a) -

b) -

c) -

d)
Observació: En els següents exercicis parlarem de convergència o divergència. En aquesta observació aclarirem el criteri que hem utilitzat en tota la pràctica.
En dividirem tres casos diferents:

- La cota de l'error absolut és menor que la tolerància amb menys iteracions de les màximes. -> Direm que convergeix.
- La cota de l'error absolut no és menor que la tolerància després d'arribar a la iteració màxima. -> Direm que no convergeix.
- La cota de l'error absolut és negativa. -> Direm que no convergeix.
  - Si la cota és negativa vol dir que l'error de la iteració actual és més gran que la anterior, és a dir, que aquesta iteració empitjora l'aproximació.

Jacobi: Convergeix en 2504 iteracions.

Gauss-Seidel: Convergeix en 1273 iteracions.

e)

Jacobi: Divergeix en 184144 iteracions.

Gauss-Seidel: Divergeix en 101183 iteracions.

f)

Sense haver d'executar el programa ja podem assegurar que per a $\omega = -5$ i $\omega = 5$ el mètode no convergirà, doncs de teoria tenim el resultat de que per a que SOR convergeixi $0 \leq \omega \leq 2$. Tot i així hem executat el programa per aquests valors i, evidenment, hem vist que no convergeix.

ω	Convergència?	Num iteracions
-5	No, la cota de l'error absolut és negativa	2
0.1	Si	23373
0.2	Si	11343
0.3	Si	7180
1	Si	1273
1.8	No, la cota de l'error absolut és negativa	2
1.9	No, la cota de l'error absolut és negativa	2
5	No, la cota de l'error absolut és negativa	2

Observació: Els casos d'ω en que el mètode SOR divergeix i hem posat que ho fa en 2 iteracions, és degut a que en la primera no calculem la fita de l'error absolut doncs encara no tenim la de la iteració anterior!

g)

Les conclusions sobre els casos de ω=-5 i ω=5 són les mateixes que en l'apartat anterior. Després de comprovar-les, hem vist que evidentment el mètode divergia en aquests casos.

Al parlar de convergència dividirem els tres mateixo casos que en l'apartat anterior:

ω	Convergència?	Num iteracions
-5	No, la cota de l'error absolut és negativa	2
0.1	No convergeix ja que amb les 300.000 iteracions no aconsegueix una cota de l'error absolut inferior a la demanada.	300.000
0.2	No convergeix ja que amb les 300.000 iteracions no aconsegueix una cota de l'error absolut inferior a la demanada. Tot i així, arriba a una cota millor que per ω=0.1	300.000
0.3	No convergeix ja que amb les 300.000 iteracions no aconsegueix una cota de l'error absolut inferior a la demanada. Tot i així, Arriba a una cota millor que per ω=0.2	300.000
1	No, la cota de l'error absolut és negativa.	101.183
1.8	Si.	13.370
1.9	Si.	6372
5	No	2

Observació: Els casos d'ω en que el mètode SOR divergeix i hem posat que ho fa en 2 iteracions, és degut a que en la primera no calculem la fita de l'error absolut doncs encara no tenim la de la iteració anterior!

h) Si p(x)=0 llavors la matriu és tridiagonal dominant per files.

És molt fàcil de veure, recordem la matriu que estem iterant:

$$a_i = -\frac{1}{2}\left[1+\frac{h}{2}p(x_i)\right],\; b_i = \left[1+\frac{h^2}{2}q(x_i)\right],\; c_i = -\frac{1}{2}\left[1-\frac{h}{2}p(x_i)\right]$$

$$A = \begin{pmatrix} b_1 & c_1 & & & 0 \\ a_2 & b_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & a_{n-1} & b_{n-1} & c_{n-1} \\ 0 & & & a_n & b_n \end{pmatrix}.$$

Amb p(x)=0 tenim que ai=-1/2, ci=-1/2 bi>1, doncs q(x) és l'exponencial, que sempre és positiva diferent de zero i h^2 també ho és. Llavors és directe que bi>ai+ci per tant és tridiagonal dominant per files en sentit estricte.

De la proposició 1.2.2 dels apunts de JCTatjer, amb aquesta hipòtesi en tenim prou per a concloure la convergència del mètode de Jacobi.

De la proposició 1.2.3 dels apunts de JCTatjer, amb aquesta hipòtesi en tenim prou per a concloure la convergència del mètode de Gauss-Seidel.

Com compilar i executar el programa.

- **To compile:**
  `gcc -Wall -ansi -std=c99 -c -pedantic program.c`
- **To link:**
  `gcc program.o -o program.exe -lm`
- **To execute:**
  - For (d):
    `./program.exe 0 6.283185307179586 100 0 0 3000 1e-10 > program100.out`
  - For (e);
    `./program.exe 0 6.283185307179586 1000 0 0 250000 1e-10 > program1000.out`
  - For (f):
    `./program.exe 0 6.283185307179586 100 0 0 30000 1e-10 'w' > sor30000_'w'.out`
  - For (g):
    `./program.exe 0 6.283185307179586 1000 0 0 300000 1e-10 'w' > sor300000_'w'.out`
    where program is the name of the file and 'w' the chosen omega value.