1. Write a code to generate grids on planar regions using the “length algorithm” or Amsden-Hirt’s algorithm combined with SOR iteration. Your code should consist of one or more subroutines and a main program.

Details:

a) One of your subroutine should include at least the following parameters:
   \texttt{gridgenAH(n1,n2,\omega,\textit{tol},\textit{n})}, where
   \begin{itemize}
   \item \textit{n1}: Number of grid points in the \(\xi\)-direction.
   \item \textit{n2}: Number of grid points in the \(\eta\)-direction.
   \item \(\omega\): Relaxation parameter of SOR iteration.
   \item \textit{tol}: Tolerance used to establish stop criteria of SOR iteration.
   \item \textit{n}: Maximum number of iterations of SOR procedure.
   \end{itemize}

b) Write another subroutine: \texttt{gridquality(x,y,n1,n2)}.
   to report on the quality of the grids obtained. It should include computation of
   \(J_{i,j}\): the jacobian ,
   \(\theta_{i,j}\): angle between grid curves at every grid point \((x(i,j),y(i,j))\),
   \(\text{ADO}\): Average Deviation from Orthogonality, and
   \(\text{MDO}\): Maximum Deviation from Orthogonality.

c) Write a main program which calls subroutine \texttt{gridgenAH} and others. In this program you should define the data parameters needed by \texttt{gridgenAH} and by any other subroutine. You may also define the boundary transformation \(\partial T\) that transform the four sides of the rectangular computational domain into the boundaries of the physical domain and the branch cut. Also, an initial grid may be defined in this main program. This initial grid will be used as an initial guess to start the iterative algorithm defined in \texttt{gridgenAH}.

d) Apply your code for the Amsden-Hirt’s algorithm to the following simply connected physical domains:
   i) Dome (Convex Domain)
      Region enclosed by
      coordinate axes: \(x=0\), and \(y=0\),
      vertical line: \(x=1\), and
      curve: \(y = -4(x - 1/2)^2 + 2\)
ii) Swan Domain  
Region enclosed by  
coordinate axes: \( x = 0 \), and \( y = 0 \),  
curve: \( y = 1 - 2x + 2x^2 \),  
curve: \( x = 1 + 2y - 2y^2 \)

iii) Prototypical Antenna Domain  
Region enclosed by  
segment: \( y = 1 \), \(-3 \leq x \leq 0\),  
segment: \( y = 0 \), \(-3 \leq x \leq 0\),  
segment: \( x = 0 \), \( 1 \leq y \leq 3 + 1/2 \),  
segment: \( x = 0 \), \(-3 + 1/2 \leq y \leq 0\),  
semicircle: \( x \geq 0 \), \( x^2 + (y - 1/2)^2 = 3^2 \)

e) Perform three experiments for each domain:  
i) \( n_1 = 41, n_2 = 41 \); ii) \( n_1 = 21 \) \( n_2 = 61 \); iii) \( n_1 = 41, n_2 = 61 \).  
Use \( \text{tol} = 10^{-5} \), \( n = 5000 \) (if your problem needs more iterations to converge increase \( n \) as needed). Also, use the formula given in the notes to obtain the optimum \( \omega \) for the system of Laplace’s equation. Run your code for two other values of \( \omega \), record and report the number of iterations and verify that the theoretical optimum \( \omega \) indeed leads to a faster convergence of SOR. You can be more precise by making a graph of the number of iterations versus values of \( \omega \): \( N\text{It}_S(\omega) \).

f) Report your results in a table that includes:  
\((n_1 \times n_2)\): grid size,  
MDO: Maximum deviation from orthogonality,  
ADO: Average deviation from orthogonality,  
\(|J|_{\text{min}}\): Minimum absolute value of the jacobian,  
\((x_{\text{min}}, y_{\text{min}})\): Point where Jacobian is minimum or maybe zero. Find a way to determine if the Jacobian changes its sign. That will be an indicator of the presence of zero jacobian.  
\(\text{ItsTotal}\): Number of iterations.  
Make comments on your results. Try to establish a correlation between jacobian=0 and points of non-uniqueness.

g) Draw only one grid: \( 41 \times 61 \) for each domain. Ask MATLAB graphical subroutine to mark the points on the graph where the jacobian is minimum and where the deviation from orthogonality is the highest.
2. Write a code to generate grids on simply connected planar regions using the “Smoothness algorithm” or Winslow’s algorithm combined with SOR iteration. Your code should consist of one or more subroutines and a main program.

Details:

a) One of your subroutine should include at least the following parameters:
gridgenWinslowSC(n1,n2,ω,tol,n), with the parameters defined as in (1a).

b) Use the same subroutine “grigquality” defined in (1b) to compute the jacobian $J_{i,j}$, the angle $\theta_{i,j}$ between grid curves at every grid point $(x(i, j), y(i, j))$, and the grid properties ADO (Average Deviation from Orthogonality), and MDO (Maximum Deviation from Orthogonality).

c) Write a main program which calls subroutine gridgenWinslowSC and others and follow analogous instructions to those given in (1c).

d) Apply your code for Winslow’s algorithm to the same three physical domains defined in (1d).

e) Perform two experiments for each domain
i) $n1=21, n2=61$ and ii) $n1=41, n2=61$.
Use $tol = 10^{-5}$, $n = 5000$ (if your problem need more iterations to converge increase $n$ as needed). Determine a good relaxation parameter $\omega$ by trial and error. Run your code for some values of $\omega$, record and report in a table the number of iterations, and choose an empirical optimum $\omega$.

f) Report your results in a table similar to the one obtained in (1f), compare both and make comments.

g) Same as (1g). Compare both graphs and make comments.

3. Write a code to generate grids on multiple connected planar regions using the “Smoothness algorithm” or Winslow’s algorithm (adapted to domains with holes) combined with SOR iteration. Your code should consist of one or more subroutines and a main program.

Details:

a) As in the previous two cases, in your main program define the boundary conditions that form part of the BVP which determine the transformation $T$. However, in this case only two sides of the rectangle correspond to physical boundaries. On the other two sides of the rectangular computational domain impose continuity conditions (periodic conditions).

b) Modify the subroutine gridgenWinslowSC and create gridgenWinslowMC. Perform similar work as described in (2b) and (2c).
c) Apply your code for Winslow’s algorithm to the following multiple connected physical domains:

i) Three-Leafed Rose Domain

- Region bounded by
  - Three-leafed rose: \( x(t) = 0.3(2 + \cos 3t) \cos t, \quad y(t) = 0.3(2 + \cos 3t) \sin t, \quad 0 \leq t \leq 2\pi \)
  - outer circle: \( x^2 + y^2 = 2^2 \)

ii) Pacman Shaped Hole:

- Parametric equations:
  \[ x(\theta) = r(\theta) \cos(\theta) \quad \text{and} \quad y(\theta) = r(\theta) \sin(\theta), \]
  where
  \[ r(\theta) = \begin{cases} 
  2, & \text{if } 0 \leq \theta < \frac{5\pi}{6}; \\
  \frac{1}{1-\sqrt{3}} \left( \frac{1}{\sin(\theta)} - \frac{1}{1-\sqrt{3}} \cos(\theta) \right), & \text{if } \frac{5\pi}{6} \leq \theta < \pi; \\
  -\frac{1}{1-\sqrt{3}} \left( \frac{1}{\sin(\theta)} + \frac{1}{1-\sqrt{3}} \cos(\theta) \right), & \text{if } \pi \leq \theta < \frac{7\pi}{6}; \\
  2, & \text{if } \frac{7\pi}{6} \leq \theta \leq 2\pi. 
\] (1)

- And outer circle: \( x^2 + y^2 = 5^2 \)

d) Perform two experiments for each domain: i) \( n_1 = 71, \quad n_2 = 21 \); ii) \( n_1 = 71, \quad n_2 = 41 \).

Use \( tol = 10^{-5}, \quad n = 5000 \) (if your problem needs more iterations to converge increase \( n \) as needed).

e) Report your results in a table similar to the one obtained in (1f). Make comments on your results.

f) Draw only one grid: \( 71 \times 41 \) for each domain.

**Recommendations:**

i) Read carefully all the items and try to give answer to all of them. In this work, we are trying to analyze the properties of different grids, make sure you explain different and analogies in your comments.

ii) I may have overlooked something in the problems proposed, if you think so feel free to add anything necessary.

iii) I have carefully worked out all the parts in the project myself. I think this is reasonable and doable in these 10 days. However, you will need TO START EARLY. I will be more than willing to clarify questions as you go along.

iv) THIS IS AN INDIVIDUAL PROJECT! You can use any resource available to you except your classmates. Make an effort to write your own algorithm and obtain your very own answer. Two individuals very rarely write the same code (programming is very personal).

“You are capable of something better. Give it your best. You will never again will have such an opportunity. Pray about it. Work at it. Make it happen.”

Pres. Hinckley at the Inauguration of Pres. Samuelson