Universidad de Chile
Facultad de Ciencias Físicas y Matemáticas
Departamento de Ciencias de la Computación

# Tarea 1:

Comportamiento térmico inducido por planta industrial

| | |
|---|---|
| Integrantes: | Rodrigo Fuentes |
| Profesora: | Nancy Hitschfield |
| Auxiliares: | Mauricio Araneda |
| | Pablo Pizarro |
| | Pablo Polanco |
| Ayudantes: | María José Trujillo |
| | Ivan Torres |

Fecha de entrega: 22 de Abril del 2018

# Índice de Contenidos

# 1. Descripción del problema

En términos generales, el problema consiste en modelar el comportamiento térmico de una costa cualquiera inducido por una planta industrial que se instala en esta. Para eso, se utilizará el método numérico de sobrerelacación sucesiva.

En particular, el problema se puede dividir en 3 subproblemas:

1. Modelar el terreno, consistente en un corte transversal en dirección este-oeste que posee mar, una pequeña costa y dos grandes montañas.
2. Fijar las condiciones de borde, diferentes para cada terreno.
3. Iterar sobre el modelo, utilizando el método de sobrerelajación sucesiva, según las condiciones de temperatura que este tenga.

Los principales objetivos del trabajo realizado son estudiar el funcionamiento de la sobrerelajación sucesiva, y lograr tener resultados experimentales consistentes con lo esperado teórica e intuitivamente.

# 2. Generación del terreno

## Que hay que modelar

Como se dijo anteriormente, se busca generar un corte transversal de este a oeste de una costa. En total, el corte tiene un ancho de 4[km] y un alto de 2[km]. En la figura (1) se puede ver el terreno y las medidas en detalle.

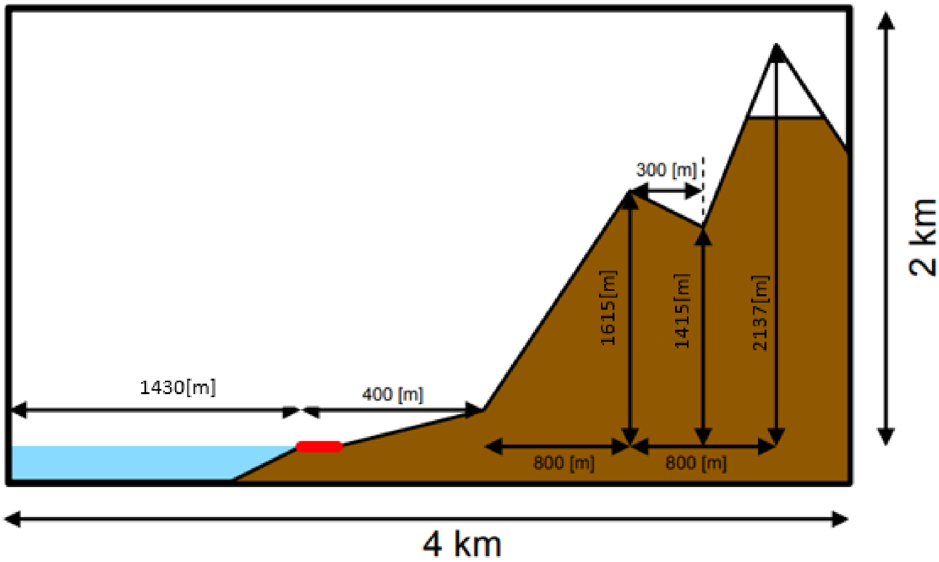


Figura 1: Corte transversal de costa a modelar. En celeste: agua, en rojo: Planta industrial, en café: Tierra, en blanco: Aire y nieve

## Del papel al computador

Se decide modelar el problema utilizando programación orientada a objetos, puesto que es más simple y fácil de extender o modificar.

Para esto, se crea la clase corte. Esta tiene numerosas variables, pero las más importantes se reducen a:

- matrix: Matriz que para cada posición, representa la temperatura a la que está dicha locación en el corte.
- ancho: Ancho de la matriz en metros. Para efectos de la tarea es un valor constante de 4000.
- alto: Análogo a la anterior. En este caso el valor es de 2000 + alturaMar, esta última representa a que altura estará el mar. Esto solo para que sea visible en el plot.
- dh: Cantidad de metros cuadrados que representa cada celda. Si dh es 20, la matriz sera de 200 x 100.
- El resto de las variables de instancia se pueden ver en el constructor de la clase corte.

Es importante notar, que para efectos de la tarea, lo único que diferencia un tipo de terreno de otro es la temperatura que tiene, por lo que basta con la matriz $matrix$ para reprensentar el corte. Para las iteraciones se utiliza otra matriz para diferenciar el aire del resto, pero esto se explica más adelante.

Una vez inicializadas las variables, es necesario fijar las condiciones iniciales de temperatura para cada tipo. Para esto, se utilizan dos métodos de la clase corte: $primeraColumna(self)$ y $fijarCondicionesBorde(self)$:

- $primeraColumna(self)$: Se encarga de fijar las condiciones iniciales para la primera columna de aire. Esto es importante puesto que dada una altura, todas las celdas de aire a esa altura tienen la misma temperatura, por lo tanto al fijar la primera columna, el resto se puede igualar a esta.

- $fijarCondicionesBorde(self)$: Se encarga de fijar las condiciones de borde para cada segmento del corte. Estos son 7: El mar, la planta, y luego las 5 rectas que forman las montañas. Para cada segmento, se ve si cada celda esta por sobre o por debajo la recta divisoria y se le asigna la temperatura correspondiente.

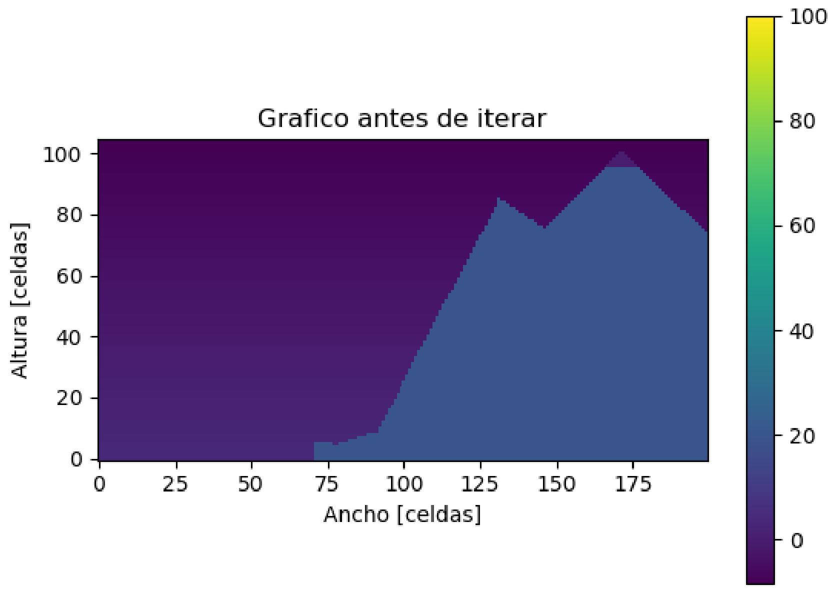


Figura 2: Imagen del modelo sin iteraciones. Por temas de visibilidad, la planta todavía no ha sido agregada

# 3. Solución numérica

## Saber sobre que iterar

Una vez que se tienen las condiciones iniciales del sistema, es importante definir sobre que se quiere iterar. En este caso, como las condiciones de la tarea dicen que a grandes distancias la planta no afecta a la atmósfera. Se asume que los bordes de la matriz se mantendrán a la misma temperatura que el momento inicial.

Con lo anterior, se tiene que los únicos puntos sobre los que se tiene que iterar son aquellos que pertenezcan a la atmósfera y no estén al borde del modelo. Esto es muy conveniente porque todos esos puntos se iteran de la misma forma, por lo que no hay que incluir casos de borde como esquinas u orillas.

Anteriormente se dijo que lo único que diferenciaba dos tipos de terreno distintos era la temperatura que tenían, y si bien esto es cierto, para iterar se hace necesario diferenciar lo que es aire de lo que no lo es. Para esto, se utiliza otra de las variables de instancia de la clase corte, *matrixTipos*. Esta es una matriz del mismo tamaño que *matrix* pero que en cada posición guarda un -1 si es que la posición correspondiente es aire, o un 0 si no lo es. Esto sirve luego para saber sobre que celdas iterar.

## La sobrerelajación sucesiva

Primero que todo, es importante recordar la ecuación de diferencias de laplace, dada por:

$$u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j} = 0 \tag{1}$$

Esta se puede escribir de la siguiente forma, adecuada para iterar:

$$u_{i,j} = u_{i,j} + r_{i,j} \tag{2}$$

Con:

$$r_{i,j} = \frac{u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j}}{4}$$

Notar que para poder iterar con esta fórmula, es necesario, para una celda, tener los valores de sus 4 celdas adyacentes. En los casos de borde esto no se cumple, y se hace necesario modificar un poco la ecuación. Por eso es conveniente no tener que incluir estos casos.

Luego, teniendo la ecuación (2), el objetivo de la iteración es que $r_{i,j}$ se haga menor a una tolerancia dada. El método de sobrerelajación sucesiva hace que la velocidad de convergencia de las iteraciones aumente, esto lo hace multiplicando el $r_{i,j}$ por un factor $\omega$, llamado factor de relajación. Este varía entre 1 y 2, y encontrarlo no es tan trivial, puesto que depende de los coeficientes y dimensiones de la matriz.

Con esto, la ecuación queda de la forma:

$$u_{i,j} = u_{i,j} + \omega r_{i,j} \tag{3}$$

Notar que si $\rho = 0$, $r_{i,j}$ es el ya mencionado anteriormente Sin embargo, para extender la fórmula cuando $\rho$ no es cero, se tiene:

$$r_{i,j} = \frac{u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j} - h^2\rho(dx, dy)}{4}$$

Con $dx$ y $dy$ las distancias del punto hacia el centro de planta en las coordenadas x e y respectivamente.

## Iterando

Una vez que se tienen las formulas a utilizar y se sabe que hacer, es necesario traspasarlas al programa en forma de métodos para la clase corte.

Los métodos utilizados para iterar fueron tomados de la clase auxiliar 3, de Pablo Pizarro. Estos consisten en 3:

- *_single_iteration*(...): Se encarga de realizar una iteración. Es decir, calcula la nueva matriz dada la que ya se tiene.
- *_convergio*(...): Se encarga de checkear si la iteración convergio, dada la diferencia entre la matriz antes y después de iterar una vez.
- *start*(...): Utilizando los dos métodos recién mencionados, llama a *_single_iteration*(), llamando a *_convergio* continuamente. Cuando finalmente la iteración converge, imprime la cantidad de iteraciones que demoró y el margen de error que se obtuvo. Es importante notar que los dos métodos anteriormente mencionados sólo son llamados por *start*(), por lo que son privados.

# 4. Resultados

Primero que todo, es importante mencionar que todos los resultados que a continuación se muestran, fueron obtenidos con una discretización de 20 metros (dh = 20) y una tolerancia de 0.1, estos valores se consideran adecuados dadas las grandes dimensiones del problema. Además, por temas de visibilidad, los plots están hechos con el parámetro vmax = 100, esto hace que todas las temperaturas que son superiores a dicho valor, se muestren del mismo color (de lo contrario, las temperaturas cercanas a la planta son tan altas que por contraste las montañas apenas se reconocen)

## Iteraciones con $\rho = 0$

A continuación se presentan los gráficos obtenidos de hacer las iteraciones con $\rho = 0$. Se decidió estudiar las variaciones del numero de iteraciones respecto a $\omega$ para T = 12. Para todos los demás tiempos, el $\omega$ es el óptimo calculado con la fórmula dada (1.953).

De la figura (3) hasta la (7): Gráficos para $\omega$ óptimo. De la (8) hasta la (11): Gráficos para T=12 con diferentes $\omega$.

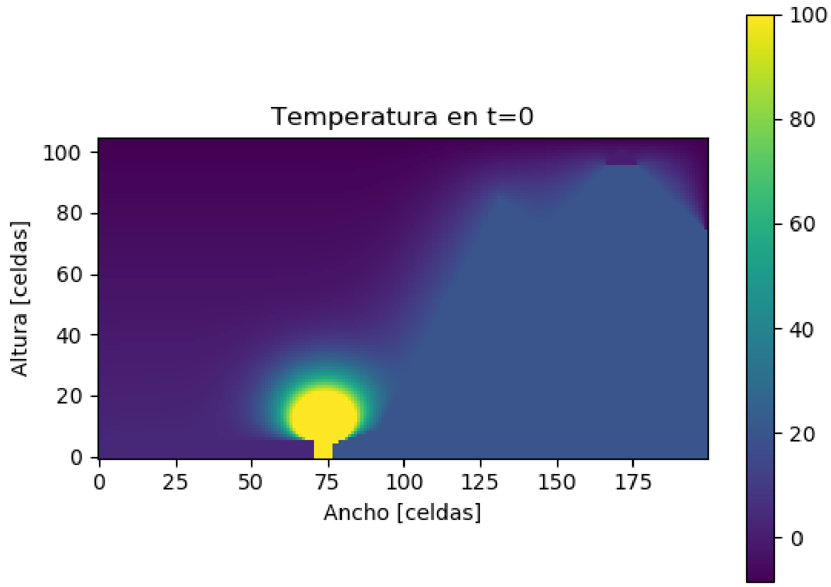


Figura 3: Cantidad de iteraciones: 531

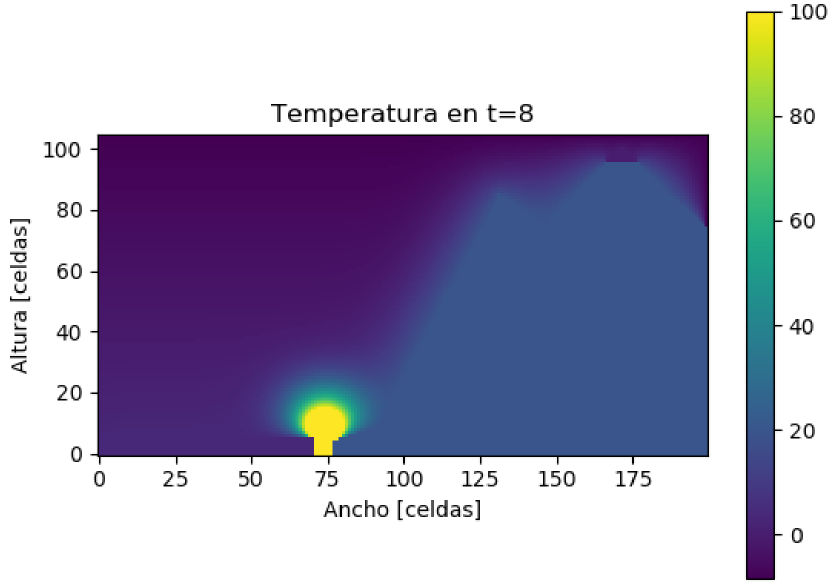


Figura 4: Cantidad de iteraciones: 345

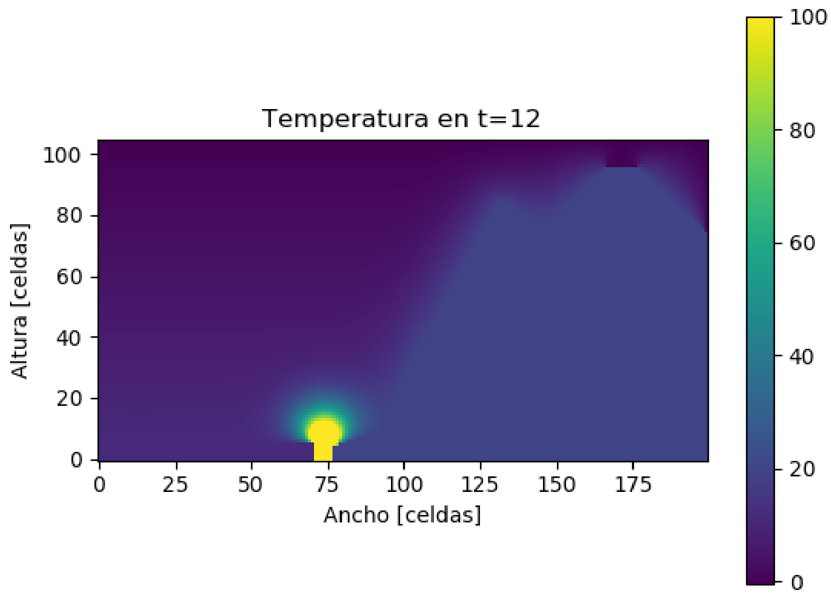


Figura 5: Cantidad de iteraciones: 256

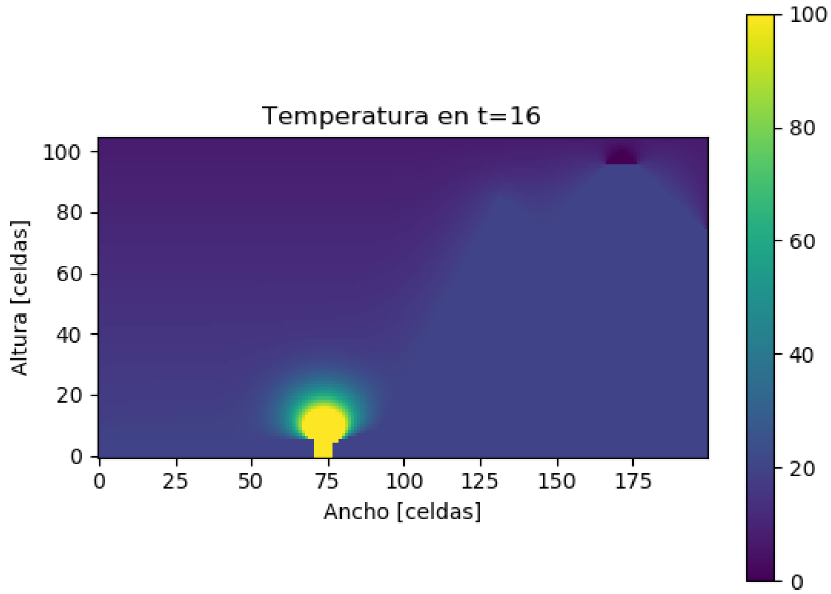


Figura 6: Cantidad de iteraciones: 393

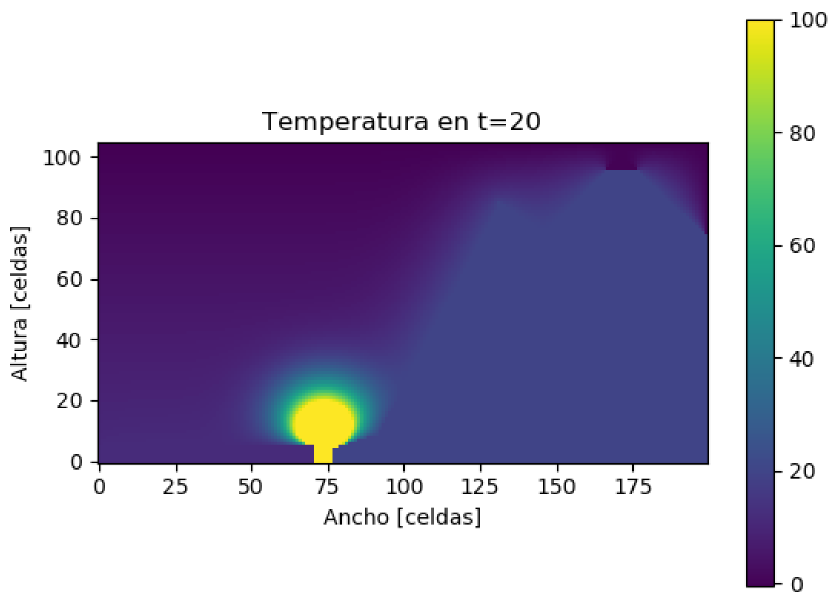


Figura 7: Cantidad de iteraciones: 463

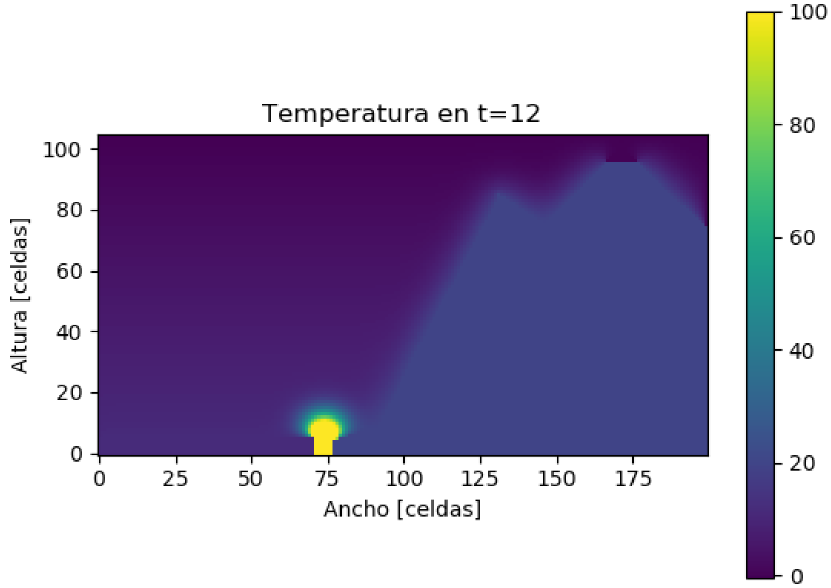


Figura 8: $\omega = 1{,}2$; Cantidad de iteraciones: 419

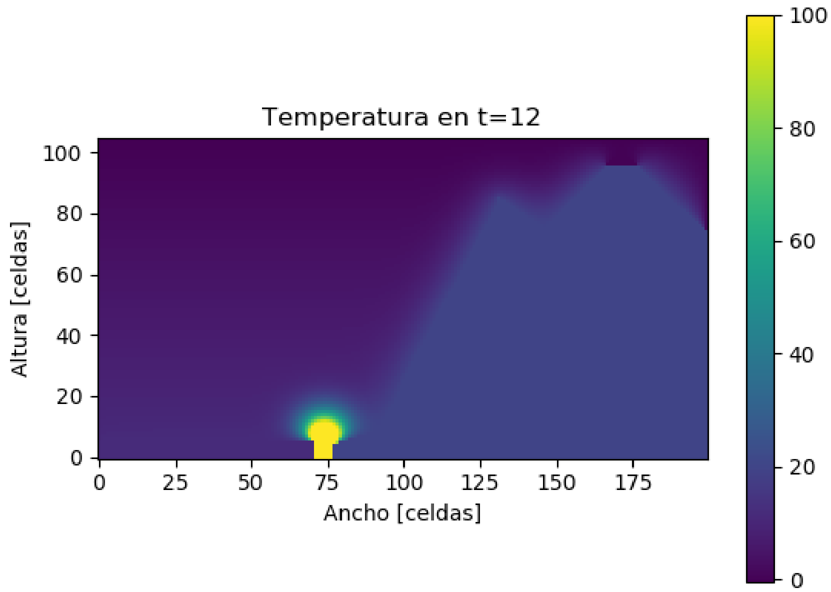


Figura 9: $\omega = 1{,}4$; Cantidad de iteraciones: 338

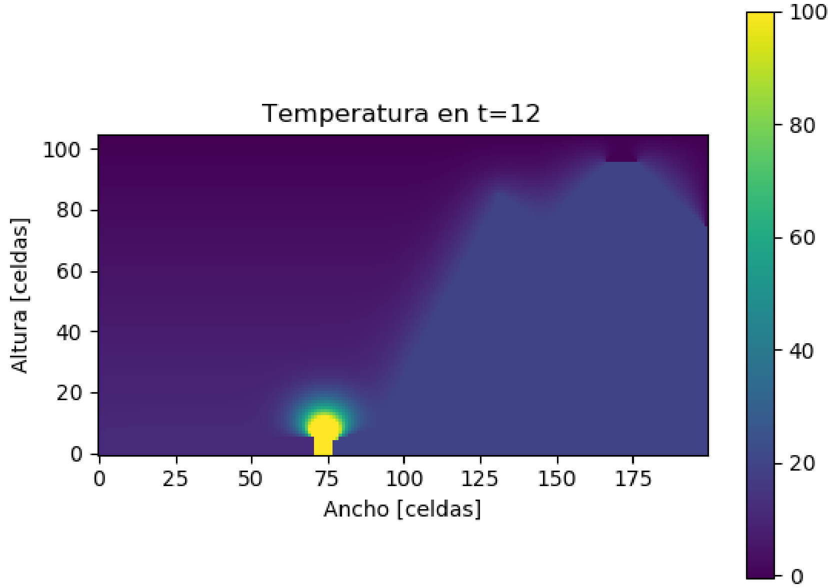


Figura 10: $\omega = 1{,}6$; Cantidad de iteraciones: 296

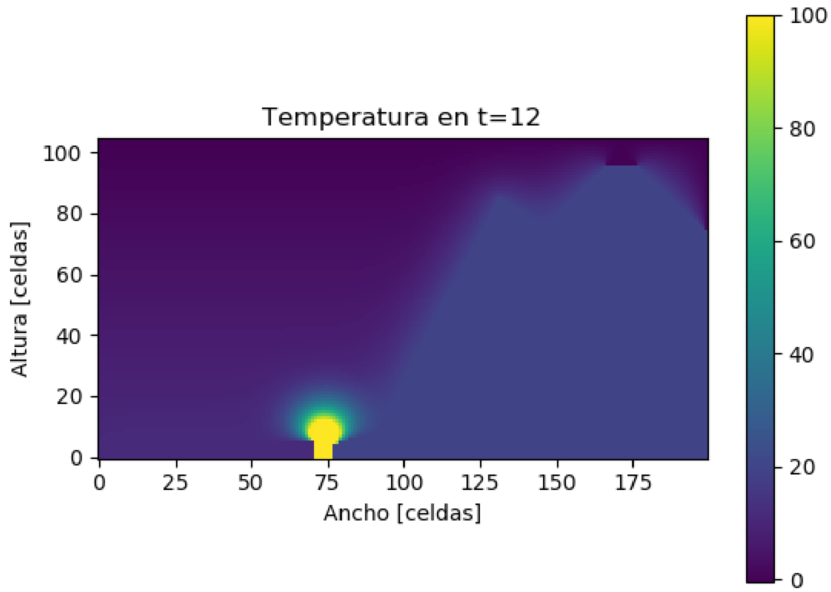


Figura 11: $\omega = 1{,}8$; Cantidad de iteraciones: 271

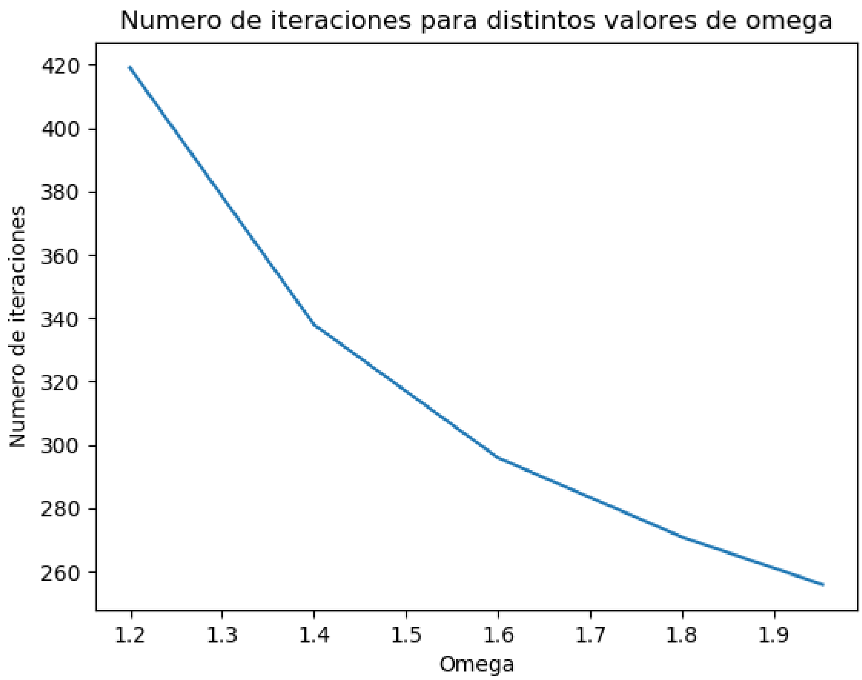


Figura 12: Tiempo = 12 horas

## Iteraciones con $\rho \neq 0$

A continuación se muestran los resultados obtenidos de iterar con la función $\rho$ dada en la tarea. Como se dijo anteriormente, todas estas iteraciones usan el $\omega$ óptimo calculado. Para cada gráfico, se indican las iteraciones obtenidas.

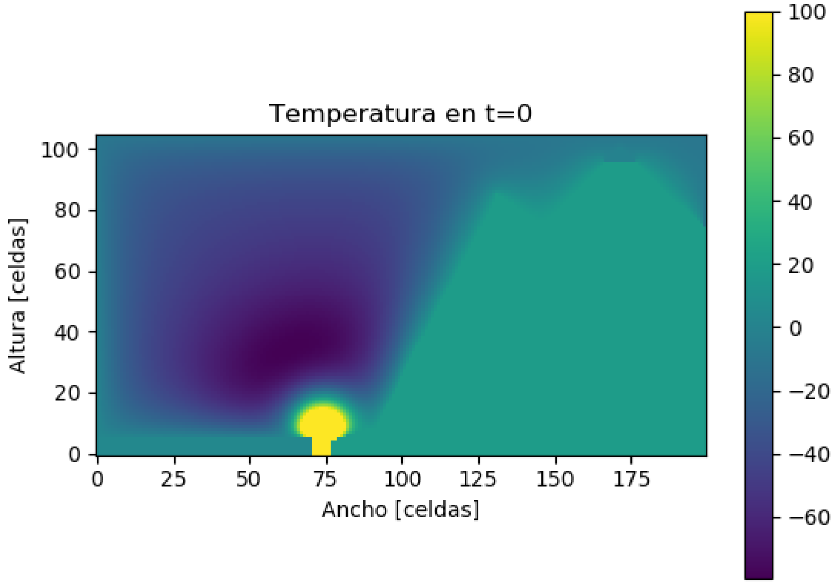


Figura 13: Cantidad de iteraciones: 635

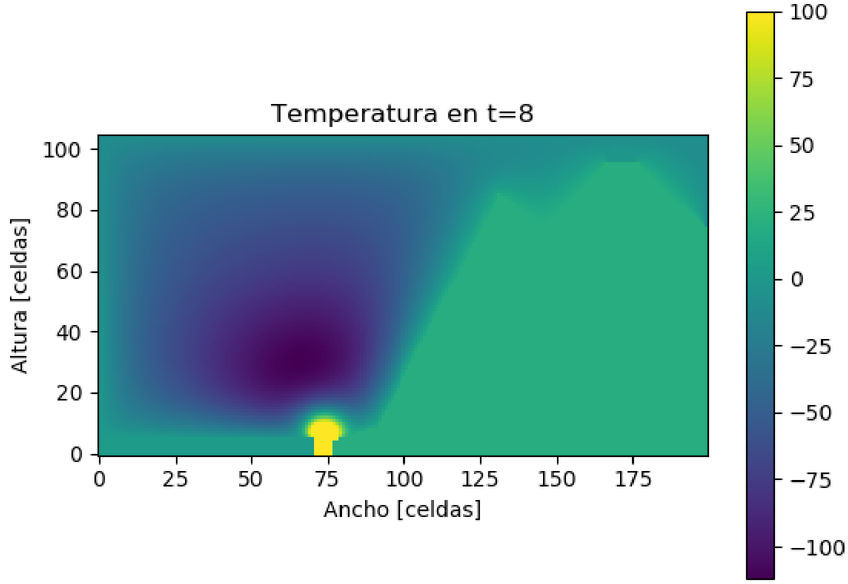


Figura 14: Cantidad de iteraciones: 845

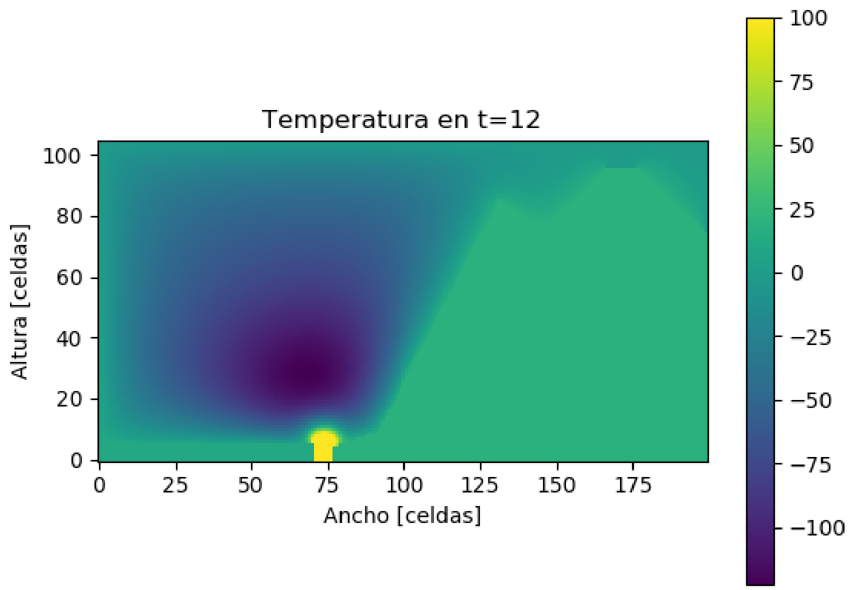


Figura 15: Cantidad de iteraciones: 944

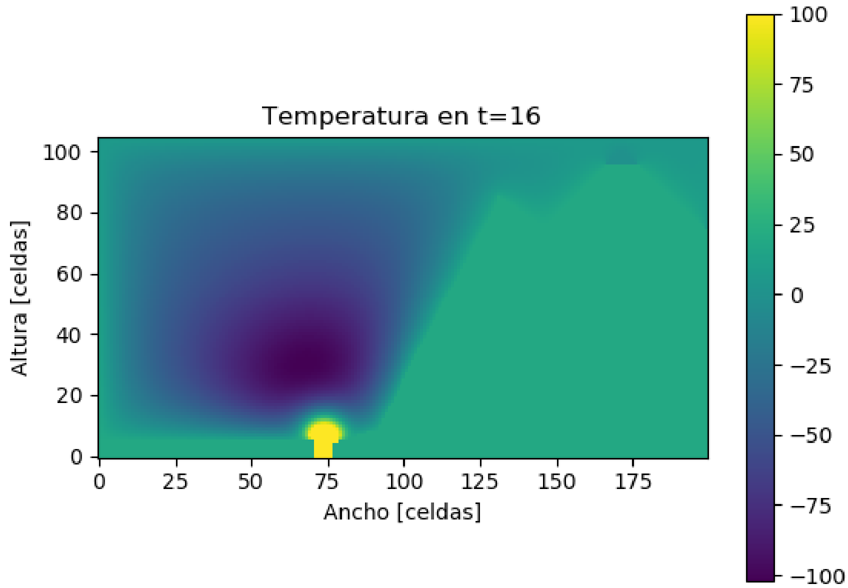


Figura 16: Cantidad de iteraciones: 886

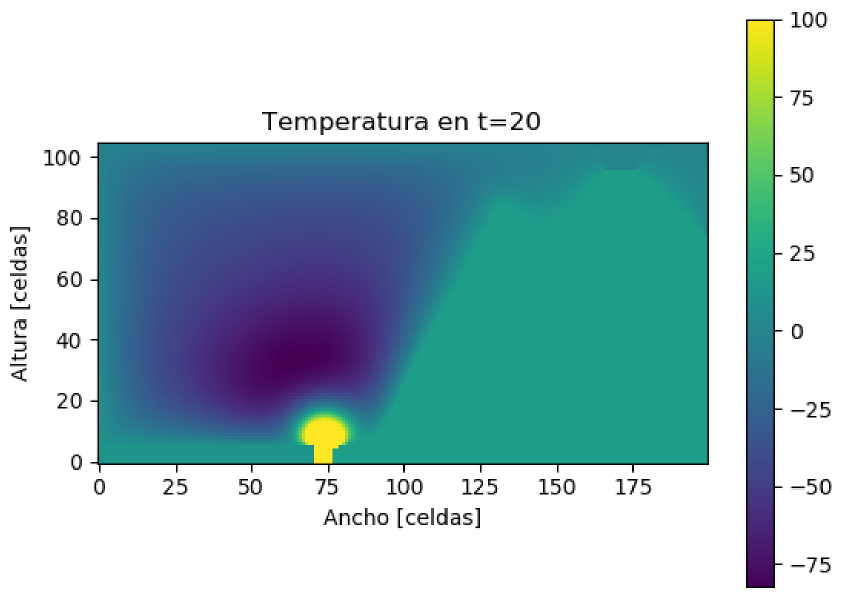


Figura 17: Cantidad de iteraciones: 711

# 5. Análisis de resultados

## ¿Ocurrió lo esperado?

Primero que todo es importante analizar si el método de relajación sucesiva funcionó. Anteriormente se mencionó que el principal objetivo de este es lograr que la convergencia de las iteraciones sea más rápida. Viendo el gráfico de la figura (12), se ve claramente como ocurrió lo deseado, puesto que para un mismo tiempo de 12, el número de iteraciones baja de 419 para un $\omega = 1{,}2$ (El más cercano a 1 de los probados) a 256 para $\omega = 1{,}953$ (El óptimo calculado).

Ahora, por otro lado, al ver los gráficos obtenidos con $\rho \neq 0$, surge la duda de por qué se generan temperaturas negativas en puntos tan cercanos a la planta, esto se ve en detalle más adelante en esta misma sección.

## Influencia del mar y las condiciones geográficas

Como es de esperarse, la temperatura del mar y de las condiciones geográficas no afecta mucho a la temperatura del sistema. Esto porque sus temperaturas que a lo más son de $\approx 20°C$ no tienen como competir con la temperatura de la planta que sobrepasa los $1200°C$.

Sin embargo, si funcionan como un "moldeador"para definir por donde se propagará la temperatura de la planta. Esto porque, al tener el mar y las montañas temperaturas constantes, la de la planta se propaga sólo por el aire. A pesar de esto, si el modelo se hubiera hecho sin mar, sin montañas, y simplemente fijando una temperatura fija a los bordes, probablemente se hubieran obtenido resultados bastante similares.

## Temperaturas medias

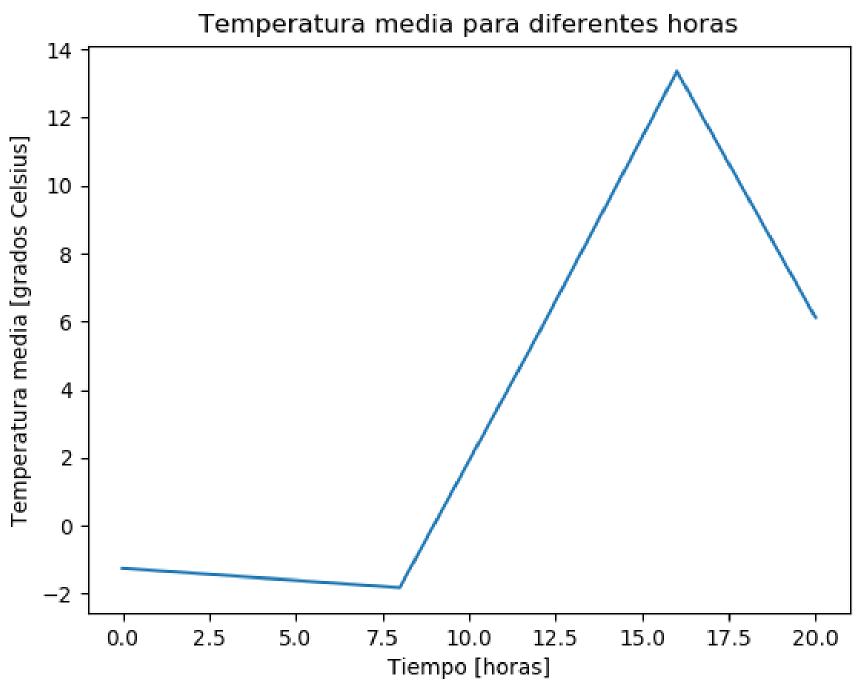


Figura 18: Temperaturas medias de toda la atmósfera para diferentes tiempos

En la figura (18) se ve como el comportamiento de las temperaturas medias del sistema es muy similar al de las iniciales. Esto puede reflejar que quizás la temperatura de la planta no se propagó grandes distancias, y que si bien la temperatura subió o bajó un poco (dependiendo de la hora), no hubieron grandes cambios a nivel general del sistema, esto demuestra que quizás el nivel de tolerancia era muy alto, por lo que las iteraciones parararon muy temprano. Sin embargo, disminuir dicho nivel hubiera provocado un aumento considerable en la demora del programa.

Por lo mencionado recién, es más valido analizar las temperaturas en un entorno mas cercano a la planta, para ver variaciones significativas.

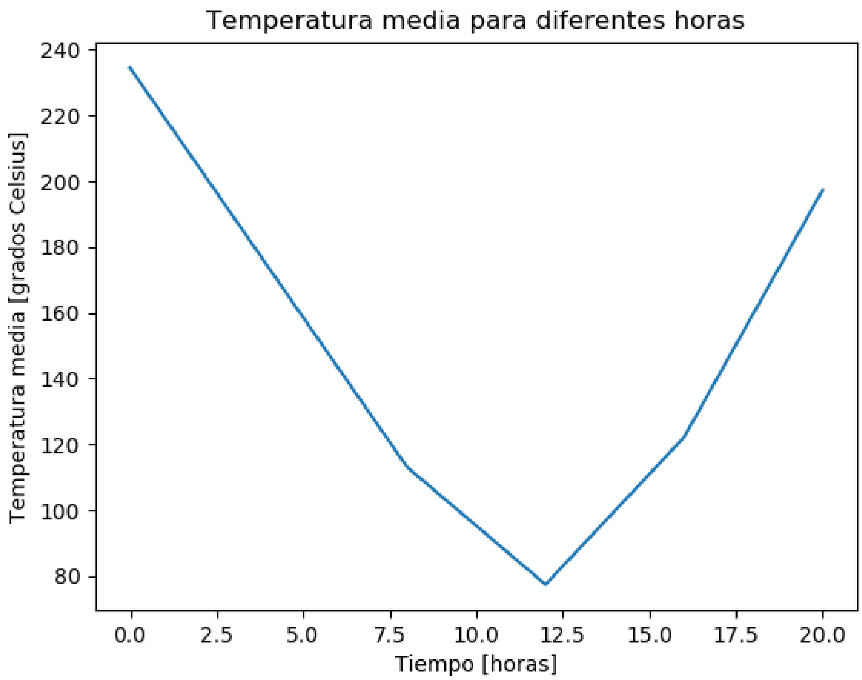


Figura 19: Temperaturas medias para diferentes horas, para puntos donde la distancia es menor a 300m

Notar que al ver el comportamiento de las temperaturas en los puntos cercanos a la planta, ocurre algo radicalmente opuesto a lo anteriormente visto. Esto porque ahora se excluyen los puntos donde la planta no influyó demasiado.

En los puntos lejanos a la planta, predomina la temperatura inicial, que se comporta de forma similar al gráfico de la figura (18). En el caso de la figura (19), el comportamiento se asimila al de la planta (como es de esperarse por cercanía), visto en la figura (20).

Con esto se ve claramente que la planta afecta bastante a su entorno local, pero que, al menos con una tolerancia de esta magnitud, los puntos mas lejanos no son influenciados de gran manera por esta.

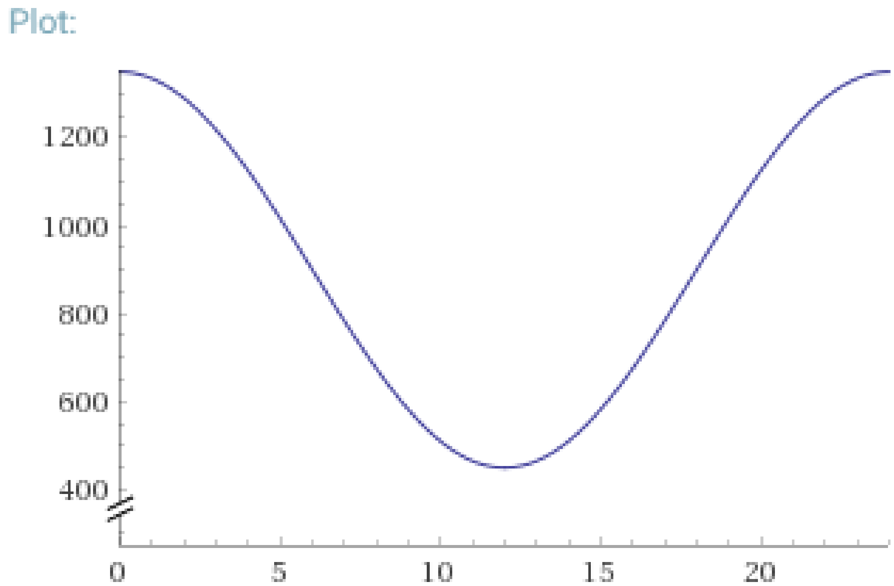


Figura 20: Comportamiento de la temperatura de la planta. Eje x: Tiempo en horas; Eje y: Temperatura. Graficado en wolfram alpha.

## Influencia de $\rho \neq 0$

Localmente, se puede ver en la figura (21) que se cumple lo que se espera: Un comportamiento similar al anterior, pero con temperaturas más bajas. No es difícil imaginar que esto ocurrirá luego de mirar la ecuación (3) ya que se le esta restando a las temperaturas vecinas un valor positivo.

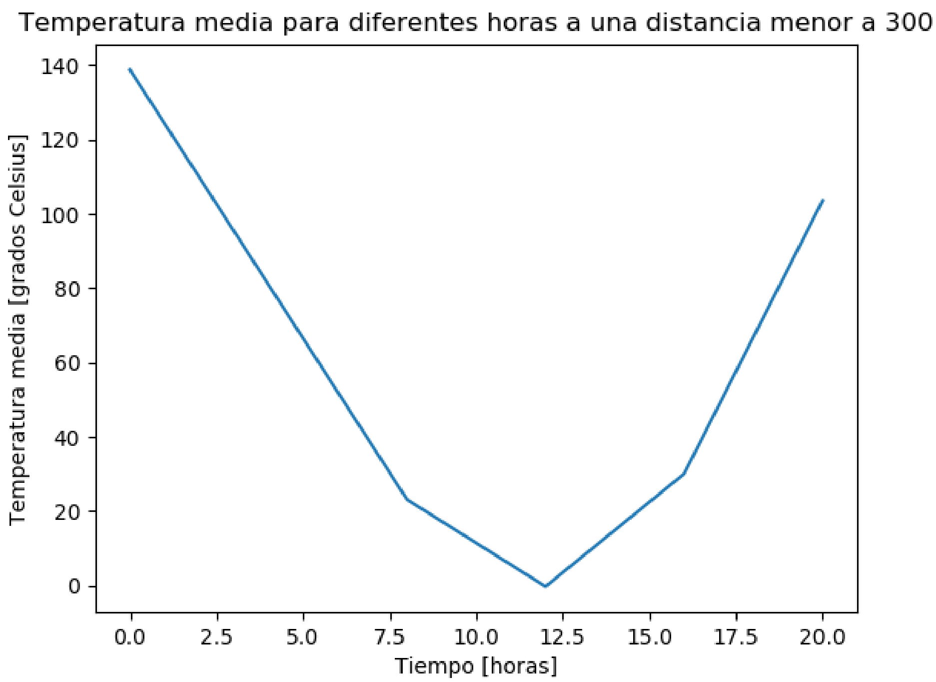


Figura 21: Temperaturas medias para puntos cercanos a la planta con $\rho \neq 0$

Lo curioso ocurre en distancias más largas, como se ve en la figura (22). Puesto que se generan temperaturas negativas. Esto, si bien es raro, tiene una explicación. Como se dijo anteriormente, debido a que el nivel de tolerancia era alto, la planta no influyó en puntos muy lejanos a ella. Por esto, al restarle $h^2 * \rho(x, y)$ a estas temperaturas que están alrededor de los $15°C$, se generan valores negativos, que luego se ven propagados por las iteraciones, generando lo visto en los gráficos anteriormente vistos.

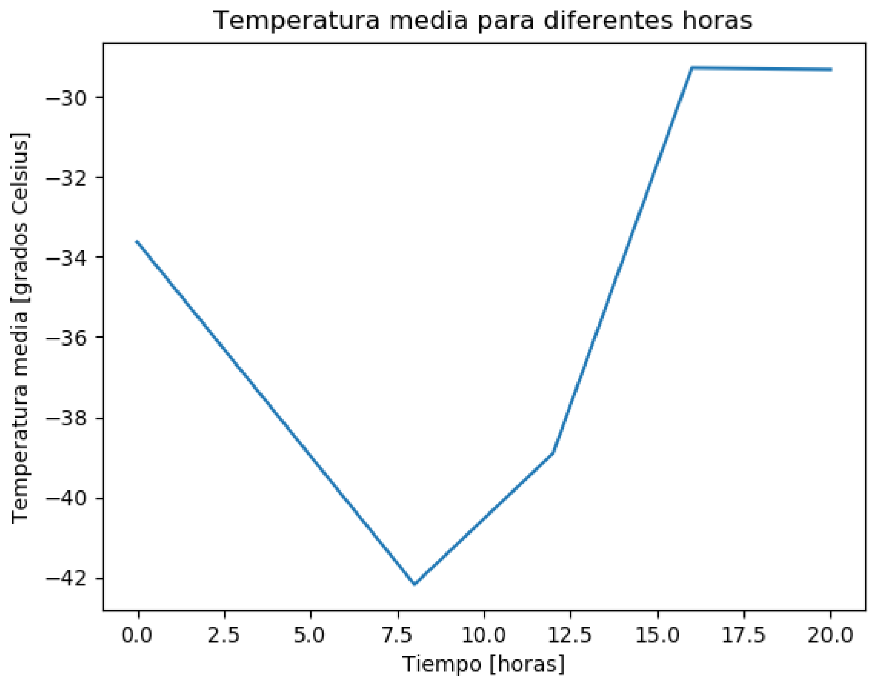


Figura 22: Temperaturas medias para todos los puntos de la atmósfera

En la figura (23) se ve el gráfico de la función $\rho$, como se ve mientras más es la distancia en x e y, menos valor tiene esta función. Por esto, en los gráficos de las figuras (13) a la (17) se pueden distinguir 3 "tipos"de puntos en la atmósfera: Aquellos cercanos a la planta en los que la temperatura subió (de color amarillo y verde), aquellos un poco más lejanos en los que la planta no afecto tanto pero la función $\rho$ si (de color azul oscuro) y aquellos ya más lejanos que no se vieron muy afectados (de color azul más claro).

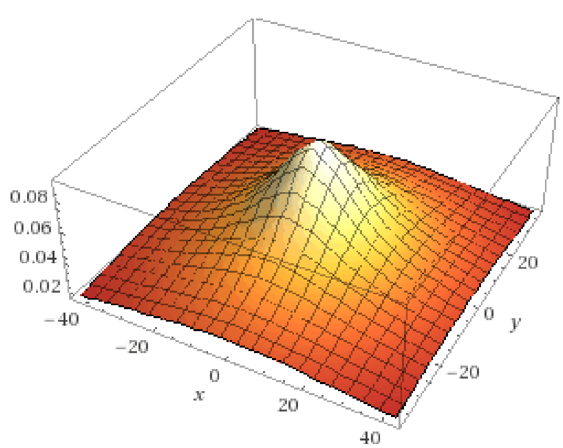


Figura 23: Función $\rho$ utilizada en la tarea. Graficada en wolfram alpha.

# 6. Conclusiones

## Lo bueno

Como aspectos positivos del trabajo realizado, se tiene que los resultados obtenidos están acordes a lo esperado, como se mencionó anteriormente. A pesar de que en un principio puedan parecer desconcertantes, luego de hacer el análisis explicado, se logra entender porque dan gráficos como los vistos.

## Lo malo

Hay dos grandes aspectos que se deberían mejorar para futuras tareas:

- El orden del código, especialmente de la clase Corte, podría haber sido mucho mejor. Tiene métodos que son demasiado largos, engorrosos y que podrían haber sido divididos en métodos mas cortos y simples. Esto hace que el código sea poco extendible y difícil de modificar. Si se buscara hacer una versión mejorada, definitivamente se le podría hacer refactoring a varios aspectos.
- Hay que analizar los resultados obtenidos antes de desecharlos por "no calzar con lo esperado a simple vista". La tarea estuvo mucho tiempo estancada, cuando en realidad los resultados que estaba dando si eran los correspondientes, a pesar de que no fueran similares a lo que uno imaginaría. Por esto se perdió mucho tiempo que podría haber sido utilizado en solucionar el primer punto expuesto.