Problema 1: (1 punto)
Muestra teóricamente que la fución $f(x) = e^x - \frac{4}{2+x}$ tiene una raíz en el intervalo $[0, 1]$. Aplica alguno de los algoritmos vistos en clase para encontrar la raíz de $f(x) = 0$. Explica los pasos que realizaste para alcanzar la raíz.

Solución. Para demostrar que una función continua $f : \mathbb{R} \to \mathbb{R}$ tiene una raíz en un intervalo $[a, b] \subset \mathbb{R}$, podemos utilizar el **teorema de Bolzano**, que nos asegura la existencia de **al menos** una raíz $c \in (a, b)$ si se cumple alguna de las siguientes dos condiciones:

$$f(a) < 0 < f(b) \qquad \text{ó} \qquad f(a) > 0 > f(b).$$

Es decir, que nuestra función tenga signo distinto en los puntos límite del intervalo. Entonces, para aplicarlo a este ejercicio, primero notemos que

$$\begin{aligned} f(0) &= e^0 - \frac{4}{2+0} \\ &= 1 - 2 \\ &= -1. \end{aligned}$$

Mientras que

$$\begin{aligned} f(1) &= e^1 - \frac{4}{2+1} \\ &= e - \frac{4}{3} \\ &> 0. \end{aligned} \tag{1}$$

Nótese que (1) se sigue de que $e \approx 2.7 > 1.33....$ Entonces, como

$$f(0) < 0 < f(1),$$

podemos concluir, por el teorema de Bolzano, que f tiene al menos una raíz en el intervalo $(0, 1)$. Por completitud del ejercicio, también mencionamos que claramente f no tiene raíces en los puntos límite.
Ahora que hemos confirmado la existencia de al menos una raíz, podemos aplicar un método iterativo para encontrar una raíz. Elegimos el **método de bisección**. Decidimos usar este método ya que es el más simple y no requiere de pasos adicionales (como calcular derivadas o encontrar pseudo-inversas). El código se encuentra en el archivo `p1.c`. Para compilar y ejecutar, usamos un `makefile`, que corremos usando el siguiente comando:

```
make run-p1 ARGS="1e-15 10000".
```

El primer argumento corresponde a la *tolerancia*, y el segundo argumento corresponde al máximo número de iteraciones. Al correr el programa, se imprime a la consola el siguiente mensaje:

```
Bisection method converged at iteration 49
        root = 0.478600339499129
f(0.478600339499129) = -0.000000000000001
```

Esto nos confirma que una raíz que se encuentra en el intervalo $(0, 1)$ es

$$c = 0.478600339499129.$$

Para llegar a este resultado, el algoritmo de bisección realizó 49 iteraciones. En cada iteración, busca el punto medio del intervalo dado, y evalúa la función en ese punto. Si es menor a la tolerancia, acabamos. Si no, reducimos el intervalo de búsqueda dependiendo del signo de la función en el punto medio y puntos extremos, buscando que el signo sea distinto. ◇

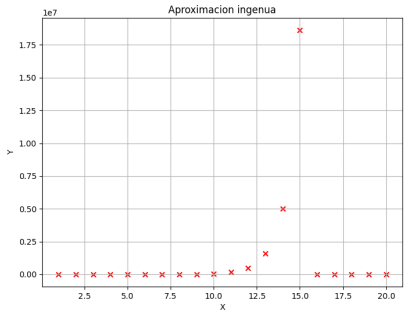
Problema 2: (2 puntos)
Escribe un programa que calcule $x(\sqrt{x+1} - \sqrt{x})$ para $x = 10^n$, con $n = 1, 2, \ldots, 20$.

- Grafica los resultados. Proporciona una explicación de los sucedido. **(1 punto)**
- Proporciona una expresión alternativa y compara los resultados obtenidos. Explica. **(1 punto)**

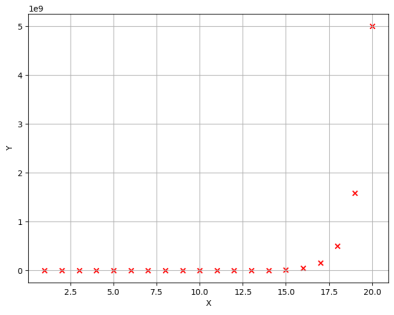
Solución. Para correr el programa, usar

```
make run-p2 ARGS="p2.txt p2_new.txt"
```

Esto nos genera 2 archivos, uno con la secuencia de la función sin modificar (p2.txt) y el otro contiene la secuencia con la función modificada.



(a) $x(\sqrt{x+1} - \sqrt{x})$



(b) $\frac{x}{\sqrt{x+1}+\sqrt{x}}$

Figure 1: Comparación de gráficas de la misma función, con diferente representación.

Notemos que al graficar los puntos de la función $x(\sqrt{x+1} - \sqrt{x})$ (gráfica 1a), al principio de los valores de n, no tenemos problemas. Sin embargo, cuando alcanzamos el valor $n = 16$, la

expresión resulta 0. Esto es debido a la precición finita de punto flotante. Como $\sqrt{x+1}$ y $\sqrt{x}$ son my similares para números grandes, su diferencia es muy pequeña en esos rangos, por lo que $\sqrt{x+1} - \sqrt{x} = 0$ para $x \geq 10^{16}$ de acuerdo a la precisión de la máquina.
Para contrarestar esta limitante, evitamos el uso de la resta mediante manipulación algebráica usando $a^2 - b^2 = (a+b)(a-b)$:

$$\begin{aligned} x(\sqrt{x+1} - \sqrt{x}) &= x(\sqrt{x+1} - \sqrt{x}) \cdot \frac{\sqrt{x+1} + \sqrt{x}}{\sqrt{x+1} + \sqrt{x}} \\ &= \frac{x(x+1-x)}{\sqrt{x+1} + \sqrt{x}} \\ &= \frac{x}{\sqrt{x+1} + \sqrt{x}}. \end{aligned}$$

Entonces, realizando nuevamente los cálculos, obtenemos la gráfica 1b, que muestra el comportamiento real esperado. Ahora, sin el término que se vuelve 0 por el límite de precisión, la expresión se evaulúa correctamente. ◇

Problema 3: (2 puntos)
La ecuación $f(x) = x - 3x^{-x} = 0$ tiene una solución $x^* \approx 1.05$.

- Analiza las siguientes iteraciones de punto fijo desde un punto de vista teórico y determina si convergen hacia x^*, además, identifique cuál de los métodos presenta una convergencia más rápida. **(1.5 puntos)**

  (i) $x_{n+1} = 3e^{-x_n}$, (ii) $x_{n+1} = (2x_n + 3e^{-x_n})/3$, (iii) $x_{n+1} = 1.05x_n + 3e^{-x_n}$, (iv) $x_{n+1} = (x_n + 3e^{-x_n})/2$

- Presenta una tabla de datos comparativos de tal manera que se alcance un error absoluto de al menos 10^{-10} considerando $x_0 = 1.05$. (**0.5 puntos**)

Solución. Este programa se ejecuta usando

`make run-p3`.

Nos genera 4 archivos:

`p3_1.txt` — Tabla de iteraciones de g_1.
`p3_2.txt` — Tabla de iteraciones de g_2.
`p3_3.txt` — Tabla de iteraciones de g_3.
`p3_4.txt` — Tabla de iteraciones de g_4.

1. $x_{n+1} = 3e^{-x_n}$, Esta función no satisface la condición de $|g'(x)| \leq k < 1$. Efectivamente,

$$|g_1'(x)| = 3e^{-x} > 1 \quad \forall x \leq 1.05$$

Entonces, el teorema de punto fijo no nos garantiza la convergencia, ya que en ninguna vecindad del punto $x = 1.05$ se logra cumplir esta propiedad.

2. $x_{n+1} = (2x_n + 3e^{-x_n})/3$, Para esta función, tenemos

$$|g_2'(x)| = \frac{|2 - 3e^x|}{3} < 1 \qquad \forall x \in (1, 1.1)$$

y también $g_2(x) \in (1, 1.1)$ para $x \in (1, 1.1)$. Es decir, podemos aplicar el teorema de punto fijo con esperanza de obtener la raíz de f.

3. $x_{n+1} = 1.05x_n + 3e^{-x_n}$, Esta función no tiene sentido ni considerarla, ya que

$$g_3(x) > x \qquad \forall x$$

Es decir, no es posible que exista un punto fijo.

4. $x_{n+1} = (x_n + 3e^{-x_n})/2$ Para esta función verificamos que las condiciones se cumplen:

$$g_4(x) \in (1, 1.1) \qquad \forall x \in (1, 1.1)$$

y

$$|g_4(x)| = \frac{|1 - 3e^{-x}|}{2} < 1 \qquad \forall x \in (1, 1.1)$$

Entonces, podemos concluir que, teóricamente, solo las opciones 3 y 4 convergen. Ahora bien, si queremos conocer cuál converge más rápido, basta ver que

$$|g_4'(x)| = \frac{|1 - 3e^{-x}|}{2} \leq \frac{|2 - 3e^x|}{3} = |g_2'(x)|.$$

para $x \in (1, 1.1)$. Es decir, la constante de convergencia de g_4 es menor que la de g_2.

n	$\lvert x_n - g_1(x_n)\rvert$	$\lvert x_n - g_2(x_n)\rvert$	$\lvert x_n - g_3(x_n)\rvert$	$\lvert x_n - g_4(x_n)\rvert$
0	1.868×10^{-4}	6.225×10^{-5}	1.102×10^{0}	9.33763×10^{-15}
1	1.868×10^{-4}	6.225×10^{-5}	1.102×10^{0}	9.33763×10^{-05}
2	1.961×10^{-4}	1.972×10^{-5}	4.563×10^{-1}	2.32798×10^{-06}
3	2.059×10^{-4}	6.244×10^{-6}	3.513×10^{-1}	5.80947×10^{-08}
4	2.161×10^{-4}	1.978×10^{-6}	3.035×10^{-1}	1.44972×10^{-09}
5	2.269×10^{-4}	6.263×10^{-7}	2.779×10^{-1}	3.61771×10^{-11}
6	2.382×10^{-4}	1.983×10^{-7}	2.640×10^{-1}	9.02611×10^{-13}
7	2.501×10^{-4}	6.281×10^{-8}	2.570×10^{-1}	2.24265×10^{-14}
8	2.626×10^{-4}	1.989×10^{-8}	2.547×10^{-1}	6.66134×10^{-16}
9	3.757×10^{-4}	6.300×10^{-9}	2.559×10^{-1}	—
10	2.895×10^{-4}	1.995×10^{-9}	2.596×10^{-1}	—
11	3.039×10^{-4}	6.318×10^{-10}	2.655×10^{-1}	—

Table 1

◇

Problema 4: (5 puntos) Se requiere resolver un problema de difusión del cual se obtiene un sistema de ecuaciones pentadiagonal. El sistema de ecuaciones tiene la siguiente forma

$$\mathbf{Ax} = \mathbf{b},$$

con

$$-5x_{i-2} - 10x_{i-1} + 50x_i - 10x_{i+1} - 5x_{i+1} = 200, \qquad i = 3, \ldots, 1998.$$

En los extremos de la matriz $\mathbf{A}$ se eliminan los términos que quedan fuera (cuando el índice es negativo o mayor a 2000). Así,

- La primera ecuación es: $50x_1 - 10x_2 - 5x_3 = 40$;
- La segunda ecuación es: $-10x_1 + 50x_2 - 10x_3 - 5x_4 = 100$;
- La penúltima ecuación es: $-5x_{1997} - 10x_{1998} + 50x_{1999} - 10x_{2000} = 100$;
- La última ecuación es: $-5x_1 - 10x_{1999} - 50x_{2000} = 40$.

Con el sistema generado (**1 punto**), obtener:

- La solución del sistema de ecauciones con dos métodos distintos. Decir por qué utilizaste esos métodos de solución. (**2 puntos**)
- Los eigenvectores correspondientes a la matriz $\mathbf{A}$. Presentar los 10 eigenvalores más pequeños y los 10 más grandes. (**2 puntos**)

Solución.
Compilamos y ejecutamos con

`make run-p4`

Guardamos los el vector de constantes como `vector.txt`, y la matriz de 2000×2000 como `matrix.txt`.
Para resolver el sistema, usamos el método de gradiente conjugado y el método de Gauss-Siedel. Elegimos estos ya que la matriz es diagonalmente dominante, simetrica y positiva definida. Específicamente, estos métodos (en teoría) trabajan mejor con matrices con muchos ceros, a comparación de otras como LU o por medio de factorización de Cholesky.
Con la ejecución mencionada anteriormente, obtenemos 2 archivos:

`conjugate_s.txt` y `gauss_s.txt`.

El método de gradiente conjugado converge[a] en 28 iteraciones con una tolerancia de 10^{-12}, mientras que el de Gauss-Siedel en 40, con la misma tolerancia.
Para encontrar los eigenvectores y eigenvalores, usamos el método de la potencia y potencia inversa. En contraste, el método de iteración de subespacio parece ser más ineficiente debido al proceso de Jacobi intermedio que se utiliza para rotar la matriz (en este caso, gigantesca).
Al ejecutar el programa, obtenemos 4 archivos.

`biggest.txt` — Eigenvalores más grandes.
`smallest.txt` — Eigenvalores más pequeños.
`biggest_evec.txt` — Eigenvectores correspondientes a los eigenvalores más grandes.
`smallest_evec.txt` — Eigenvectores correspondientes a los eigenvalores más pequeños.

n	Grandes	Pequeños
1	64.977933	20.001248
2	64.977284	20.012309
3	64.977484	20.012727
4	64.977752	20.012647
5	64.977983	20.012541
6	64.978182	20.012506
7	64.978342	20.012607
8	64.978473	20.013074
9	64.978581	20.014888
10	64.978674	20.019698

Table 2: Eigenvalores más grandes y pequeños

Notemos que no son estrictamente decrecientes/crecientes. Esto se debe a la imprecisión de los puntos flotantes, y como una solución depende la otra (al hacer la reducción), estos errores

van acarreando. Estos resultados son con tolerancia de 10^{-5}. ◇

[a] como el vector inicial se genera aleatoriamente, puede que la convergencia sea distinta