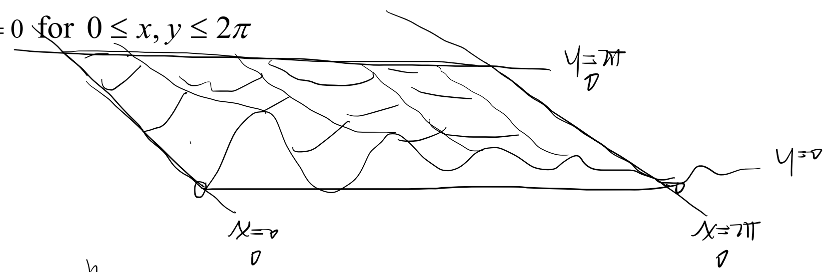
**EN 530.766**
**Fall 2023**
**HW 4**

$\frac{P^{k+1}_{i,j}-P^{k}_{i,j}}{\Delta t} + \frac{1}{\Delta^2}[P_{i-1,j}+P_{i+1,j}+P_{i,j-1}+P_{i,j+1}-2P_{i,j}] = 0$

er the 2-D Laplace equation $\frac{\partial P}{\partial t} + \nabla^2 P = 0$

$$\left(u_{xx}+u_{yy}\right)=0 \text{ for } 0 \le x, y \le 2\pi$$

with the following boundary conditions




BC
$$u(0,y)=0$$
$$u(2\pi,y)=0$$
$$u(x,0)=\sin(2x)+\sin(5x)+\sin(7x)$$
$$u(x,2\pi)=0$$

R

$\Sigma|r| = \sum_i\sum_j \sqrt{(r_{ij})^2}$

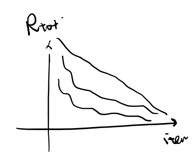
$||r|| = \sqrt{\sum_j\sum_i (r_{ij})^2}$

IC = 0

IC =

IC = random

$r^k = \left(\frac{P_{i-1,j}+P_{i+1,j}+P_{i,j-1}+P_{i,j+1}-4P_{i,j}}{\Delta^2}\right)$


(1) Write a computer code and obtain the numerical solution using the point Jacobi and Gauss-Seidel iterative schemes. Use a mesh with $\Delta x = \Delta y = 2\pi/20$. Track the convergence by calculating the residual, $r^k = \left(\delta_x^2/\Delta x^2 + \delta_y^2/\Delta y^2\right)u_{i,j}^k$. Conduct these simulation with two different initial guesses (1) u(i,j) =0; (2) u(i,j)=$x_i$ $y_j$; and (3) u(i,j)=random number distribution between -1 and 1.

$\epsilon < 10^{-4}$

$10^{-5}$ $10^{-4}$

accepted value,

$R_{tot}$ $10^{-6}$ $10^{-7}$

Does the convergence behave as expected? Discuss.

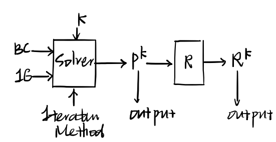
$[\ r^k_{ij}\ ]$

(2) Try the above problem with the SOR point Jacobi and SOR point Gauss-Seidel schemes: Investigate and comment on the convergence properties for various values of the under- and over-relaxation parameter for both schemes. Can you find optimal values of the relaxation parameter for these schemes?

(3) Experiment with a SOR Jacobi method where you alternate between an overrelaxation and an underrelaxation as you iterate. Can you find a pair of relaxation parameters that speed up the solution process compared to the non-relaxed Jacobi method? For more information about this "Scheduled Relaxation Jacobi" Method check out this paper. *Xiang Yang and Rajat Mittal, "Acceleration of the Jacobi iterative method by factors exceeding 100 using scheduled relaxation", Journal of Computational Physics, Vol 274, DOI: 10.1016/j.jcp.2014.06.010.*

(4) Challenge problem – how about experimentally determining a SRJ scheme with 3 different values of the relaxation parameter. How much faster can you get compared to the 2 parameter SRJ scheme.

*https://www.dropbox.com/s/sf6qssxurtn48pp/2014-Acceleration%20of%20the%20Jacobi%20iterative%20method%20by%20factors%20exceeding%20100%20using%20scheduled%20relaxation.pdf?dl=0*



$\nabla^2 P = 0$ Elliptic

$\nabla^2 P = \frac{\partial P}{\partial t}$

$\frac{P_{i,j}^{k+1} - P_{i,j}^{k}}{\Delta t} = \left[\frac{\delta_x^2 + \delta_y^2}{\Delta^2}\right] P_{ij}$

$P_{i,j}^{k+1} - P_{i,j}^{k} = \frac{\Delta t}{\Delta^2}\left[P_{i-1,j} + P_{i+1,j} + P_{i,j-1} + P_{i,j+1} - 4P_{i,j}^{k}\right]$

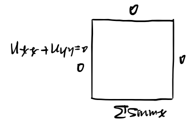
For stability. Dim = 2, $r = \frac{\Delta t}{\Delta^2} \le \frac{1}{4}$, let $\frac{\Delta t}{\Delta^2} = \frac{1}{4}$

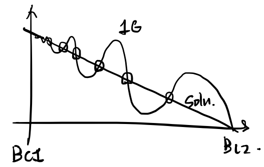
$P_{ij}^{k+1} = \frac{1}{4}\left[P_{i-1,j} + P_{i+1,j} + P_{i,j-1} + P_{i,j+1}\right]$

Jacobi: $P_{ij}^{k+1} = \frac{1}{4}\left[P_{i-1,j}^{k} + P_{i+1,j}^{k} + P_{i,j-1}^{k} + P_{i,j+1}^{k}\right]$

Gauss-Sadiel: $P_{ij}^{k+1} = \frac{1}{4}\left[P_{i-1,j}^{k+1} + P_{i+1,j}^{k} + P_{i,j-1}^{k+1} + P_{i,j+1}^{k}\right]$

$r^k = \frac{1}{\Delta^2}\left[P_{i-1,j}^{k} + P_{i+1,j}^{k} + P_{i,j-1}^{k} + P_{i,j+1}^{k} - 4P_{i,j}^{k}\right]$



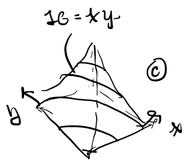



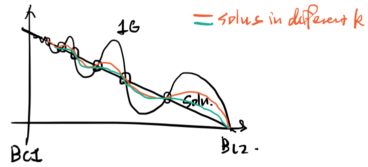

(a) IC = 0. R wave number

k soln = $k_E = k_{BC}$

(c) IC = random.

$E = I_C - u_{Soln}$

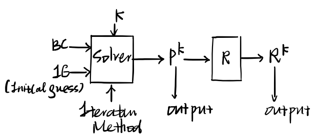


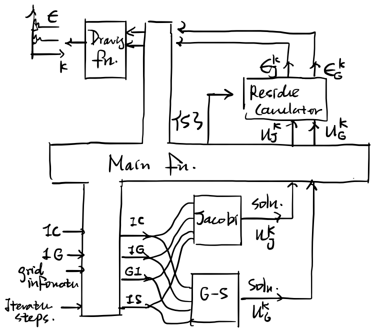




(1) Write a computer code and obtain the numerical solution using the point Jacobi and Gauss-Seidel iterative schemes. Use a mesh with $\Delta x = \Delta y = 2\pi/20$. Track the convergence by calculating the residual, $r^k = \left(\delta_x^2/\Delta x^2 + \delta_y^2/\Delta y^2\right)u_{i,j}^k$. Conduct these simulation with two different initial guesses (1) u(i,j) =0; (2) u(i,j)=$x_i$ $y_j$; and (3) u(i,j)=random number distribution between -1 and 1.

Does the convergence behave as expected? Discuss.

Algorithm:




Grid Information

$x_{max} = 2\pi$

$y_{max} = 2\pi$

$\Delta = \frac{2\pi}{20}$

$\Delta x = \Delta$

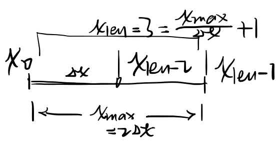
$\Delta y = \Delta$

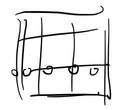
~~$\Delta t = \frac{\Delta^2}{4}$~~

$k = 1000$

$x_{len} = \frac{x_{max}}{\Delta x} + 1$

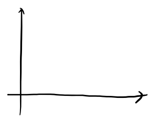
$y_{len} = \frac{y_{max}}{\Delta y} + 1$



$R_{ij} =$

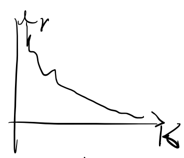
$R_{tot} = \sum_{i,j}^{n,n} |r_{ij}|$



P[0, $y_{len}$][0, $x_{len}$] = [0]

BC -
```
for i in range[0, x_len - 1]:
    P[0][i] = sin 2iΔx + sin 5iΔx + sin 7iΔx
    P[y_len - 1][i] = 0
for j in range[0, y_len - 1]:
    P[j][0] = 0
    P[j][x_len - 1] = 0
```

```
Jacobi ( P_input, ~~Δx, Δy, Δt~~, x_len, y_len,    )
    P_old = P_new = P_input
    for j ∈ [1, y_len - 2]:
        for i ∈ [1, x_len - 2]:
            P_new ij = 1/4 [ P_old i-1,j + P_old i+1,j + P_old i,j-1 + P_old i,j+1 ]
    return P_new ij
```



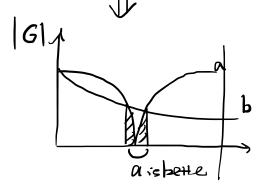

$\Downarrow$

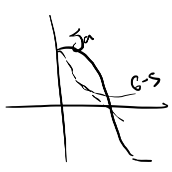
IG:
```
IG = 0
for j in range[1, y_len - 2]:
    for i in range[1, x_len - 2]:
        P[j][i] = IG
```

```
P_input = P
for t in range[0, k]:
    P_Jacobi = Jacobi ( P_input, ~~Δx, Δy~~,    , x_len, y_len,    )
    P_GS = G-S ( P_input, ~~Δx, Δy~~,    , x_len, y_len,    )
    r = P

    r_J = Res( P_Jacobi
    r_G-S = Res( P_G-S
```

```
G-S ( P_input, ~~Δx, Δy, Δt~~, x_len, y_len,    )
    P_G = P_input
    for j ∈ [1, y_len - 2]:
        for i ∈ [1, x_len - 2]:
            P_G ij = 1/4 [ P_G i-1,j + P_G i+1,j + P_G i,j-1 + P_G i,j+1 ]
    return P_G ij
```



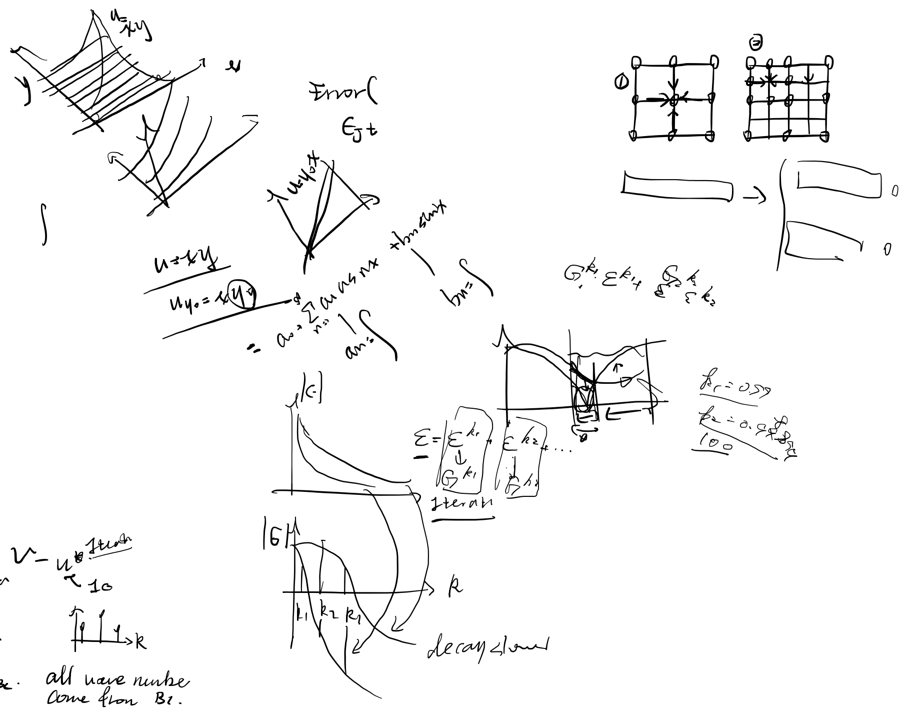
$\epsilon_J[t] = \text{Error}(r_{\text{Jacobi}}, \ldots)$

$\epsilon_{G-S}[t] = \text{Error}(P_{G-S}, r_{G-S})$

ePlotting (          )

```
Res.(r, P, Δ, xlen, ylen):
  rc = r
  for j ∈ [0, ylen-1]:
    for i ∈ [0, xlen-1]:
      rc[j][i] = (P[i-1,j] + P[i+1,j] + P[i,j-1] + P[i,j+1] - 4P[i,j])/Δ²
  return rc
```

```
Error(r, xlen, ylen):
  e = 0
  for j ∈ [0, ylen-1]:
    for i ∈ [0, xlen-1]:
      e = e + |r[i][j]|
  return e;
```




1. $\epsilon_{\text{Iteration}} = v - u^{\text{Iteration}}$

① If $u^0 = 0$.

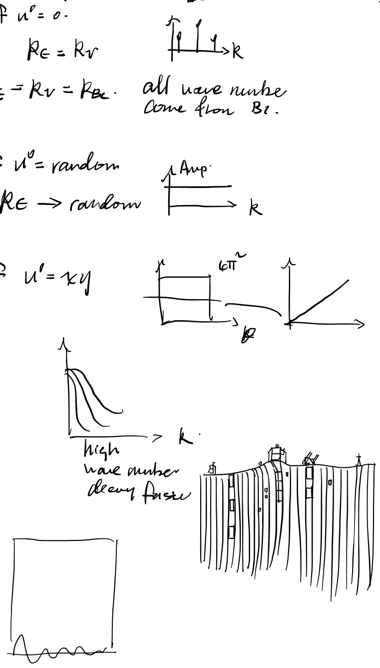
$k_\epsilon = k_v$

$k_\epsilon = k_v = k_{BC}$. all wave number come from BC.

② If $u^0$ = random

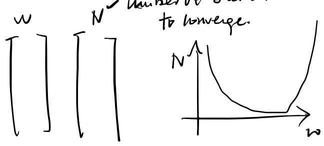
$k_\epsilon \to$ random

③ If $u^0 = xy$




optimization

2. $w$   $N$ ✓ number of iterations to converge.




3. use 2 w.

$w_1$ overrelaxation   1.1   Switch.   Improve Jacobi

$w_2$ underrelaxation   0.9

1.2 1.3 1.4

0.8 0.7 0.6 0.5

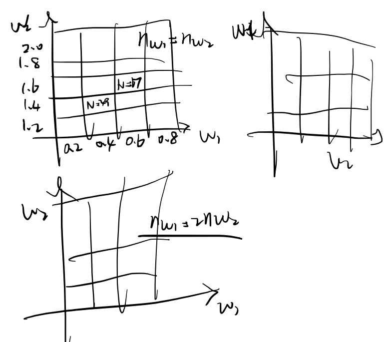
① $w_1$

② $w_2$

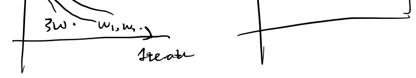
③ ~~times~~

number

④ order




4.



(2) Try the above problem with the SOR point Jacobi and SOR point Gauss-Seidel schemes: Investigate and comment on the convergence properties for various values of the under- and over-relaxation parameter for both schemes. Can you find optimal values of the relaxation parameter for these schemes?

$$P^* = P^{k+1}_{old}$$

SOR Method:

$$P^{k+1} = (1-\omega)P^k + \omega P^*$$

For Jacobi, $P^*$ is

$$P^* = \frac{1}{4}\left[P^k_{i-1,j} + P^k_{i+1,j} + P^k_{i,j-1} + P^k_{i,j+1}\right]$$

Gauss-Seidel, $P^*$ is

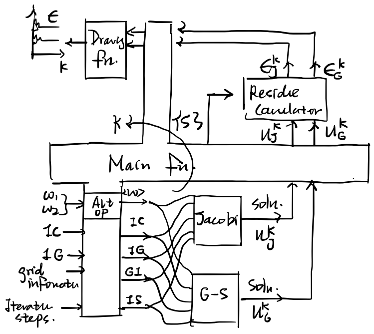
$$P^* = \frac{1}{4}\left[P^{k+1}_{i-1,j} + P^k_{i+1,j} + P^{k+1}_{i,j-1} + P^k_{i,j+1}\right]$$

Then, the improved SOR Jacobi

$$P^{k+1} = (1-\omega)P^k + \frac{\omega}{4}\left[P^k_{i-1,j} + P^k_{i+1,j} + P^k_{i,j-1} + P^k_{i,j+1}\right]$$

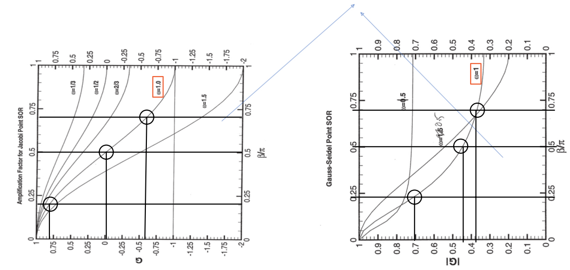
Similiar, get SOR Gauss-Seidel:

$$P^{k+1} = (1-\omega)P^k + \frac{\omega}{4}\left[P^{k+1}_{i-1,j} + P^k_{i+1,j} + P^{k+1}_{i,j-1} + P^k_{i,j+1}\right]$$




## G-S Point SOR

Smoother!



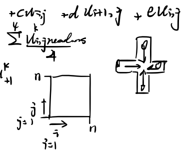
Least Error

$u_{xx} + u_{yy} = 0$

Jacobi $a u_{i-1,j} + b u_{i,j-1} + c u_{i,j} + d u_{i+1,j} + e u_{i,j+1} = 0$

Jacobi $u^{k+1}_{i,j} = \frac{\sum^{4} u^k_{i,j\,\text{neighbours}}}{4}$

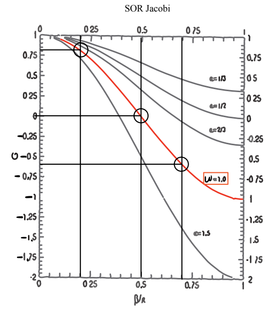
GS $u^{k+1}_{i,j} = \frac{\sum^{2} u^{k+1}_{-1} + \sum u^k_{+1}}{4}$



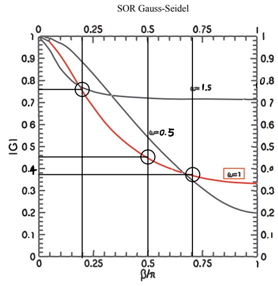
$k \cdot \Delta x$

$\frac{2\pi}{20}$

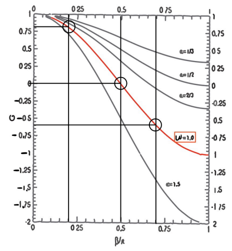
SOR Jacobi



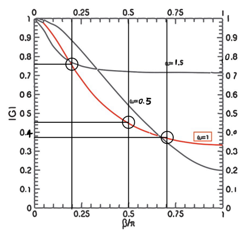
SOR Gauss-Seidel



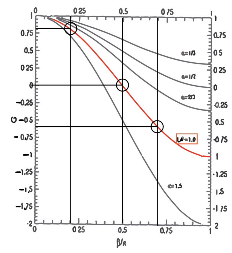
SOR Jacobi



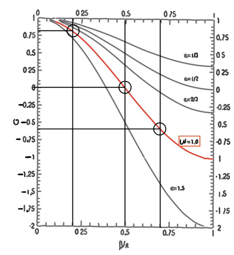

SOR Gauss-Seidel



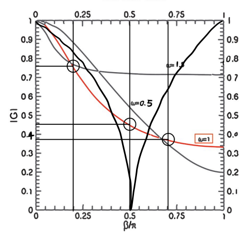

SOR Jacobi



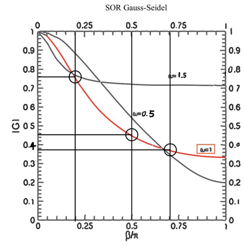

SOR Jacobi

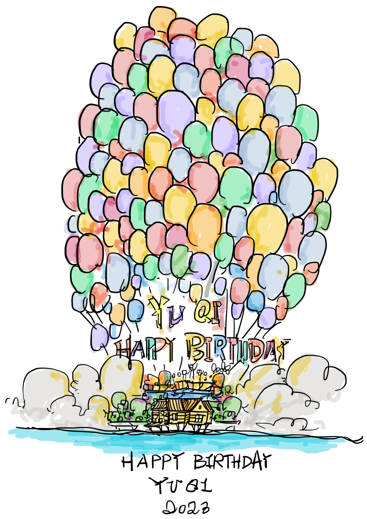



SOR Gauss-Seidel

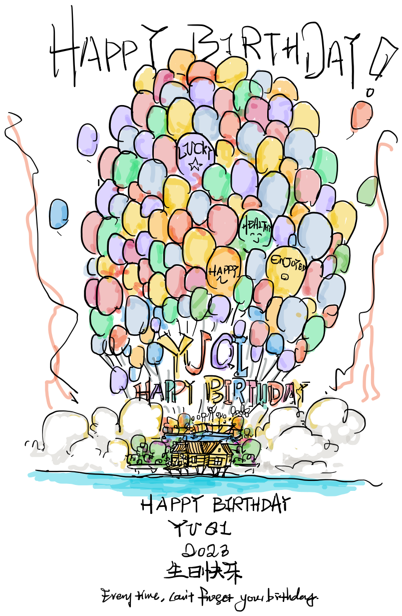



SOR Gauss-Seidel




YU QI
HAPPY BIRTHDAY
HAPPY BIRTHDAY
YU QI
2023


YU QI
HAPPY BIRTHDAY


HAPPY BIRTHDAY!
LUCKY
HEALTHY
HAPPY
YUQI
HAPPY BIRTHDAY
HAPPY BIRTHDAY
YU QI
2023
生日快乐
Every time, can't forget your birthday.