Roll No: EE18B067 Name:Abhishek Sekar

Date: December 12, 2021

1. [Fredholm Integral]
   A common problem in mathematical physics is that of solving the Fredholm integral equation

$$f(x) = \phi(x) + \int_a^b K(x,t)\phi(t)dt$$

where the function f(x) and K(x,t) are given and the above problem is to obtain $\phi(x)$.

- Describe a numerical method for solving the above equation.
- Solve the following equation

$$\phi(x) = \pi x^2 + \int_0^\pi 3\left(0.5\sin 3x - tx^2\right)\phi(t)dt$$

Obtain the exact solution of the above and compare your numerical solution with it.

Solution:
Numerical Approach
The above Fredholm integral equation is reduced to a single numerical integration problem if we were permitted to use 'x' as a symbol.
However, this is just cheating and not a wholly numerical approach in spirit!
Thus, we discretize along both x and t.
For each node of x, we solve a numerical integration problem.
We collect all these values obtained across all the nodes of x.Now, this just reduces to a function interpolation problem.
Writing the above approach mathematically,
let $\{t_i\}$ correspond to the nodes we use to discretize t and $\{x_j\}$ be the nodes for x. Then we have for each node x_j,

$$f(x_j) = \phi(x_j) + \int_a^b K(x_j,t)\phi(t)dt$$

Without loss of generality, let us use a Gaussian Quadrature approximation for the integral using N+1 legendre nodes and let N+1 nodes be used for the interpolation too.
The approximation is as follows:

$$\int_{-1}^1 f(x)dx = \sum_{i=0}^N w_i f(l_i)$$

Where $\{l_i\}$ are the legendre nodes and $\{w_i\}$ the corresponding weights.
This approximation can be easily expanded to $\int_a^b f(t)dt$ case by making the transformation, $f(x) = f\left(\frac{a+b}{2} + \frac{b-a}{2}x\right)$. Now this f(x) can be used in 1 with a scaling of $\frac{b-a}{2}$.
From the above, transformation, we have,

$$\int_b^a f(x)dx = \frac{b-a}{2}\sum_{i=0}^N w_i f\left(\frac{a+b}{2} + \frac{b-a}{2}l_i\right) \tag{1}$$

$$f(x_j) = \phi(x_j) + \frac{b-a}{2}\sum_{i=0}^N w_i K\left(x_j, \frac{a+b}{2} + \frac{b-a}{2}l_i\right)\phi\left(\frac{a+b}{2} + \frac{b-a}{2}l_i\right) \tag{2}$$

Repeating 2 for all x_j, leaves us with the below linear system.

$$\begin{bmatrix} f(x_0) \\ f(x_1) \\ \vdots \\ f(x_N) \end{bmatrix}_{N\text{x}1} = \begin{bmatrix} \phi(x_0) \\ \phi(x_1) \\ \vdots \\ \phi(x_N) \end{bmatrix}_{N\text{x}1} + \left(\frac{b-a}{2}\right) \cdot \begin{bmatrix} \sum\limits_{i=0}^{N} w_i K\left(x_0, \frac{a+b}{2} + \frac{b-a}{2} l_i\right) \phi\left(\frac{a+b}{2} + \frac{b-a}{2} l_i\right) \\ \sum\limits_{i=0}^{N} w_i K\left(x_1, \frac{a+b}{2} + \frac{b-a}{2} l_i\right) \phi\left(\frac{a+b}{2} + \frac{b-a}{2} l_i\right) \\ \vdots \\ \sum\limits_{i=0}^{N} w_i K\left(x_N, \frac{a+b}{2} + \frac{b-a}{2} l_i\right) \phi\left(\frac{a+b}{2} + \frac{b-a}{2} l_i\right) \end{bmatrix}_{N\text{x}1}$$

$$\Rightarrow \begin{bmatrix} \phi(x_0) \\ \phi(x_1) \\ \vdots \\ \phi(x_N) \end{bmatrix}_{N\text{x}1} +$$

$$\left(\frac{b-a}{2}\right) \cdot \begin{bmatrix} w_0 K\left(x_0, \frac{a+b}{2} + \frac{b-a}{2} l_0\right) \phi\left(\frac{a+b}{2} + \frac{b-a}{2} l_0\right) & \dots & \dots & w_N K\left(x_0, \frac{a+b}{2} + \frac{b-a}{2} l_N\right) \phi\left(\frac{a+b}{2} + \frac{b-a}{2} l_N\right) \\ w_0 K\left(x_1, \frac{a+b}{2} + \frac{b-a}{2} l_0\right) \phi\left(\frac{a+b}{2} + \frac{b-a}{2} l_0\right) & \dots & \dots & w_N K\left(x_1, \frac{a+b}{2} + \frac{b-a}{2} l_N\right) \phi\left(\frac{a+b}{2} + \frac{b-a}{2} l_N\right) \\ \vdots & \vdots & \ddots & \vdots \\ w_0 K\left(x_N, \frac{a+b}{2} + \frac{b-a}{2} l_0\right) \phi\left(\frac{a+b}{2} + \frac{b-a}{2} l_0\right) & \dots & \dots & w_N K\left(x_N, \frac{a+b}{2} + \frac{b-a}{2} l_N\right) \phi\left(\frac{a+b}{2} + \frac{b-a}{2} l_N\right) \end{bmatrix}_{N\text{x}N} \cdot 1_{N\text{x}}$$

Where, 1 is the vector with all ones.
From the above equation, we have N simultaneous equations in terms of $\{\phi(x_i)\}$ and $\left\{\phi\left(\frac{a+b}{2} + \frac{b-a}{2} l_i\right)\right\}$.
Now, the above linear system can only be solved if $x_i = \frac{a+b}{2} + \frac{b-a}{2} l_i$ otherwise we'll have an under-determined system.
With this change made, we can solve the linear system to find the value of $\phi(x)$ at the nodes $\{x_i\}$ which can be used to approximate the function as

$$p(x) = \sum_{i=1}^{n} \phi(x_i) L_i(x) \qquad \text{(The approximation)} \qquad (3)$$

$$L_i(x) = \prod_{\substack{j=1 \\ j \neq i}}^{n} \frac{(x - x_j)}{(x_i - x_j)} \qquad \text{(The } i^{th} \text{ lagrange polynomial)} \qquad (4)$$

Exact solution to the given equation
The given equation is

$$\phi(x) = \pi x^2 + \int_0^\pi 3\left(0.5 \sin 3x - t x^2\right) \phi(t) dt$$

The exact solution to the above integral is $\phi(x) = \sin(3x)$.
We'd expect $\phi(x)$ to be of the form $A \sin 3x + Bx^2$ as the integral is only over t.
On actually checking we arrive at the above answer.
Let us verify that $\phi(x) = \sin(3x)$ is indeed the answer.

$$\begin{aligned}
\text{R.H.S} &= \pi x^2 + \int_0^{\pi} 3\left(0.5\sin 3x - tx^2\right)\phi(t)dt \\
&\Rightarrow \pi x^2 + \int_0^{\pi} 3\left(0.5\sin 3x - tx^2\right)\sin 3tdt \\
&\Rightarrow \pi x^2 + \frac{3\sin 3x}{2}\cdot\int_0^{\pi}\sin 3tdt - 3x^2\cdot\int_0^{\pi} t\sin 3tdt \\
&\Rightarrow \pi x^2 + \frac{3\sin 3x}{2}\cdot\left[\frac{-\cos 3t}{3}\right]_0^{\pi} - 3x^2\cdot\left[\left[\frac{-t\cos 3t}{3}\right]_0^{\pi} + \int_0^{\pi}\frac{\cos 3t}{3}dt\right] && \text{(Integrating by parts)} \\
&\Rightarrow \pi x^2 + \frac{3\sin 3x}{2}\cdot\left(\frac{2}{3}\right) - 3x^2\cdot\left(\frac{\pi}{3}\right) && \left(\text{As } \int_0^{\pi}\tfrac{\cos 3t}{3}dt = 0\right) \\
&\Rightarrow \sin 3x
\end{aligned}$$

Hence, L.H.S = R.H.S and thus our solution is correct.
Numerical Method to solve the above equation
Starting from where we left in 1,2 and 3,for this problem we have,

$$\begin{bmatrix}\phi(x_0)\\ \phi(x_1)\\ \vdots\\ \phi(x_N)\end{bmatrix}_{N\text{x}1} = \begin{bmatrix}\pi x_0^2\\ \pi x_1^2\\ \vdots\\ \pi x_N^2\end{bmatrix}_{N\text{x}1} + \left(\frac{\pi}{2}\right)\cdot\begin{bmatrix}\sum_{i=0}^{N} w_i K\left(x_0, x_i\right)\phi\left(x_i\right)\\ \sum_{i=0}^{N} w_i K\left(x_1, x_i\right)\phi\left(x_i\right)\\ \vdots\\ \sum_{i=0}^{N} w_i K\left(x_N, x_i\right)\phi\left(x_i\right)\end{bmatrix}_{N\text{x}1}$$

Where, $K(x,t) = 3\left(0.5\sin 3x - tx^2\right)$.
This can be converted to a linear system of the form Ax = b, where

$$A = \begin{bmatrix}1-\frac{\pi}{2}w_0K(x_0,x_0) & -\frac{\pi}{2}w_0K(x_0,x_1) & \dots & -\frac{\pi}{2}w_0K(x_0,x_N)\\ -\frac{\pi}{2}w_1K(x_1,x_0) & 1-\frac{\pi}{2}w_1K(x_1,x_1) & \dots & -\frac{\pi}{2}w_1K(x_1,x_N)\\ \vdots & \vdots & \ddots & \vdots\\ -\frac{\pi}{2}w_NK(x_N,x_0) & -\frac{\pi}{2}w_NK(x_N,x_1) & \dots & 1-\frac{\pi}{2}w_NK(x_N,x_N)\end{bmatrix}_{N\text{x}N}$$

$$x = \begin{bmatrix}\phi(x_0)\\ \phi(x_1)\\ \vdots\\ \phi(x_N)\end{bmatrix}_{N\text{x}1}$$

$$b = \begin{bmatrix}\pi x_0^2\\ \pi x_1^2\\ \vdots\\ \pi x_N^2\end{bmatrix}_{N\text{x}1}$$

Plots:
Doing this in the code by using different number of nodes, we obtain the following plots.

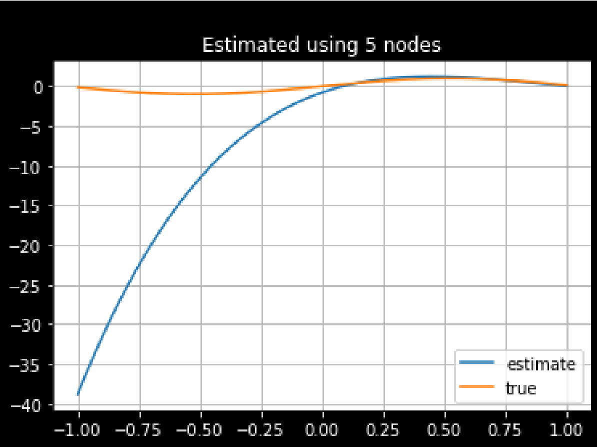


Figure 1: Approximation using 5 nodes

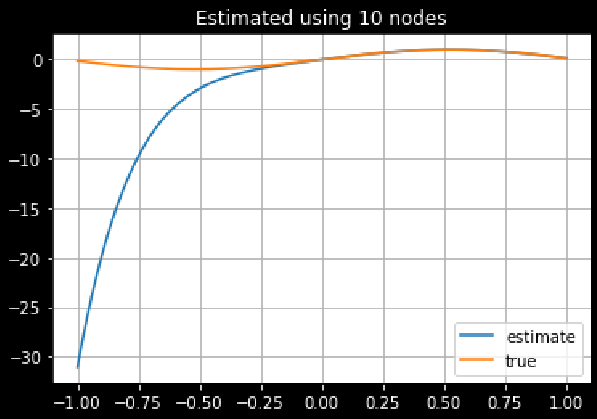


Figure 2: Approximation using 10 nodes

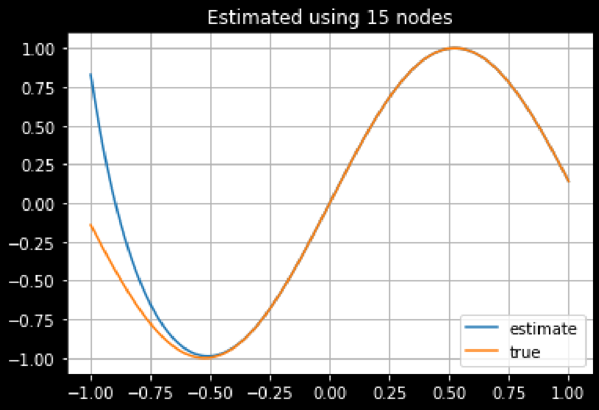


Figure 3: Approximation using 15 nodes

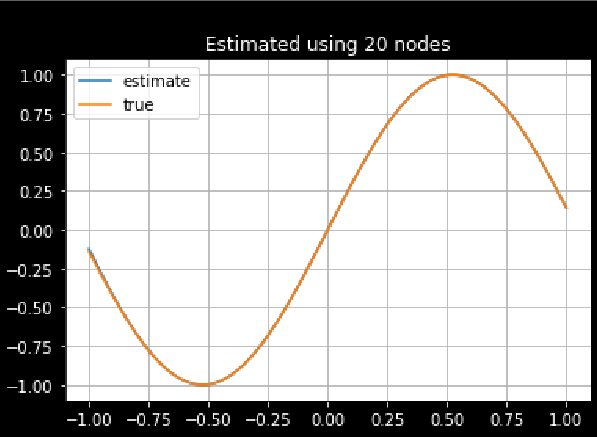


Figure 4: Approximation using 20 nodes

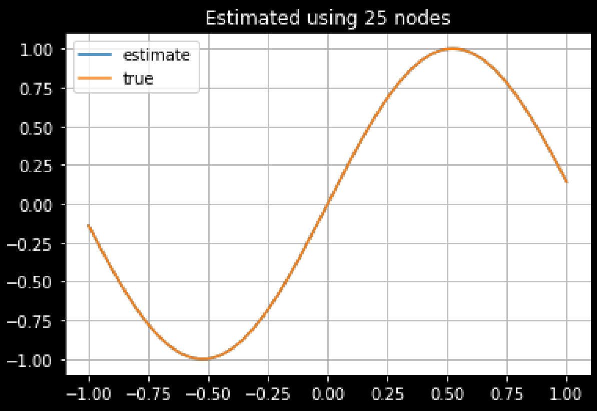


Figure 5: Approximation using 25 nodes

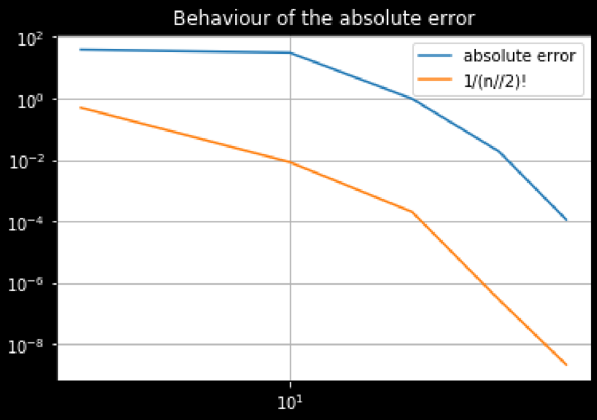


Figure 6: Plot of the error

Observations

- We see that our approximation scheme works splendid for as little as 20 nodes. The approximated output cannot be differentiated from the true value.

- As we use a Gaussian Quadrature based scheme, we'd expect the error to exhibit some inverse factorial based dependence which it does. However, the error order is not directly computable and this is just a visualization.
- We see that as we increase the number of nodes, the error drops significantly as we'd expect.

2. [Rectangular rule]
Evaluate I = $\int_0^1 \frac{e^{-x}}{\sqrt{x}} dx$ by subdividing the domain into n $\in \{5, 10, 20, 50, 100, 200, 500, 1000, 2000, 5000, 10000\}$ panels.

- Using rectangular rule.
- Make a change of variable x = t^2 and use rectangular rule.

Compare the two methods above in terms of accuracy and cost. Explain the differences, if any.

Solution:
True value of the integral:
We obtain the true value of the integral from wolfram alpha and it turns out to be approximately 1.4936482656248540508.
Integral using substitution
On making the variable change x = t^2,dx = 2tdt and the limits remain the same.
Thus the integral becomes,

$$\int_0^1 \frac{e^{-x}}{\sqrt{x}} dx = \int_0^1 \frac{e^{-t^2}}{t} 2tdt$$
$$\Rightarrow \int_0^1 2e^{-t^2} dt$$

Rectangular Rule:
Rectangular rule is given as follows:

$$\int_a^b f(x)dx = h \cdot \left(\sum_{i=0}^{N} f(x_i) \right) + \mathcal{O}\left(h^2\right) \tag{5}$$

for a grid spacing of h and N panels. Therefore, applying the rectangular rule on the two integrals, we get the following equations

$$\int_0^1 \frac{e^{-x}}{\sqrt{x}} dx \approx h \cdot \left(\sum_{i=0}^{N} \frac{e^{-x_i}}{\sqrt{x_i}} \right)$$
$$\int_0^1 2e^{-t^2} dt \approx h \cdot \left(\sum_{i=0}^{N} 2e^{-x_i^2} \right)$$

Using these equations and plotting the error for the given number of panels we obtain the below figure

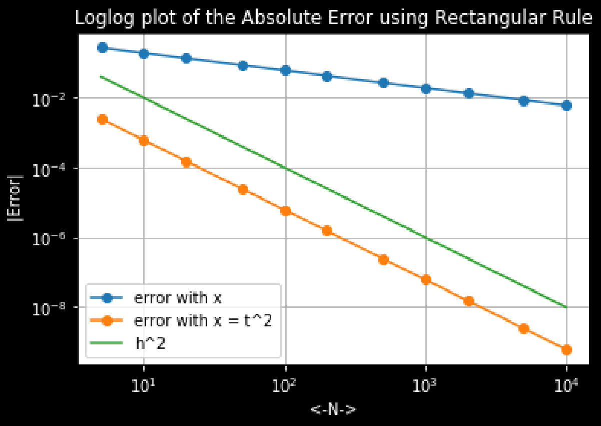


Figure 7: Plot of the error

Observations

- As far as computational cost is concerned, we make the same number of function evaluations in both cases. However, as we make an additional division and a square root computation in the first case, we expect it to be more expensive computationally to evaluate the integral in that manner when compared to the substitution case.
- The above intuition is substantiated by checking the running time of both the methods for 10000 panels as shown below.

```
Time taken without substitution = 0.011966943740844727
Time taken with substitution = 0.010973453521728516
```

- As far as the accuracy is concerned, we again see that the substitution case outshines the case without substitution.Moreover, we observe that the case without substitution doesn't even follow the error trends of Rectangular Rule!
- Our assumption of using taylor series to derive the Rectangular Rule might not be strictly valid at the endpoints as the function and it's derivative are unbounded at the endpoints.
- The variable change draws a parallel to the runge phenomenon, where the runge function was suddenly approximated well as we made the change from uniform nodes. Likewise, the variable change appropriately generates a new series of nodes which turn out to be better for the approximation.

3. [Householder's method]
The Householder's method is a generalization of the Newton method and the sequence of iterates is given by

$$x_{n+1} = x_n + d\frac{\left(\frac{1}{f}\right)^{d-1}(x_n)}{\left(\frac{1}{f}\right)^{d}(x_n)}$$

where $\left(\frac{1}{f}\right)^{k}(x_n)$ is the k^{th} derivative of the function $\frac{1}{f}$ evaluated at x_n. Note that taking d=1, we obtain the Newton method. Prove that if f(x) is d+1 times continuously differentiable function, i.e., $f^{(d+1)}$ exists and is continuous, and if the sequence of iterates converge to a root a, then we have

$$|x_{n+1} - a| \leq K\,|x_n - a|^{d+1} \qquad \text{for some K>0 eventually}$$

The above statement means that the order of convergence of the above method is d+1.

Solution:
Koenig's Theorem:
Let us have a look at Koenig's theorem which shall prove to be useful in computing the order of convergence of Householder's method.
Note: We assume that the zero is of multiplicity one to show the order of convergence. It is possible to expand this approach to zeros with a higher multiplicity.
Theorem:
Consider $h(x) = c_0 + c_1x + c_2x^2 + \ldots$, $c_0 \neq 0$ be *meromorphic* for $|x| < R$, and have a single simple pole at x = r.
If $|r| < \sigma R < R$, then, $\frac{c_v}{c_{v+1}} = r + \mathcal{O}\left(\sigma^{v+1}\right)$.
A *meromorphic* function in loose terms is essentially a function which is complex differentiable except at a set of points comprising the poles of the function.
A pole of a function is essentially a value that makes the function blow up to ∞, i.e. , its like a root of the denominator when the function is expressed as a ratio of two polynomials.
Proof:
Consider a function $\Psi(x) = (r - x)\cdot h(x) = b_0 + b_1x + b_2x^2 + \ldots$.

As the pole is of multiplicity one, $\Psi(x)$ is analytic for $|x| < R$.
For simplicity, let us rewrite the inequality constraint as follows:

$$\sigma^{-1}R < \rho < R$$

Therefore, $\Psi(\rho)$ converges and there exists a term of maximal absolute value (in the sequence $\{b_0, b_1\rho, b_2\rho^2, \dots\}$) which we denote by γ.
Therefore, for any v, we have,

$$|b_v| \leq \gamma\rho^{-v} \tag{6}$$

Now, compare the coefficients $\{b_i\}$ with $\{c_i\}$.
By our construction, we see that,

$$\begin{aligned}
&rc_0 = b_0 \\
&rc_1 - c_0 = b_1 \\
&\vdots \\
&rc_v - c_{v-1} = b_v \\
&\vdots
\end{aligned}$$

Let us just consider the first v+1 equations off the above set of equations.
Multiplying the i^{th} equation (i.e., equation involving the b_i term) with r^i and adding all the equations, we get a telescopic cancellation,

$$\begin{aligned}
&rc_0 = b_0 \\
&+ \\
&r(rc_1 - c_0) = rb_1 \\
&+ \\
&\vdots \\
&r^v(rc_v - c_{v-1}) = r^v b_v \\
&\Downarrow \\
&c_v r^{v+1} = b_0 + rb_1 + \dots + r^v b_v = \Psi_v(r)
\end{aligned}$$

Now consider a *remainder* term $R_v(r) = \Psi(r) - \Psi_v(r) = r^{v+1}(b_{v+1} + b_{v+2}r + \dots)$.
Using 6, we have,

$$\begin{aligned}
R_v(r) = \Psi(r) - \Psi_v(r) &= r^{v+1}(b_{v+1} + b_{v+2}r + \dots) \\
&\leq \gamma\left(1 + \frac{|r|}{\rho} + \frac{|r|^2}{\rho^2} + \dots\right)\frac{|r|^{v+1}}{\rho^{v+1}} \\
&\Rightarrow \gamma\frac{\left|\left(\frac{r}{\rho}\right)\right|^{v+1}}{1 - \left|\frac{r}{\rho}\right|} && \text{(Using sum of a GP,as } \sigma < 1\text{)} \\
&\Rightarrow \mathcal{O}\left(\left|\frac{r}{\rho}\right|^{v+1}\right)
\end{aligned}$$

But, we see that,

$$\begin{aligned}\frac{c_v}{c_{v+1}} &= r \cdot \frac{\Psi_v(r)}{\Psi_{v+1}(r)} \\ &\Rightarrow \frac{\left(1 - \frac{R_v(r)}{\Psi(r)}\right)}{\left(1 - \frac{R_{v+1}(r)}{\Psi(r)}\right)} \\ &\Rightarrow r\left(1 + \mathcal{O}\left(\left|\frac{r}{\rho}\right|^{v+1}\right)\right)\end{aligned}$$

Thus, from the choice of ρ we had, $\left|\frac{r}{\rho}\right| < \sigma$.Thus we've proved the theorem, i.e., $\frac{c_v}{c_{v+1}} = r + \mathcal{O}\left(\sigma^{v+1}\right)$.

Applying the theorem

Consider the function $\frac{1}{f(x)}$.

This function has a pole at x = a, which is a root of f(x).

It is given that f(x) is continuously differentiable d+1 times, therefore, we can expand $\frac{1}{f(x)}$ in the form of a taylor series about a point x_n as follows:

$$\frac{1}{f(x)} = \sum_{k=0}^{\infty} \frac{\left(\frac{1}{f}\right)^k (x_n)}{k!} \cdot (x - x_n)^k$$

This is eerily reminiscent of the function h(x) we saw in Koenig's theorem.

Consider a function h$(x - x_n) = \frac{1}{f(x)}$.

Now on this function, we can apply Koenig's theorem as

- $h(x - x_n) = \sum_{k=0}^{\infty} \frac{\left(\frac{1}{f}\right)^k (x_n)}{k!} \cdot (x - x_n)^k$ therefore a similar form as in the theorem where $x \approx x - x_n$ and $c_k = \frac{\left(\frac{1}{f}\right)^k (x_n)}{k!}$.
- This function has a single simple pole at x = $a - x_n$.
- This function is meromorphic around the neighbourhood of this pole.

Therefore, on the application of Koenig's theorem, we observe,

$$\begin{aligned}\frac{c_{d-1}}{c_d} &= r + \mathcal{O}\left(\sigma^{d+1}\right) \\ \frac{\frac{\left(\frac{1}{f}\right)^{d-1}(x_n)}{(d-1)!}}{\frac{\left(\frac{1}{f}\right)^{d}(x_n)}{(d)!}} &= a - x_n + \mathcal{O}\left(\sigma^{d+1}\right) \\ x_n + d \cdot \frac{\left(\frac{1}{f}\right)^{d-1}(x_n)}{\left(\frac{1}{f}\right)^{d}(x_n)} &= a + \mathcal{O}\left(\sigma^{d+1}\right).\end{aligned}$$

For sufficiently large n, we can choose $\sigma = |x_n - a|$.

On substituting this and the result derived above into the expression for Householder's method, we get,

$$\begin{aligned}x_{n+1} &= x_n + d\frac{\left(\frac{1}{f}\right)^{d-1}(x_n)}{\left(\frac{1}{f}\right)^{d}(x_n)} \\ x_{n+1} &= a + \mathcal{O}\left(\sigma^{d+1}\right) \\ x_{n+1} &= a + \mathcal{O}\left(|x_n - a|^{d+1}\right)\end{aligned}$$

We know that, $T(n) = \mathcal{O}\left(f(n)\right) : \exists\ c > 0, \exists\ n_0 > 0,$ S.T $\forall n \geq n_0, T(n) \leq c \cdot f(n)$, i.e., eventually, $\mathcal{O}\left(f(n)\right) \leq c \cdot f(n)$.
Using this in the expression above, we have,

$$\begin{aligned} x_{n+1} &\leq a + K\left|x_n - a\right|^{d+1} && \text{(For an appropriate constant K)} \\ x_{n+1} - a &\leq K\left|x_n - a\right|^{d+1} \\ \left|x_{n+1} - a\right| &\leq K\left|x_n - a\right|^{d+1} \end{aligned}$$

Which is the result we set out to prove.
Thus, we've shown that householder's method has an order of convergence of d+1.

4. [Newton's method]
Let f(x) be a twice differentiable strictly convex function with a single simple (i.e., multiplicity of the root is one) root at x=a. Prove that the Newton method converges to the root irrespective of the initial guess.

Solution:
Strictly convex function
Let us analyze the behaviour of the given f(x).
As f(x) is given to be a strictly convex function which is twice differentiable, we have $f^{"}(x) > 0\forall x$.
This means that $f^{'}(x)$ strictly increases with x and is continuous.
Let us try proving that $f^{'}(x)$ has to be strictly increasing or strictly decreasing.
Proof A:
We seek to prove that $f^{'}(x) > 0\forall x$ or $f^{'}(x) < 0\forall x$.
Let us try proving this by contradiction.
Let there exist an x_i such that $f^{'}(x_i) = 0$, this means that f(x) will be an increasing function past x_i and a decreasing function for all values of x $< x_i$.
We see that, if $f(x_i) <= 0$, then f(x) will have two repeated roots for the above reason and if $f(x_i) > 0$, then it will have no root at all which leads us to a contradiction as there exists a single simple root.
Therefore, as $f^{'}(x)$ is continuous, it has to be strictly increasing or strictly decreasing.
Proof B:
Newton's method is shown below

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \tag{7}$$

Where the sequence $\{x_n\}$ eventually should converge to the root a.
Let us prove this theorem into different cases based on the sign/value of the first derivative we saw in the proofs above.

- **Case I:**
  Let $f^{'}$ be positive and $f(x_0) \geq 0$.
  As f(x) is convex, the function lies above any of its tangent lines, i.e., $(x_{n+1}, f(x_{n+1}))$ will be higher than the point $(x_{n+1}, 0)$ as that is the x-intercept of the tangent line drawn to the function at x_n.Thus, we have $f(x_{n+1}) \geq 0\forall n$ which follows from induction as we only consider $f(x_0) \geq 0$.
  Now, for this case, we have $\frac{f(x_n)}{f^{'}(x_n)} > 0$ and thus, $x_{n+1} \leq x_n$.Therefore, the sequence $\{x_n\}$ is a decreasing one and it either converges or diverges to $-\infty$.
  If it converges, then we have $\lim\limits_{n\to\infty} f(x_n) = \lim\limits_{n\to\infty}(x_n - x_{n+1}) \cdot f^{'}(x_n) = 0$ or $\lim\limits_{n\to\infty} x_n = a$.
  Should the sequence diverge, then from the above limit we see that, f(x) should be strictly positive and not have any root to begin with which is a contradiction.
  Thus, we've established the convergence of the Newton method for this sub-case.

- **Case II:**
  This is a mirror image of Case I, where $f^{'}$ is negative and $f(x_0) \geq 0$.
  Thus the ingredients of the proof still remain the same, however, $\{x_n\}$ is now an increasing sequence due to the sign

change of the first derivative.
Therefore, as in the previous case, if the sequence converges, it converges to the root, else it diverges to ∞ in which case f(x) has no root.

As for the case where $f(x_0) < 0$, we'll still have $f(x_1) \geq 0$ as the graph of f(x) lies above the tangent line at x_0 and thus $(x_1, f(x_1))$ will be higher than the point $(x_1, 0)$ and thus the above proof follows.
Thus, we shown that if f(x) is a twice differentiable strictly convex function with a single simple root, the Newton method converges to the root for any arbitrary starting point.

5. [Root Finding]
Prove that the function w(x) = $xe^x - a$ has only one real root for a > 0.

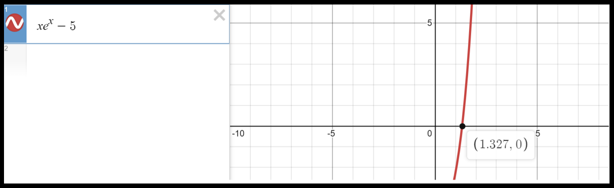
- Write a program to obtain the root of the above using (i) bisection (ii) Newton method (iii) Secant method.
- Explain in detail why,when and for what initial guess does each of the method converge.
- What happens when a < 0? Perform a complete analysis on the convergence for a < 0 as well.

Solution:
Proof for existence of root
Intuition: w(x) is nothing but the function $f(x) = xe^x$ shifted vertically downward by a.
We know that f(x) has a root at x = 0 and only one root exists, thus we expect the same trends to be followed by w(x). A more rigorous proof is shown below.

We see that as $x \to \infty, w(x) \to \infty$ and as $x \to -\infty, w(x) \to -a$.
Thus, there is a sign change as we progress from $-\infty$ to ∞ and hence there is a root which exists for this function.
To show there is only one root, let us take a look at its derivative $w^{'}(x) = (x+1)e^x$. We see that for $x > -1, w^{'}(x) > 0$ and $x < -1, w^{'}(x) < 0$.
Using this, we have the following inequality as $w(-\infty) = -a$,

$$w(-1) = -\frac{1}{e} - a \leq w(x) < -a \forall x \in (-\infty, -1]$$

Now, if a is positive, then as $w(x) < -a$ in the above interval, there exists no root in the interval $(-\infty, -1]$.
We also observe that beyond -1, the derivative is strictly positive and thus w(x) is strictly increasing. As w(x) goes to ∞ at $x = \infty$, we see that there is strictly one zero crossing from $(-\infty, \infty)$ thus only one real root exists.
In the plot below, this is visualized with a = 5



$xe^x - 5$

Figure 8: $xe^x - 5$

The three methods:

- For the above case, when $a > 0$, the bisection method is the slowest of the lot followed by secant and newton's method.
- The bisection method however doesn't run into problems and has a significantly fast rate of convergence provided good starting points are found.

- The newton's and secant method however run into problems if the starting points aren't chosen carefully. The newton's method has an issue if we start off with a point where the derivative is very close to 0 and the secant's method has a cycling issue where it might revisit previous iterates, however, this can be easily fixed by having a set of iterates that have been already visited.
- The secant method can also explode when at some point of time, the iterates reach values close to -1.

Some examples of the above cases:
An example of bisection method with starting points 0 and 1.5.

```
Estimate at step 1 is 1.125
Estimate at step 2 is 1.3125
Estimate at step 3 is 1.40625
Estimate at step 4 is 1.359375
Estimate at step 5 is 1.3359375
Estimate at step 6 is 1.32421875
Estimate at step 7 is 1.330078125
Estimate at step 8 is 1.3271484375
Estimate at step 9 is 1.32568359375
Estimate at step 10 is 1.326416015625
Estimate at step 11 is 1.3267822265625
Estimate at step 12 is 1.32659912109375
Estimate at step 13 is 1.326690673828125
Estimate at step 14 is 1.3267364501953125
Estimate at step 15 is 1.3267135620117188
Estimate at step 16 is 1.3267250061035156
```

An example of newton's method starting at x = 0

```
Estimate at step 1 is 5.0
Estimate at step 2 is 4.172281622499238
Estimate at step 3 is 3.38052341965179
Estimate at step 4 is 2.6476486189384154
Estimate at step 5 is 2.018870692039057
Estimate at step 6 is 1.5700790618651563
Estimate at step 7 is 1.363884864615359
Estimate at step 8 is 1.3276915746684355
Estimate at step 9 is 1.3267253332433395
```

An example of secant method with the same starting points of the bisection method

```
Estimate at step 1 is 1.115650800742149
Estimate at step 2 is 1.3004697589630294
Estimate at step 3 is 1.3309604691044898
Estimate at step 4 is 1.3266446694662424
Estimate at step 5 is 1.326724423285968
```

On taking x = -1 for newton's method and the starting points -2,2 for secant method, we see that the methods don't converge to the root.
Convergence Analysis for $a < 0$
From the inequalities derived, we see that no root can exist if $a < -\frac{1}{e}$ as then w(x) will always be positive.
The secant and the newton's method will mindlessly iterate in this case while the bisection method can not be initialized.
When $a > -frac1e$ we observe the existence of two roots.
The newton's method generally converges to the root closer to the initial point and a similar phenomenon is observed for the secant method too.So we cannot directly ascertain which root they converge to.
Some examples are shown below for a = -0.1,

```
Bisection method with starting points -1,0
Estimate at step 1 is -0.25
Estimate at step 2 is -0.125
Estimate at step 3 is -0.0625
Estimate at step 4 is -0.09375
Estimate at step 5 is -0.109375
Estimate at step 6 is -0.1171875
Estimate at step 7 is -0.11328125
Estimate at step 8 is -0.111328125
Estimate at step 9 is -0.1123046875
Estimate at step 10 is -0.11181640625
Estimate at step 11 is -0.112060546875
Estimate at step 12 is -0.1119384765625
Estimate at step 13 is -0.11187744140625
Estimate at step 14 is -0.111846923828125
Estimate at step 15 is -0.1118316650390625


Newton method with starting point -5
Estimate at step 1 is -2.539671022435586
Estimate at step 2 is -3.365900113544329
Estimate at step 3 is -3.5645327038276275
Estimate at step 4 is -3.5771035619400413

Secant method with starting points 0,-2
Estimate at step 1 is -0.7389056098930651
Estimate at step 2 is -4.616556552585879
Estimate at step 3 is -3.9306210052027537
Estimate at step 4 is -3.4336796109571908
Estimate at step 5 is -3.593218606737492
Estimate at step 6 is -3.5778407276753463
Estimate at step 7 is -3.577148667939669
```