of θ for Jacobi(brown) and Gauss-Seidel(blue)

The Gauss Seidel like the Jacobi method acts as a smoother(figure 1). This has been shown analytically in the attached hand written notes. The plots clearly show that the Gauss Seidel damps away higher frequencies much faster than the Jacobi. Also the order of convergence for the Gauss Seidel is 1 vs Jacobi having order 2. The higher the order the slower the scheme.

We don't analyze the behavior of the Conjugate gradient method but we learn from observing the solutions below that this method is the fastest, with the preconditioned conjugate gradient being even faster.

3 Description of the Algorithm

3.1 Residual for SOR

Input forcing and indices at which to evaluate
Evaluate and return $R_{ij} = F_{ij} - 1/h^2(U^k_{i+1,j} + U^{k+1}_{i-1,j} + U^k_{i,j+1}U^{k+1}_{i,j-1} - 4U^k_{i,j}$

3.2 SOR

Input $F^{N-1}_{ij=1}$, $U^{N-1}_{ij=1} = 0$
Assign boundary conditions
for k=1 to kmax **do**
  $R_{max} = 0$
  for j =1 to N-1 **do**
    for i =1 to N-1 **do**
      R = RESIDUAL(i,j, F_{ij}, U_{ij})
      $U_{ij} = U_{ij} + 0.25 * h^2 * \omega * R$
      $R_{max} = MAX(R_{max}, R)$
    end for
  end for
  IF $R_{max} < Tol$ EXIT

end for

3.3 Conjugate Gradient

Fix the forcing, boundary conditions and starting interior values(just set to zero here).
Compute R = Y -AX(where Y is the forcing and A is just the approximation to the laplacian and X is the starting condition)
Solve HZ = R for Z (H = 1 for no preconditioning case)
$V < -Z$
c =(R,Z) (** Dot product**)
for k=1 to kmax **do**
  $Z < -AV$
  $\omega < -C/(V, Z)$
  $X < -X + \omega V$
  $R < -R + \omega Z$
  IF $||R||_2 < Tol$ EXIT
  Solve HZ = R for Z
  $d < -(R, Z)$
  $V < -Z + d/cV$
  $c < -d$
end for

3.4 Preconditioning using SSOR for Conjugate Gradient

Z = R
for j = 1 to N-1 **do**
  for i = 1 to N-1 **do**
    $Z_{ij} = -\omega/4(2 - \omega)R_{ij} + \omega/4(Z_{i-1,j} + Z_{i,j-1})$
  end for
end for
for j = 1 to N-1 **do**
  for i = 1 to N-1 **do**
    $Z_{ij} = -4Z_{ij}$
  end for
end for
for j = N-1 to 1 **do**
  for i = N-1 to 1 **do**
    $Z_{ij} = \omega/4(Z_{i+1,j} + Z_{i,j+1}) - 0.25 * Z_{i,j}$
  end for
end for

4 Results

For this assignment I will be solving the following Poisson equation as a test case

$$\nabla u = -2cosxsiny \tag{6}$$

The boundary conditions on the square are $cosxsiny$.

4.1 SOR and Gauss Seidel

The SOR method has been tested for different values of ω including 1(Gauss Seidel). The behavior of the scheme has been presented in Table 1. and 2. and Figure 2. and 3. The residue decreases extremely rapidly at first and then the rate of decrease slows down till approximately the order of the scheme and then the residue decreases rapidly again. The quick initial decay is related to the rapid decay of the high frequencies and then it takes longer for the low frequency errors to decay. Another curious feature that is noticed is that once the residue is close to the error that is expected based on the order of the scheme the decay becomes faster and looks linear on the semilog plots. Generally the convergence seems to be faster for higher ω. For every doubling of number of grid points the number of iterations to convergence quadrupuled. The SOR method did not converge to a tolerance of 10^{-10} for $\omega = 1.9$ and a grid size of 256. Instead it was oscillating around 10^{-9}. Also for the same grid but $\omega = 0.1$ the convergence time was extremely long and I quit the process before convergence.

	256	128	64	32
0.1	-	826661	202487	50577
0.5	523307	129099	32220	8058
1.0	178247	43693	10911	2726
1.5	62352	14960	3723	919
1.9		2405	460	308

Table 1: Number of iterations to 10^{-10} convergence for SOR with different ω

	256	128	64	32
0.1	-	821.1456	53.579	3.983
0.5	1471.9	94.769	7.027	0.529
1.0	569.795	32.24	2.681	0.178
1.5	202.639	12.399	1.101	7.23E-2
1.9		2.502	0.113	2.417E-2

Table 2: Time of convergence to 10^{-10} for SOR with different ω

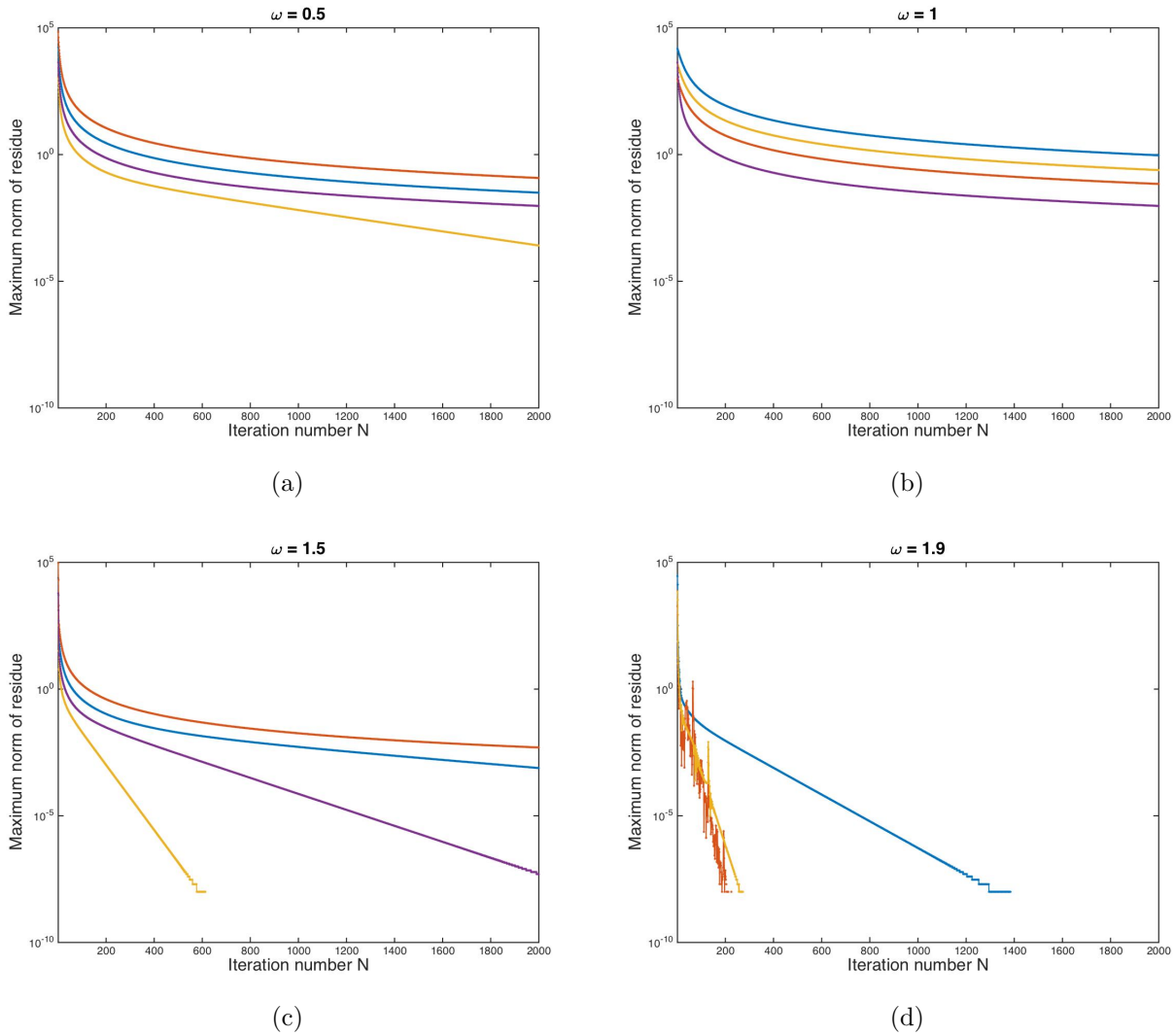


Figure 2: Behaviour of residue versus iteration for different values of ω.

4.2 Conjugate gradient(no preconditioning/preconditioned)

The conjugate gradient method was implemented and it works. The results are shown in Table 3. and 4. and figure 4. The residual decreases with iterations. The decay is extremely rapid at first followed by a slower decay phase till it has converged to approximately the order the scheme after which another rapid decay phase starts. The norm reduces in a similar way to SOR but is much faster. The error doesn't decrease monotonically but has small oscillations. The preconditioned conjugate gradient is about twice as fast as the non preconditioned. For every doubling of number of grid points the number of iterations to convergence doubled. Also conjugate gradients is a few hundred times faster than SOR methods.

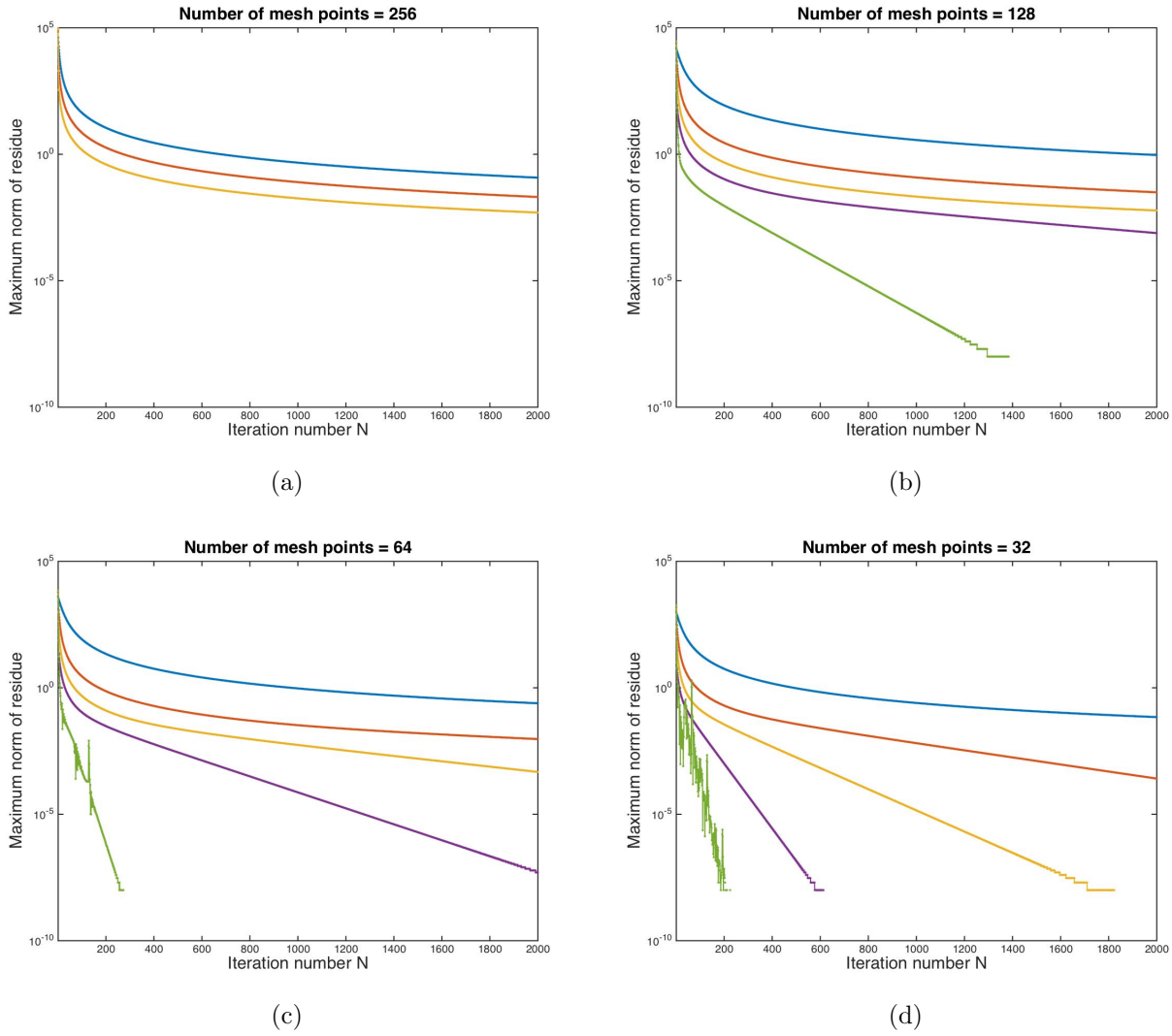


Figure 3: Behaviour of residue versus iteration for different mesh sizes

4.3 Order of the Scheme

Figure 5 shows the numerical and analytical solutions for a particular grid size. There is also a plot of the error norm versus grid size. The order of the implementation is second order as expected. We made this comparison for conjugate gradient and SOR(not shown) and they were both the same as expected.

4.4 Smoothing

Figure 6 shows the initial behavior of the solution. The array of the solution was initialized with random numbers. We see that the highest frequencies decay away very quickly and then the low frequency modes of error are taking much longer to decay. This is just a visual check of what the theory says about the smoothing properties of the Gauss Seidel.

	No Preconditioning	Preconditioned
256	840	319
128	430	165
64	217	88
32	111	46

Table 3: Number of iterations to 10^{-10} convergence for conjugate gradient with and without pre conditioning

	No Preconditioning	Preconditioned
256	1.79	1.29
128	0.2	0.17
64	2.68E-2	2.18E-2
32	5.4E-3	4.86E-3

Table 4: Time to 10^{-10} convergence for conjugate gradient with and without pre conditioning

4.5 Information for benchmarking

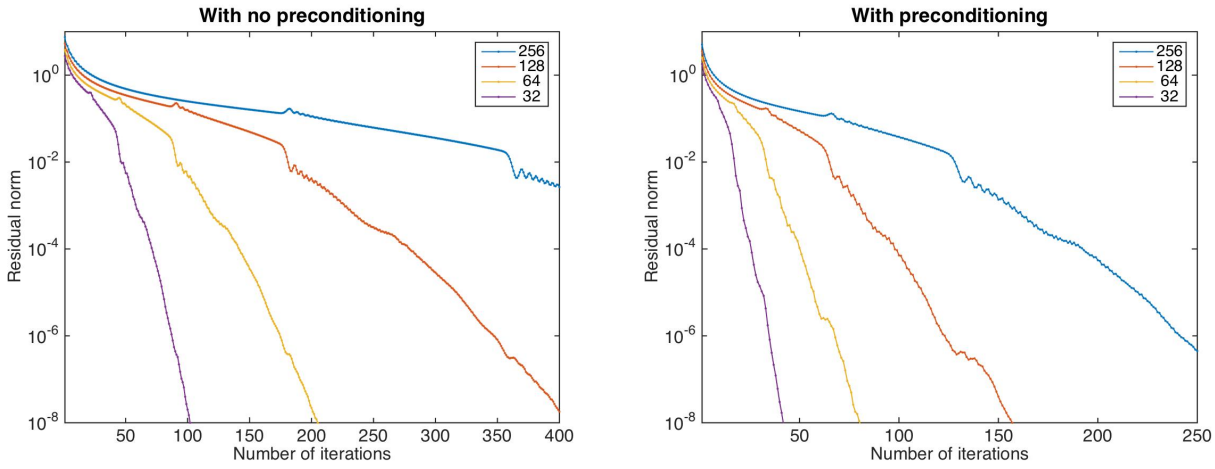
I used FORTRAN90 to code my algorithms and gfortran compiler. The computations were done on a 2.4GHz Intel Core i5 processor with 4GB of 1067Mhz DDR3 ram. The CPU times for the computations are presented in table 5.

Case	Time(s)
SOR 0.5	1471.9
SOR 1.0	569.795
SOR 1.5	202.639
Conjugate gradient without preconditioning	1.79
Conjugate gradient with preconditioning	1.29

Table 5: Times for convergence to 10E-10 on a 256X256 grid

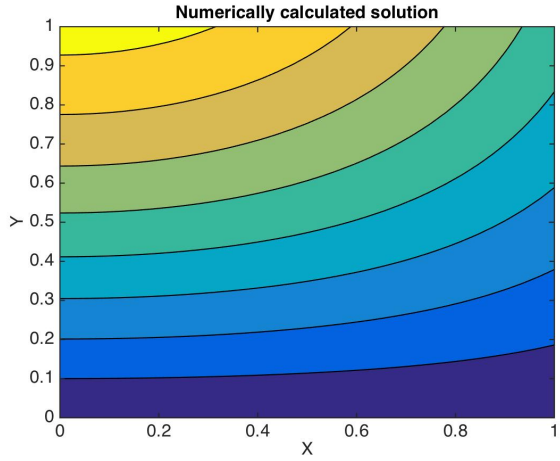
5 Conclusions

Poison equation with dirichlet boundary conditions is solved using the 5 point second order stencil using iterative methods. SOR, Gauss Seidel and conjugate gradient were the iterative methods implemented. The implementations were tested for different grid sizes and different values of ω. The order of the scheme was second order as expected. Conjugate gradient method is much faster than SOR. The preconditioned conjugate is faster than without preconditioning. The analysis for properties of the Gauss Seidel method was done analytically. The behavior(smoothing) is as expected.

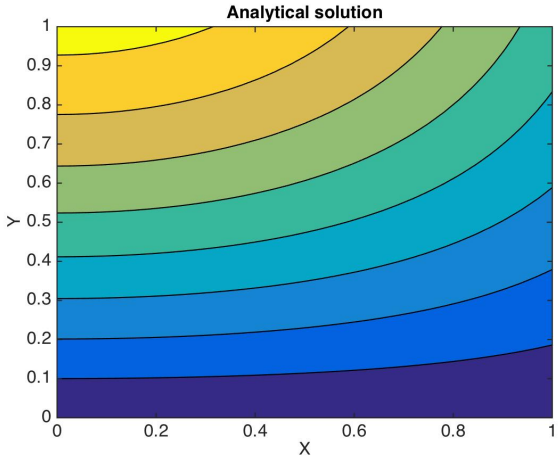


(a) (b)

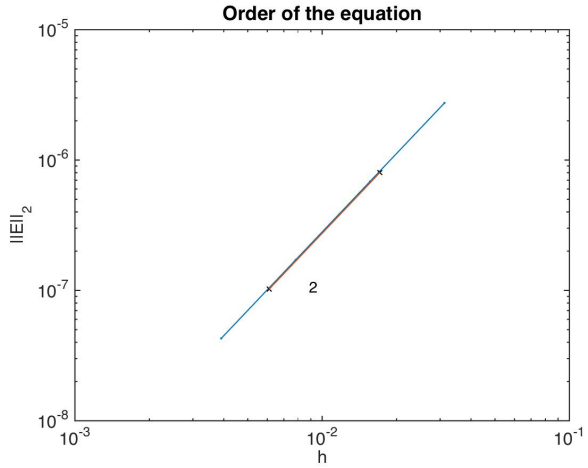
Figure 4: Behaviour of residue versus iteration for conjugate gradient



(a)

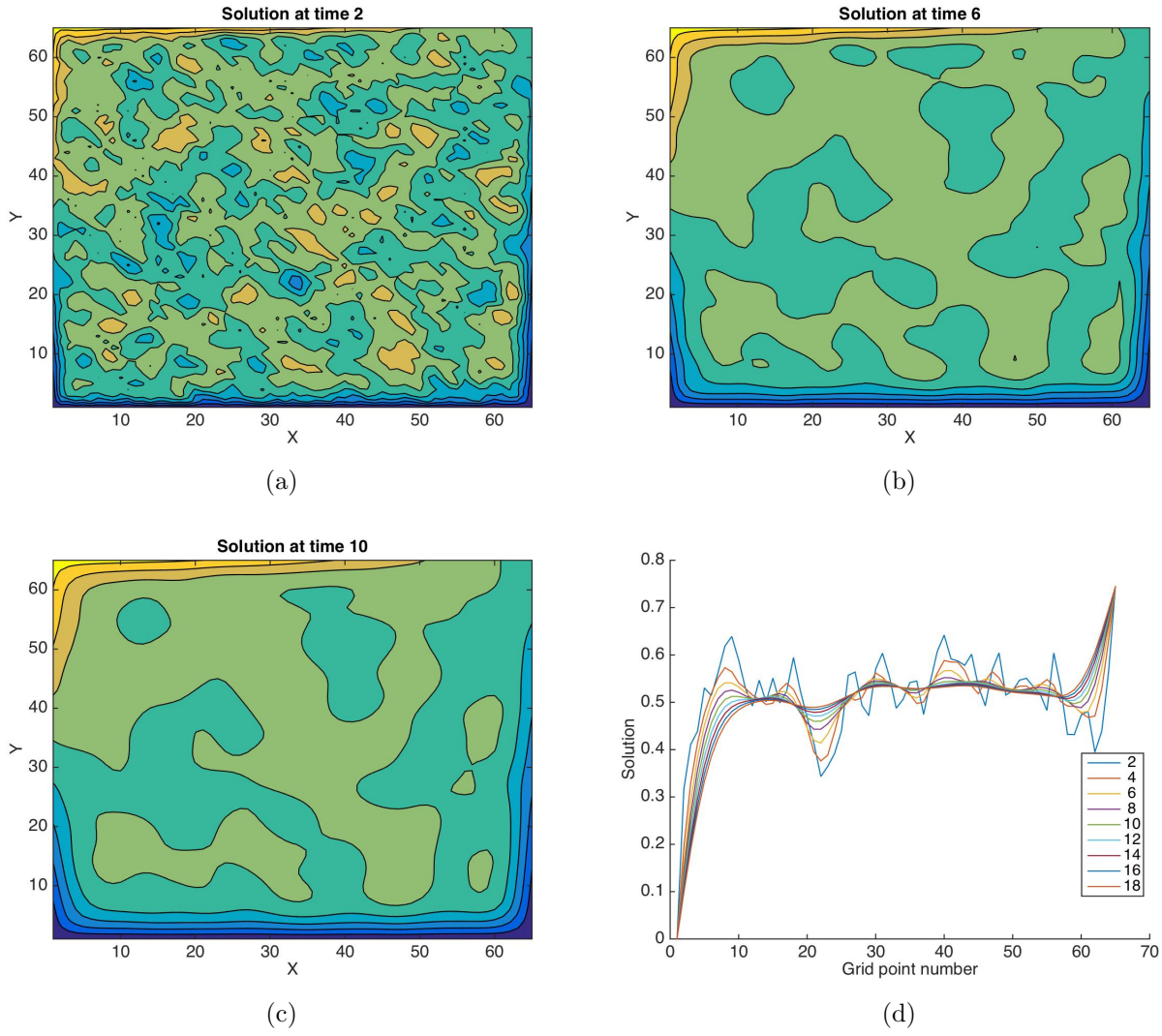


(b)



(c)

Figure 5: (a) and (b) show the numerical and analytical solutions calculated using conjugate gradient method on a 256 mesh point grid. (c) Error vs grid spacing plot to show the order of convergence.



(a) (b) (c) (d)

Figure 6: Behaviour of solution at initial iterations. Panel(d) shows the behavior of the solution at different iteration steps along a line in the domain.