7. Methods

7.5 Pade Approximants

Question 1:

a) $f_1(x) = (1+x)^{\frac{1}{2}}$

$f_1(0) = 1$

$f'_1(0) = \frac{1}{2}\left(1+(0)\right)^{-\frac{1}{2}} = \frac{1}{2}$

$f''_1(0) = -\frac{1}{4}\left(1+(0)\right)^{-\frac{3}{2}} = -\frac{1}{4}$

$$f_1^{(k)}(0) = (-1)^{(k)}2^{-k}\prod_{n=0}^{k-1}(2n-1) \qquad for\ k \geq 1$$

Therefore,

$$c_k = \frac{1}{k!}(-1)^k(2^{-1})^{(k)}\prod_{n=0}^{k-1}(2n-1) \qquad for\ k \geq 1$$

and $c_0 = 1$.

b) Radius of convergence: 1; The distance to the nearest singularity/branch point for $f_1(x)$ is 1 at $x = -1$ on a complex plane. We will require |x| <1 when we use the power series to estimate $f_1(x)$ otherwise the approximation diverges.

$\sqrt{2} = 1.414213562$

N	$\sum_{k=0}^{N} c_k$	Error
5	1.42578125	-0.011567687
10	1.409931182861328	0.0042823795
20	1.412667185988539	0.0015463763
50	1.413817654785574	0.0003959075
100	1.414073047717716	0.0001405146
150	1.414136978762613	0.0000765836

Table 1: partial sums as N increases

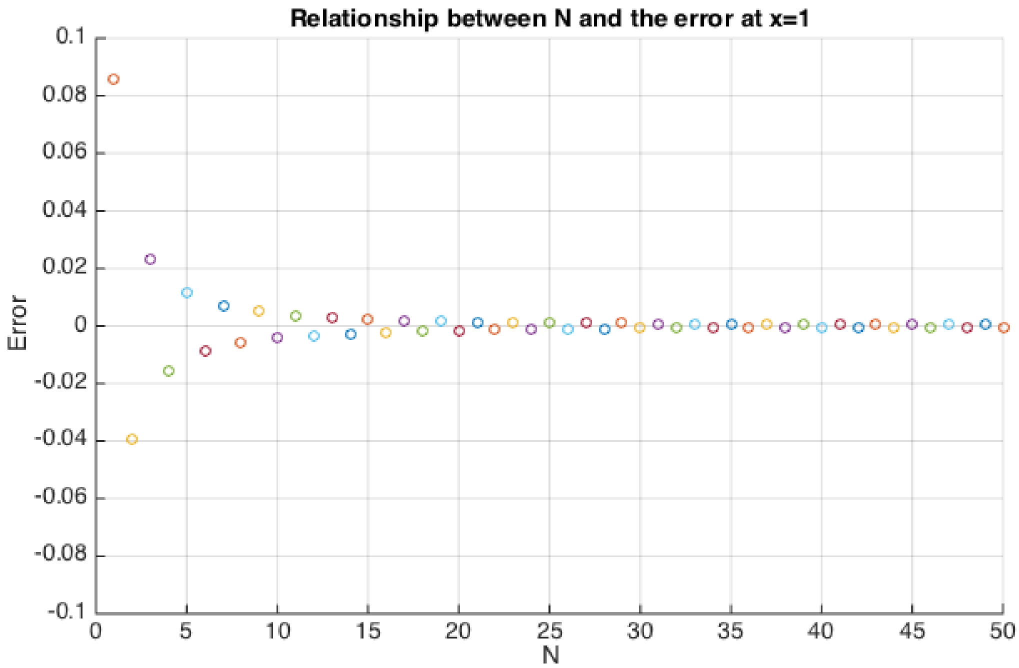


Figure 1: Relationship between N and the error at x=1

From the graph, the magnitude of the error decreases exponentially as the value of N increases linearly and it also shows a sign changing oscillation on the error as N increases and those results can be explained from the coefficient formula.

Deriving from the formula, we get:

$$c_k = \frac{1}{k}(-1)(2^{-1})(2(k-1)-1)c_{k-1} = Ac_{k-1} \;\; for\ k \geq 1$$

a) $|A| = \left|\frac{2k-3}{2k}\right| < 1$; b) $|A| \to 1\ as\ k \to \infty$; c) factor of (-1)

From b) and c), we can deduce that the partial sum should converges as N tends to large because each of the successive terms are (nearly fully) cancelling each other out. Due to the three conditions and as $Error = \sum_{i=N+1}^{\infty} c_i$, the error should decreases and shows a sign-changing oscillation as N increases.

Question 2: (assumed L=M unless indicated otherwise)

L	$Pade\ Approximant\ R_{L,L}$	Error
3	1.414201183431953	1.237894114258786e-05
5	1.414213551646055	1.072704036708672e-08
8	1.414213562372821	2.740030424774886e-13
10	1.414213562373095	4.440892098500626e-16
15	1.414213562373095	-2.22044604925031e-16
20	1.414213562373095	-2.22044604925031e-16
25	1.414213562373094	-8.88178419700125e-16
30	1.414213562373095	-4.44089209850063e-16
100	1.414213562373095	-2.22044604925031e-16

Table 2: Pade Approximation's Error as L increases at x=1

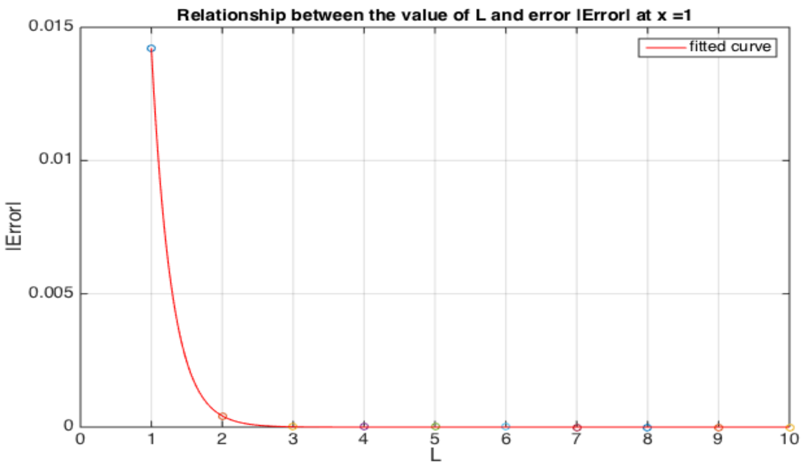


Figure 2: Relationship between L and the magnitude of error at x=1

From the graphs, we can see the error reduce significantly as L increases. By plotting a curve of best fit, we can see there is an exponential decreases as L increases. However, the error did not improve much as before for L greater than 10, we shall investigate more on this and on why the change in magnitude of the error comparing with the power series later in this project.

I cannot choose the error to be arbitrarily small as the error stays stable at 10^{-16} no matter how large L we choose for the approximation. Therefore, the smallest value to which the error can be reduced is around $2 * 10^{-16}$ and this smallest value determines by the size of L we choose.

Iteration	Error
1	-7.74176016782486e-06
	-3.29839747388397e-05
	-5.83636007991754e-05
	-5.54853634540787e-05
	-3.06133802125115e-05
	-9.87344024525672e-06
	-1.77853798783430e-06
	-1.59387396810016e-07
	-5.45320171667412e-09
	-3.03853901893799e-11

Iteration	Error
5	-4.72854468191580e-06
	-2.06209292585062e-05
	-3.74569167361647e-05
	-3.66843198878325e-05
	-2.09406993285380e-05
	-7.02491577216107e-06
	-1.32511475101285e-06
	-1.25440866429095e-07
	-4.58603579370395e-09
	-2.77399469363511e-11

Iteration	Error
10	-7.87637958771997e-06
	-3.36853673490672e-05
	-5.98672165168747e-05
	-5.72078541291620e-05
	-3.17561005812611e-05
	-1.03169513329921e-05
	-1.87503877733599e-06
	-1.69904406812987e-07
	-5.89513631177618e-09
	-3.34478689712798e-11

After running the iterative improvement, it does not improve the error much after 5 iterations for L=M=10. The methods did not show any significance of improvement on the error as the number of iteration increases.

Comparing the tables from Q1 and Q2, we can clearly see that Pade approximation converges to the actual value quicker than the power series approximation by far. From the samples I generated, the error of Pade approximation is at least less than 10^{-5} while for power series, we will need up to and include the 150 terms for an approximation having a same accuracy. Therefore, regarding to get the most accurate approximation with limited terms involved, I will suggest the use of Pade approximation over the power series.

If we aim to estimate for $\sqrt{2}$ to specified accuracy, I will suggest the use of power series because it is easier to identify the exact number of terms which we need to include for an estimation to a specified accuracy and we cannot reduce any further error with magnitude smaller than $2 * 10^{-16}$ in Pade approximation as suggested in part 1. With the use of the recurrence formula stated in Q1, we can easily find the smallest N such that $|A^N|$ is smaller than the required accuracy.

In addition, Pade approximation works at order o(M(M+L)) which includes solving M simultaneous equations for Q and L+1 calculations for P while power series requires only N calculations for each approximation. It requires more work for Pade approximation than power series although Pade approximation gives us an incomparably more accurate result than using power series up to the nearest $2 * 10^{-16}$.

Question 3:

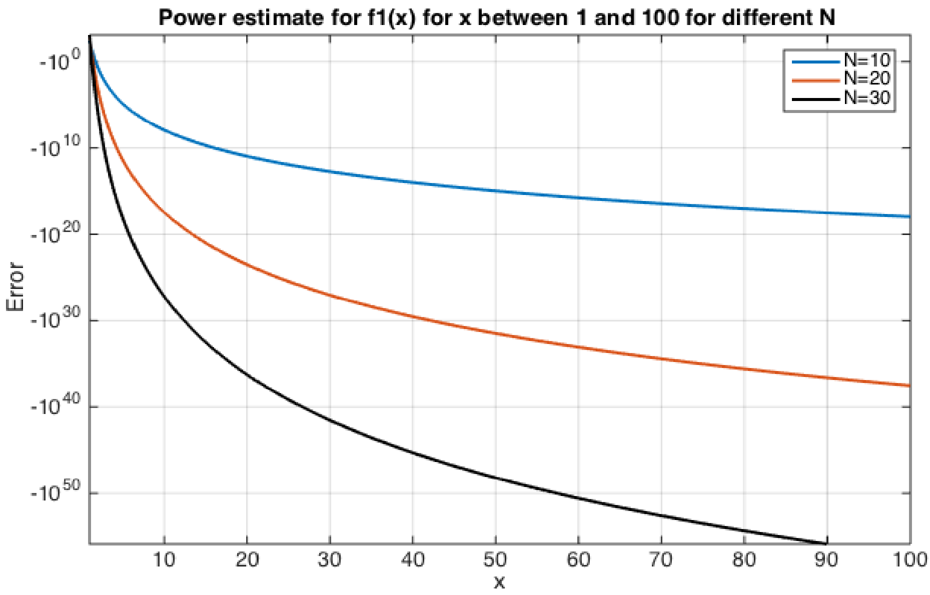


Figure 3: Power Approximation for f_1 for x between 1 and 100 for different N (even) (Semi-Log plot)

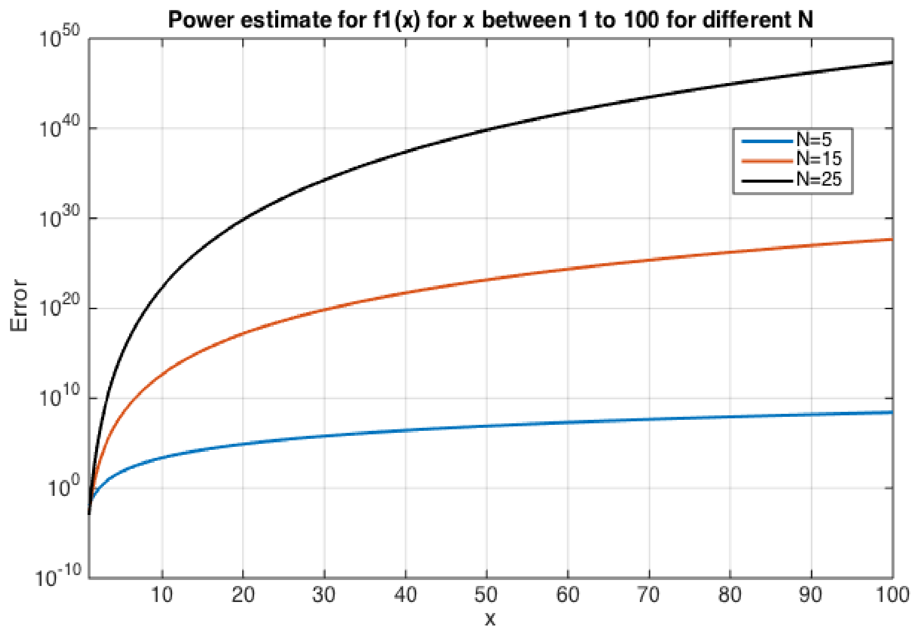


Figure 4: Power Approximation for f_1 for x from 1 to 100 for different N (odd) (Semi-Log plot)

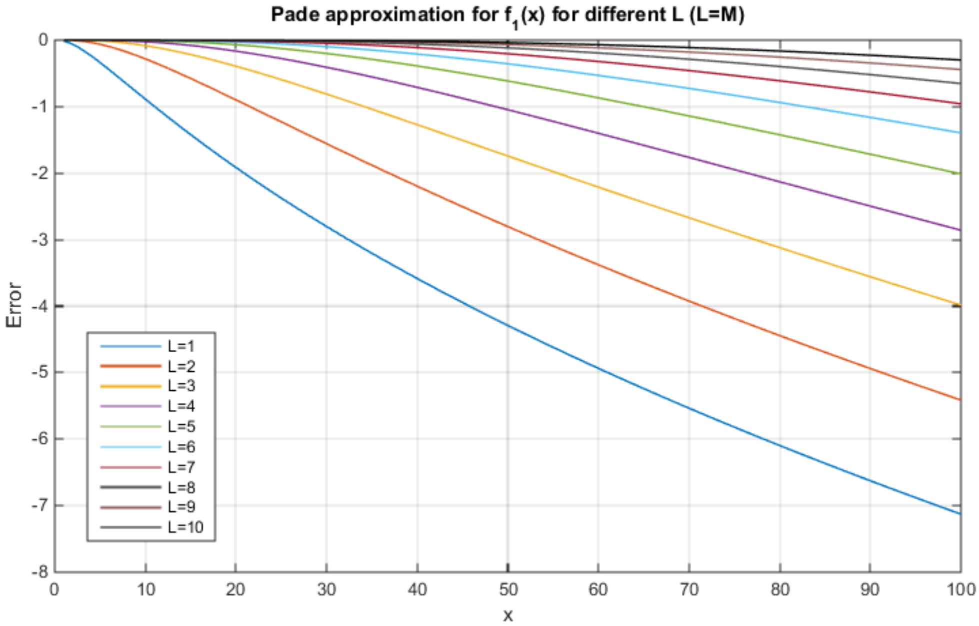


Figure 5: Pade approximation for f_1 with different M as x varies

Characteristics obtained from the graphs:

For power series approximation:

1a) the magnitude of the error increases exponentially as x increases linearly

2a) the magnitude of the error increases exponentially as N increases linearly

3a) a sign-changing oscillation on the error as N increases

For Pade approximation:

1b) the magnitude of the error increases linearly for large x

2b) the magnitude of the error decreases at inverse logarithmical manner as L increases linearly

3b) the rate of the magnitude of the error decreases as x increases linearly for L. For small L, the error increases in an exponential manner; as L increases, the error curve tends to a linear style.

Comparing two graphs, we can see the magnitude of the error in power series is greater than in Pade approximation overall from x=1 to 100. The difference in error from the two approximations diverges in an exponential manner as x increases with the range of error for Pade approximation and power series approximation varies from 0 to -8 and 0 to 10^{47} respectively. This shows Pade approximation gives higher accuracy than power series approximation.

Observation 1a) and 1b) can be explained by the fact that the rate of error reduces/ increases is affected by the distance between the evaluated point and the radius of convergence. The further we evaluated from the radius of convergence, the quicker the error increases.

There is a significant difference between the number of terms being considered and the magnitude of error within two approximation. For x greater than 1, the error in the Pade approximation decrease as more terms being considered while the error in power series approximation increases in an exponential manner as N increases. This is because the more terms we accepted in power series (which means the expansion is at higher order of x), the greater divergence we get as x being greater than 1 and the coefficient of the expansion does not tend to 0 as quick. The exponential increases can be explained by

$$\sum_{i=0}^{N} a_i(x)^i - \sum_{i=0}^{N} a_i(x+\varepsilon)^i = o(x^{N-1})$$

(We will discuss about the reason of the rate of convergence of Pade approximation later in this project.)

On the other hand, from the graph, we can see the error of the power series approximation oscillates between positive and negative and N increases. This could be because the coefficient of the power series we used to approximate has a factor of $(-1)^N$ and given that the series diverges, the leading term of the polynomial dominates the sign of the series which caused oscillate around the actual result as N changes.

In conclusion, Pade approximation gives a better approximation than power series although both approximation shows some extends of divergence from the actual result from the graph as x increases which is due to the distance from the evaluated point to the radius of convergence increases.

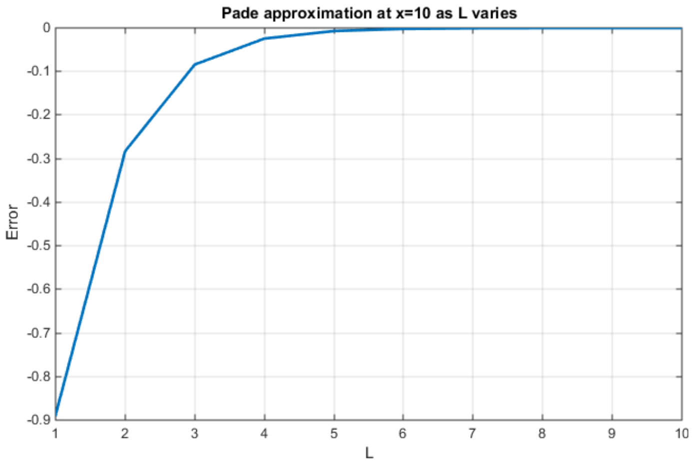


Figure 7: Pade approximation at x=10 as L varies (Linear plot)

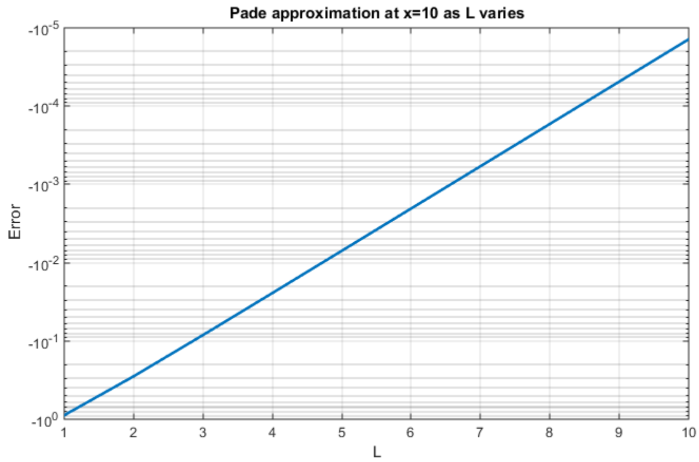


Figure 8: Pade approximation at x=10 as L varies (Semi-Log plot)

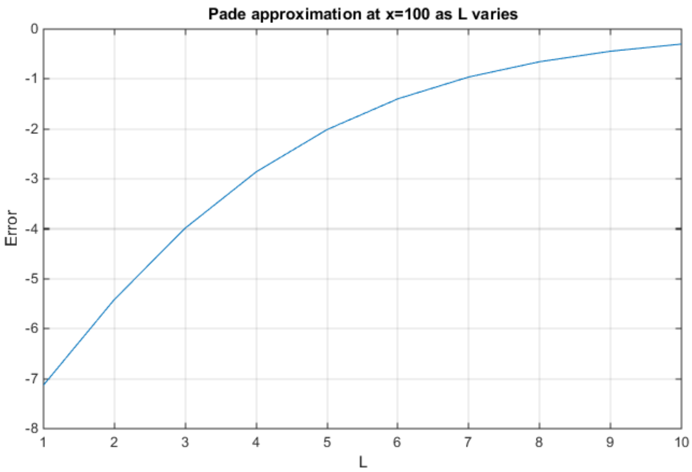


Figure 9: Pade approximation at x=100 as L varies (Linear plot)

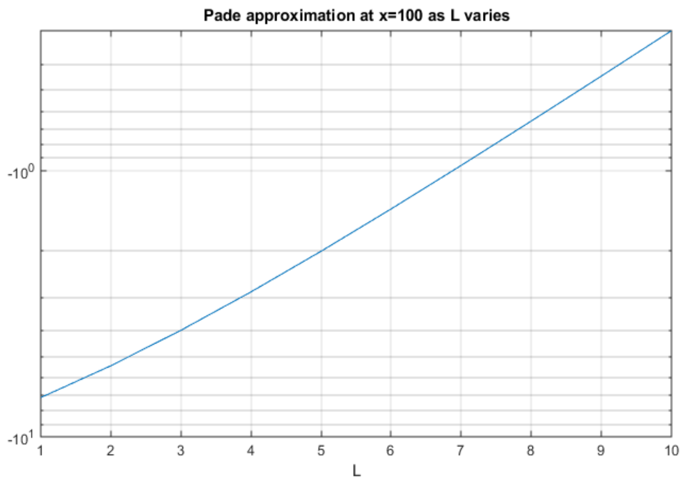


Figure 10: Pade approximation at x=100 as L varies (Semi-Log plot)

Similar to part 1, from the two we can see the magnitude of the error decreases at inverse logarithmical manner as L increases and although the relationship between the size of error and the value of L becomes less strong for large x.

From part 1 (condition 3b), we understand that the value of L needs to be relatively large (e.g. 10) as the error increases in exponential manner for small L and as L increases, the error increases in a logarithmic manner as x increases.

Therefore, to estimate $f_1(x)$ for large x with Pade approximation, the value of L needs to be relatively larger in order to achieve a fixed accuracy.

Question 4:

X	Actual Value	Pade Approximation (L=M=10)			Power Series Approximation (N=15)		
	$f_2(x)$	$f_2(x)$	Error	(%)	$f_2(x)$	Error	(%)
0.1	0.91563334	0.915633	-6.06019E-08	-6.61858E-06	0.91484044	-0.00079	-0.0866
0.2	0.85211088	0.852111	1.79275E-09	2.1039E-07	-31.42068165	-32.2728	-3787.39
0.3	0.80118628	0.801186	1.3659E-08	1.70484E-06	-15385.95186	-15386.8	-1920496
0.4	0.75881459	0.758815	1.32262E-07	1.74301E-05	-1205207.548	-1205208	-1.6E+08
0.5	0.72265723	0.722658	6.24763E-07	8.64536E-05	-35245832.94	-3.5E+07	-4.9E+09
0.6	0.69122594	0.691228	1.98126E-06	0.000286629	-553748521.2	-5.5E+08	-8E+10
0.7	0.66351027	0.663515	4.91558E-06	0.000740844	-5.67E+09	-5.7E+09	-8.5E+11
0.8	0.6387911	0.638801	1.02389E-05	0.001602852	-4.25E+10	-4.2E+10	-6.7E+12
0.9	0.61653779	0.616557	1.88182E-05	0.003052243	-2.51E+11	-2.5E+11	-4.1E+13
1	0.59634736	0.596379	3.15238E-05	0.005286148	-1.23E+12	-1.2E+12	-2.1E+14
2	0.46145532	0.461960	0.000505175	0.109474403	-4.15E+16	-4.1E+16	-9E+18
3	0.38560201	0.387274	0.001671949	0.433594569	-1.84E+19	-1.8E+19	-4.8E+21
4	0.33522136	0.338565	0.003343908	0.997522332	-1.38E+21	-1.4E+21	-4.1E+23
5	0.29866975	0.303969	0.005299605	1.774402977	-3.94E+22	-3.9E+22	-1.3E+25
6	0.27063301	0.278017	0.007384201	2.728492272	-6.08E+23	-6.1E+23	-2.2E+26
7	0.24828135	0.257783	0.009501752	3.827009971	-6.15E+24	-6.1E+24	-2.5E+27
8	0.22994778	0.241543	0.01159563	5.042723063	-4.56E+25	-4.6E+25	-2E+28
9	0.2145771	0.228211	0.013633636	6.353723642	-2.67E+26	-2.7E+26	-1.2E+29
10	0.20146425	0.217063	0.015598475	7.7425525	-1.30E+27	-1.3E+27	-6.4E+29
11	0.19011779	0.207600	0.017481876	9.195286529	-5.43E+27	-5.4E+27	-2.9E+30
12	0.18018332	0.199464	0.019281009	10.70077346	-2.00E+28	-2E+28	-1.1E+31
13	0.171398	0.192394	0.020996304	12.25002849	-6.66E+28	-6.7E+28	-3.9E+31
14	0.16356229	0.186192	0.022630063	13.83574571	-2.02E+29	-2E+29	-1.2E+32
15	0.15652164	0.180707	0.02418564	15.45194661	-5.70E+29	-5.7E+29	-3.6E+32
16	0.15015426	0.175821	0.025666934	17.09371014	-1.50E+30	-1.5E+30	-1E+33
17	0.14436271	0.171441	0.027078074	18.75697256	-3.73E+30	-3.7E+30	-2.6E+33
18	0.13906806	0.167491	0.028423156	20.43830595	-8.79E+30	-8.8E+30	-6.3E+33
19	0.13420555	0.163912	0.029706212	22.1348607	-1.98E+31	-2E+31	-1.5E+34
20	0.12972152	0.160653	0.030931099	23.84423137	-4.27E+31	-4.3E+31	-3.3E+34

Table 3: Approximation of f_2(x) with the use of Pade approximation and Power series approximation

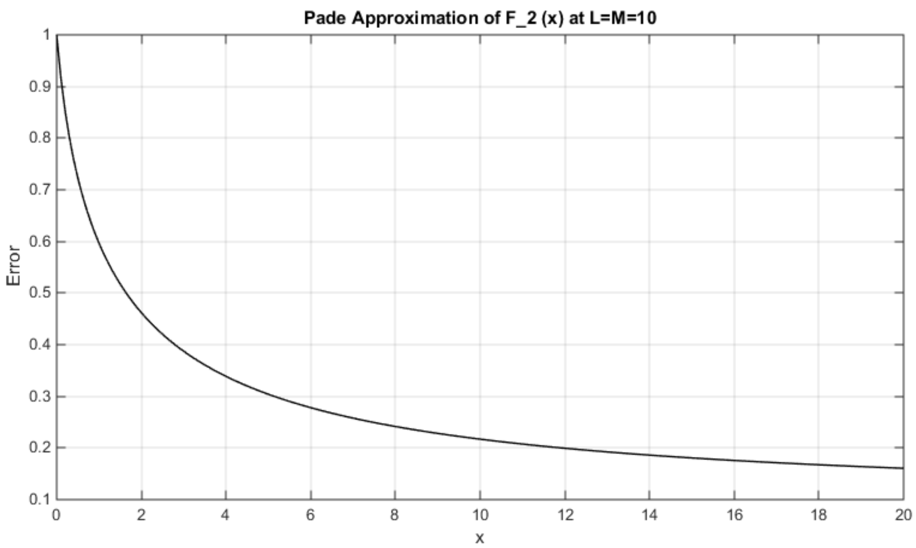


Figure 11: Pade approximation to f_2(x) between x=0.1 to 20 at L=M=10

From the table, we can easily identify that the relative error in the truncated power series is, again, incomparably large (varying from 10^9 – 10^{32}) comparing with the relative error occurs in Pade approximation for all value of x across 0.1 to 20. Although the error in power series approximation starts off relatively low when x is small, it experiences an exponential increase as x increases and diverges to over 10^{31}. This is because the partial sum of the power series diverges for any non-zero x. The series has a better approximation if we include less term in the series with small x because each extra term we consider, we are adding/subtracting a value of $k!\,x^k = kx(k-1)!\,x^{k-1} = (kx-1)(k-1)!\,x^{k-1} + (k-1)!\,x^{k-1}$ to the partial sum, which substantially means cancelling out the last term of the previous partial sum and add back$|(kx-1)(k-1)!\,x^{k-1}|$ to the magnitude of the partial sum. As the partial sum now has the magnitude of $|(kx-1)(k-1)!\,x^{k-1}| = |(kx-1) * c_{k-1}|$, the partial sums should remain stable and relatively accurate up to the smallest value of k such that $kx > 1$. Example as followed:

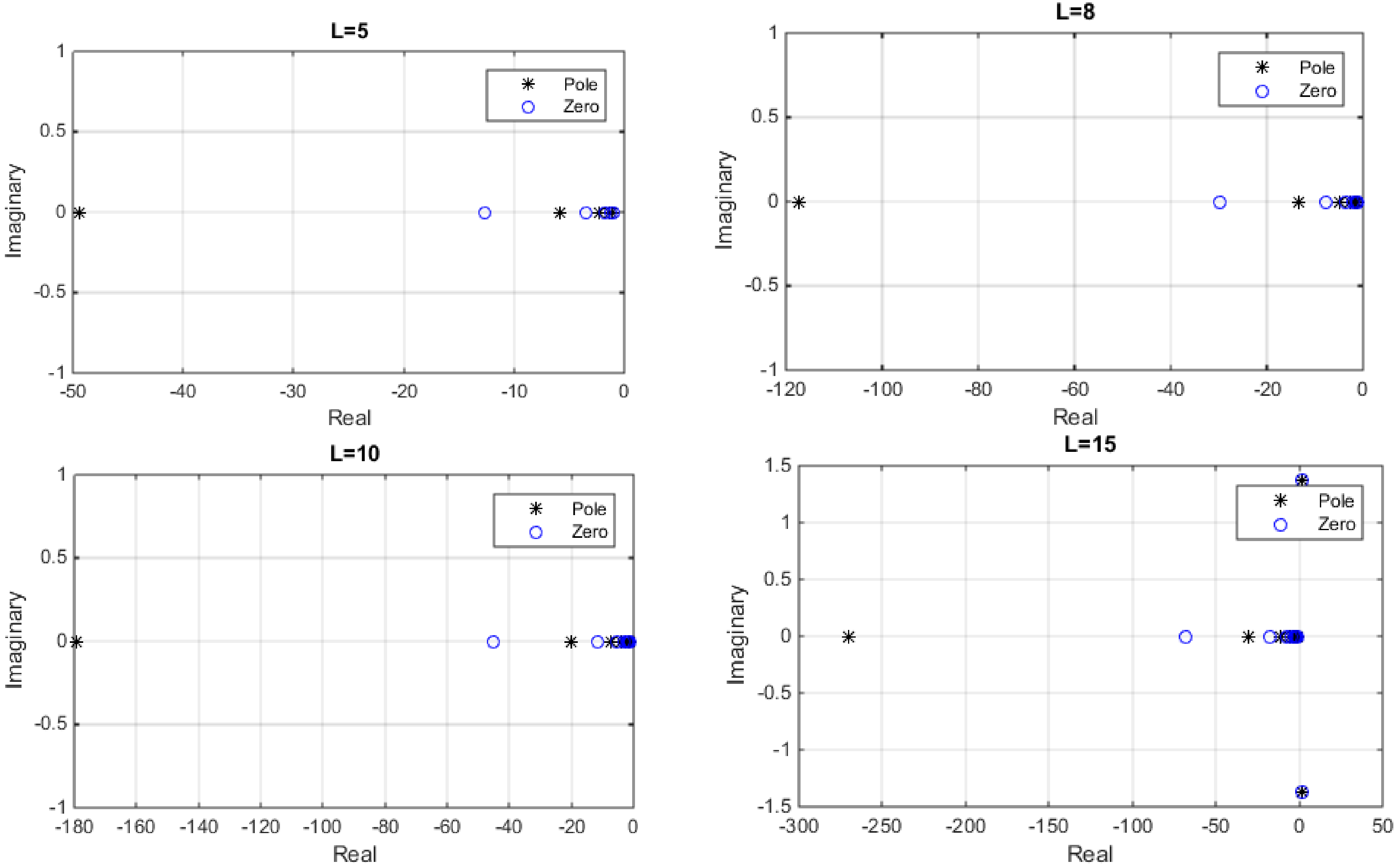
X	Actual	N=3	N=4	N=5	N=6
0.1	0.91563334	0.914000	0.916400	0.91520	0.915920
0.2	0.85211088	0.832000	0.870400	0.83200	0.878080
0.3	0.80118628	0.718000	0.912400	0.62080	1.145680

From the integral expression of $f_2(x)$, we can deduce that $f_2(x)$ is a decreasing function in x. However, according to the power series, it shows a divergent behaviour as x increases. Therefore, as a basis for calculating $f_2(x)$, truncated power series do not give much useful information at all which can also be reflected on the magnitude of error in the table.

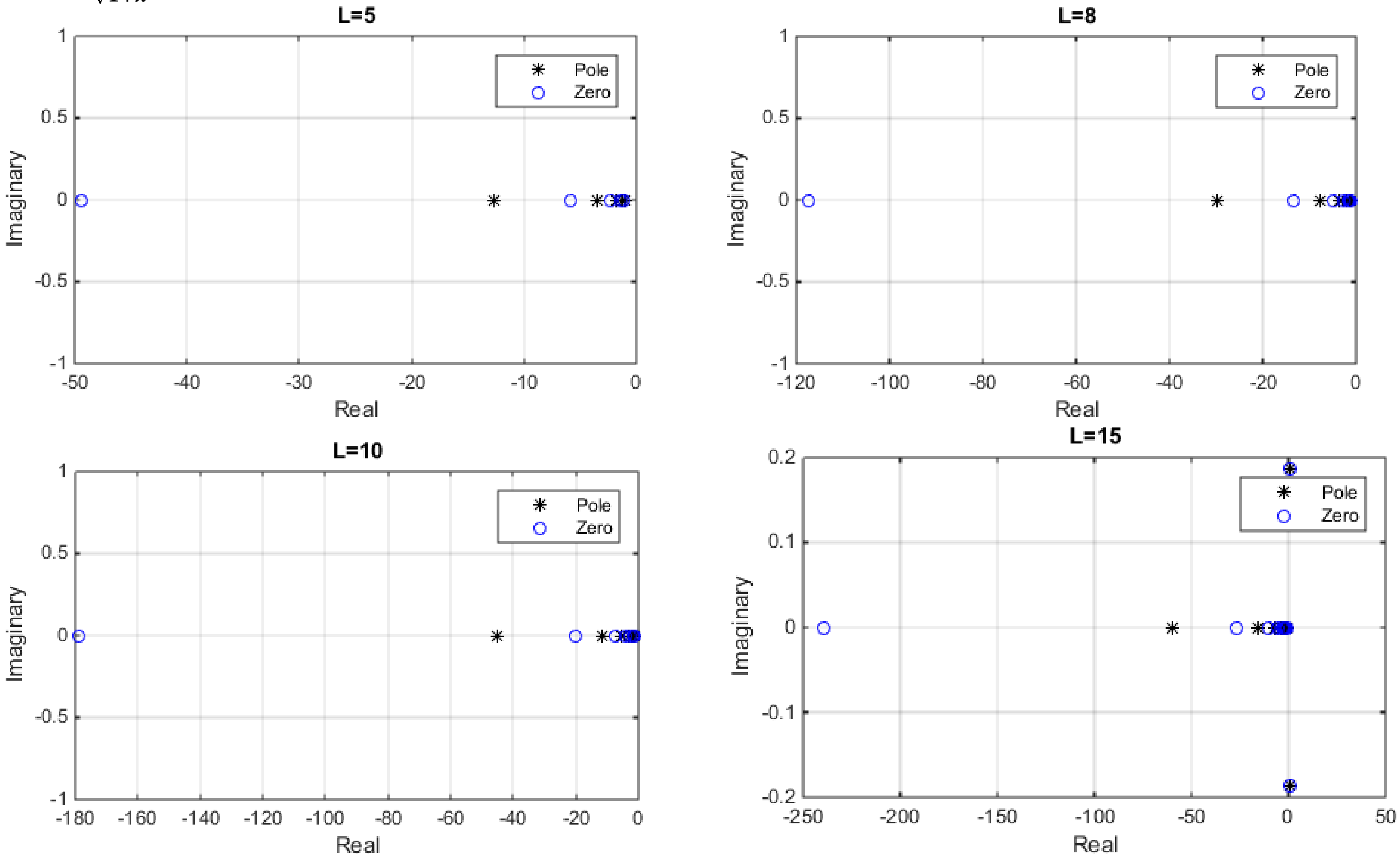
Regarding the use of Pade approximation to $f_2(x)$, it shows a promising accuracy to the actual value as x increases. A clear trend of $f_2(x)$ being a monotonic decreasing function in x is clearly reflected from the table which matches with our observation of the behaviour of $f_2(x)$. As x increases, although the relative error increases, it still under a reasonable range at x=20 with around 24%. I believe the relative error will increase as x increases but a clear sign of $f_2(x)$ converging to 0 is spotted from the approximation.

Even though the computational order of Pade approximation is higher than the power series approximation, as a basis for calculating $f_2(x)$, we should use the Pade approximation over the power series approximation as we can get barely any information from the latter method.

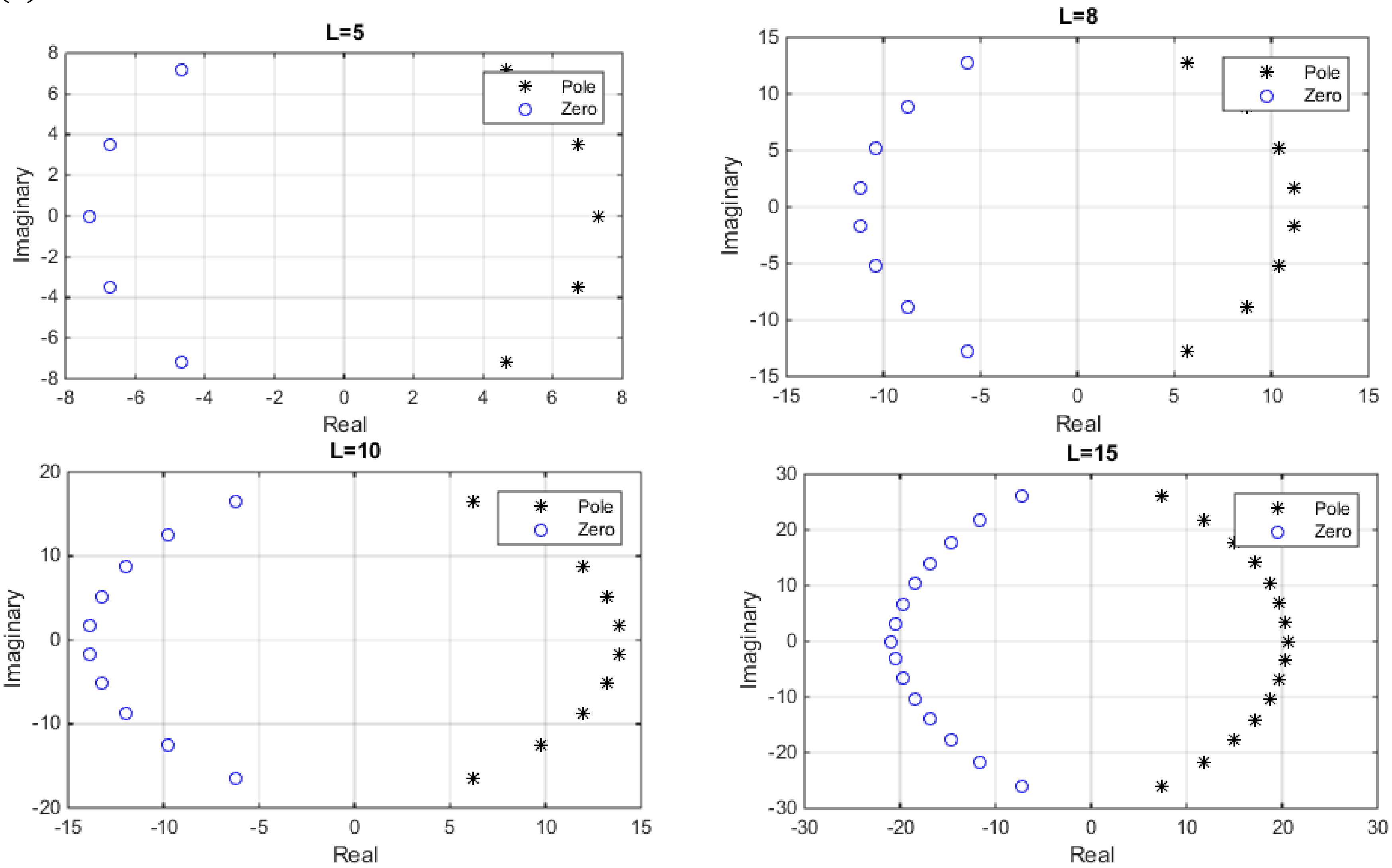
$f_1(x) = \sqrt{1+x}$



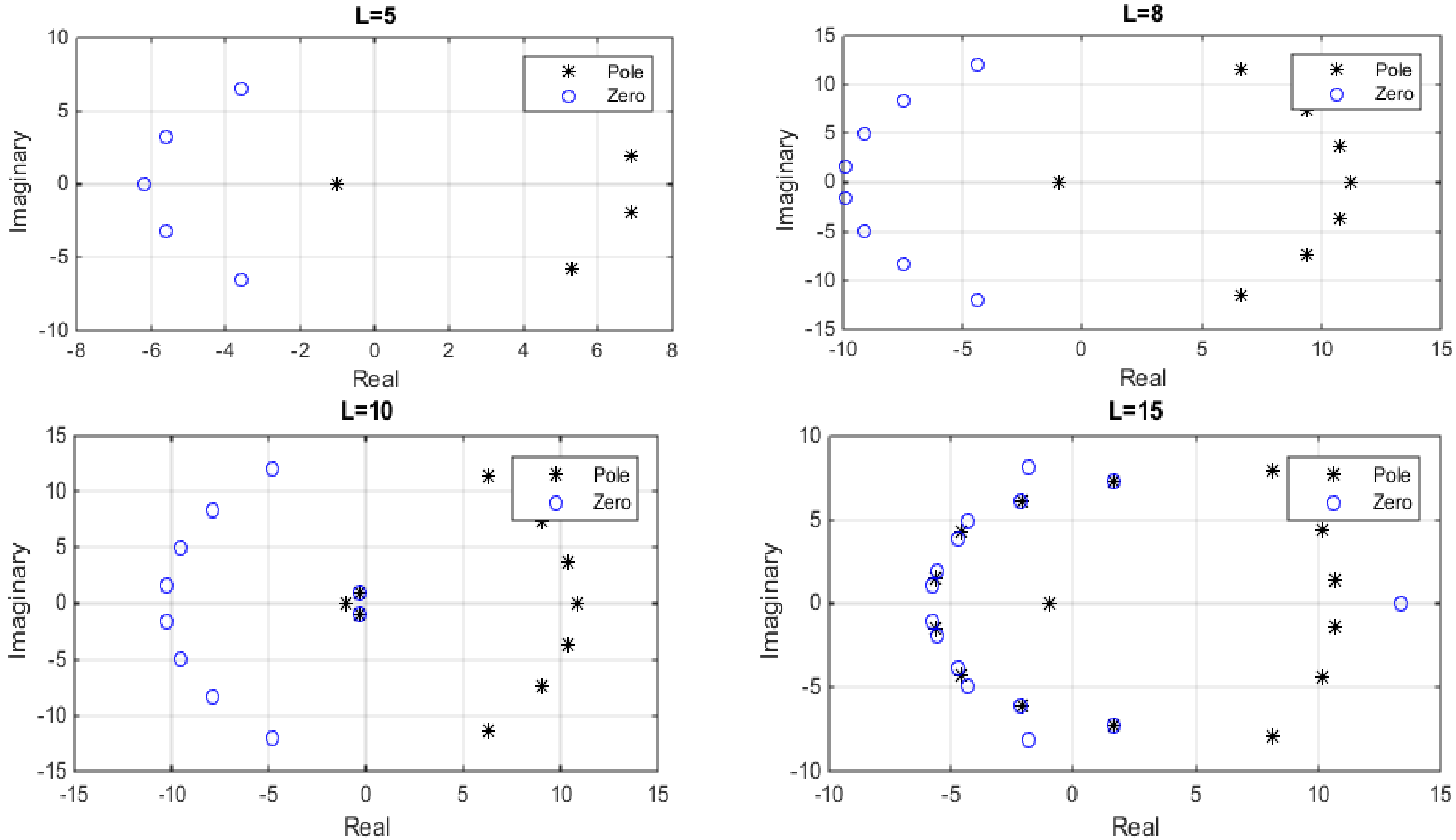
$$f_3(x) = \frac{1}{\sqrt{1+x}}$$



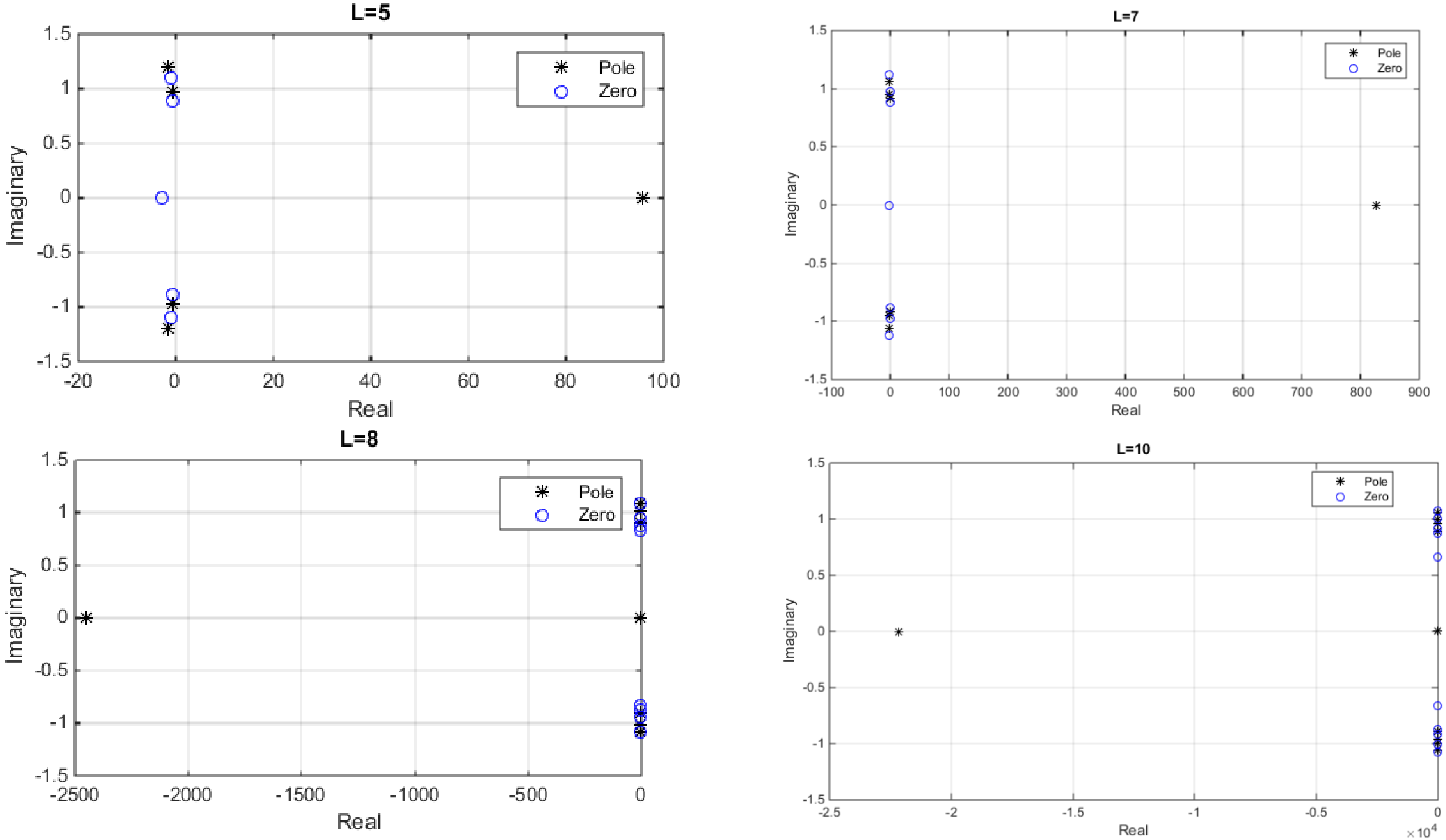
$f_4(x) = e^x$



$$f_5(x) = \frac{e^x}{1+x}$$



$f_6(x) = (1 + x + x^2)^{\frac{1}{2}}$



Question 5:

From the graphs for $f_1(x)$, we can see the position of the poles and zeros have kept the same pattern of distribution along the negative real axis starting from -1 and reaching out to negative infinity as L increases. The poles and zeros only exist on the left of -1 on the negative real axis for L smaller than 10. As L reaches 10, the poles and zeros start distributing not only on the negative real axis but also over the complex plane. However, we can clearly see that the poles and zeros are always overlapping each other on the complex plane at positions except the negative real axis. This could make those poles cancelled out by the zeros which enable us to ignore those poles when we considering the radius of convergence of the approximation of $f_1(x)$.

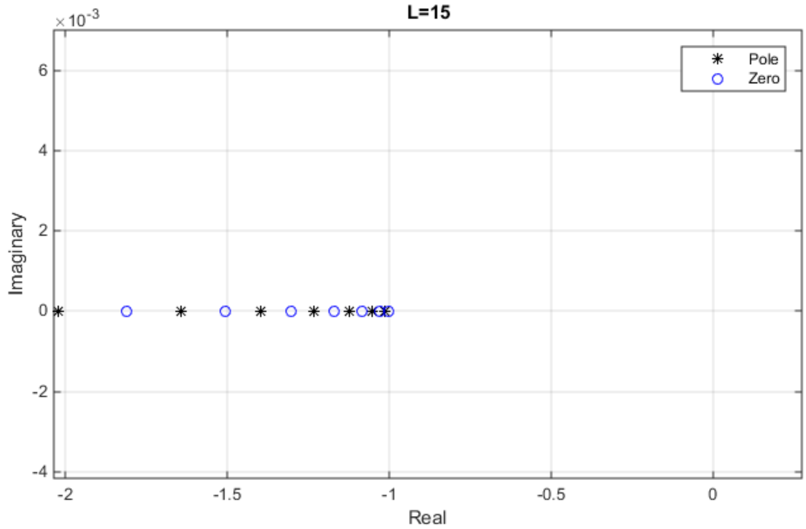
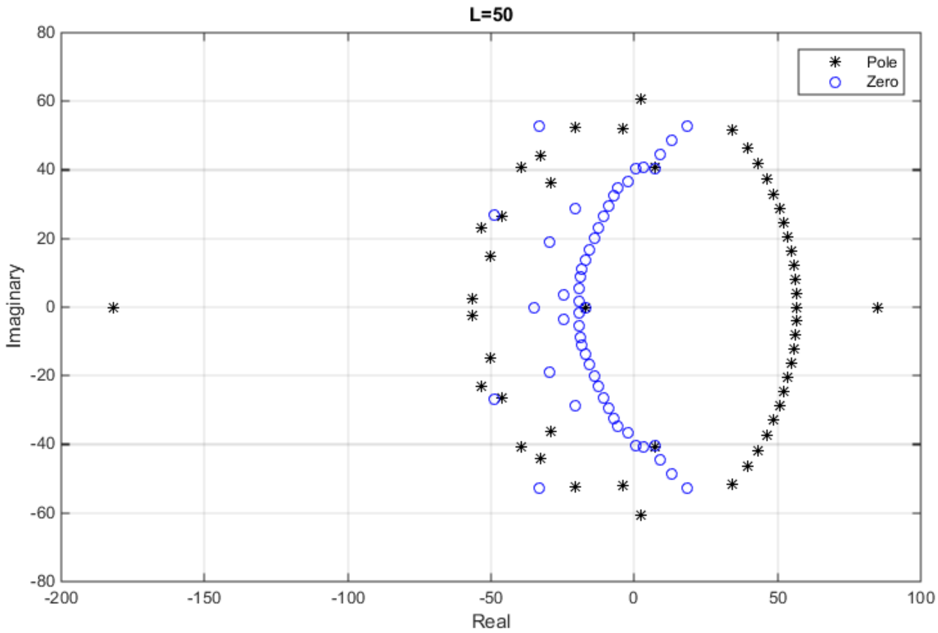


Figure 12: f_1 at L=15 (zoom in)

From the graphs for $f_3(x)$, we can spot the four graphs is fundamentally the same as the graphs for $f_1(x)$ with the position of the poles and zeros swapped. This is because $f_3 = \frac{1}{f_1}$, therefore the Pade approximation for f_3 is also the reciprocal of the approximation for f_1. Similar trend has shown here as in $f_1(x)$, the poles and zeros hold in the same pattern on the negative axis starting from the left of -1 and tending to the negative real infinity as L increases. As L becomes large, poles and zeros start distributing around the whole complex plane, however they are always overlapping each other at those position which cancelled out each other from the Pade rational function approximation to f_3.

For $f_4(x)$, we no longer only spot the poles and zeros on the negative real axis anymore. All the poles are distributed evenly in an arc shape on the positive real side on the complex plane while all the zeros distributed on the negative real side on the complex plane similarly in an arc shape. As L increases, the poles and zeros have kept in an arc shape with increasing radius on each side. However, as L becomes large, the arc shape flattens a bit and a semi-oval shape formed for poles and zeros.

In addition, as $f_4(x)$ is complete, it does not carry any branch points or poles on the complex plane. Therefore, all the poles on the complex plane are anomalous and any possible branch cuts formed by the zeros of the approximant could not match to any corresponding branch cuts of the approximated function. As L gets to around 50, the positions of the poles and zeros no longer follow the elliptic pattern and, as we can see, two odd poles appear on the real axis and tend to infinity on both sides.



For $f_5(x)$, we can see a similar arc shape distribution as $f_4(x)$ on the complex plane with a pole at -1 on the negative real axis. However, as L increases, the magnitude of the radius of the arc oscillate between 5 and 10 for both the poles and zeros but not tending to infinity as in $f_4(x)$. As L becomes large, the distribution of the poles start invading the left hand side on the complex plane and similarly the zeros getting onto the right hand side as well although the arc shapes with oscillating radius are still kept for both the poles and zeros at large L. When L is large, more poles appear outside the elliptic shape, however, they are all poles at order 0 which has no effect on the radius of convergence of the approximant in this case so we do not need to worry much. I predict, for large L, the distribution of the poles and zeros will be similar to $f_4(x)$ with a pole at x=-1 always as it is a simple product of a complete function and a function with a simple pole. Apparently, the pole at x=-1 correspond to the pole of the approximated function. As there is no other branch points or poles in the approximated function which means any poles and zeros appear in the approximant would not match with any branch cuts or poles in the approximated function.

For $f_6(x)$, we can see the majority of poles and zeros distributed on the imaginary axis except 2 points. As L increases, the positions of the poles and zeros on the imaginary axis do not change much and stay around $\frac{\sqrt{3}}{2}i$ and $-\frac{\sqrt{3}}{2}i$ on the complex plane. However, for the 2 outstanding points, both of them are on the real axis and with one of them being poles and zeros alternatively as L increases stepwise at the position of -2. For the other outstanding point which is always a pole, it alters its sign with tendency to infinity as L increases.

There are 2 branch points in $f_6(x)$ at $x = -\frac{1}{2} \pm \frac{\sqrt{3}i}{2}$ and a pole at x=∞. We may construct a branch cuts linking these two branch points to the infinity correspondingly and then relate the majority of the poles and zeros to the branch cuts of the approximated function. On the other hand, there is also a pole tending to infinity as L increase which may be correspond to the pole at x=∞.Therefore, the remaining ambiguous point at x=-2 is classified as anomalous.

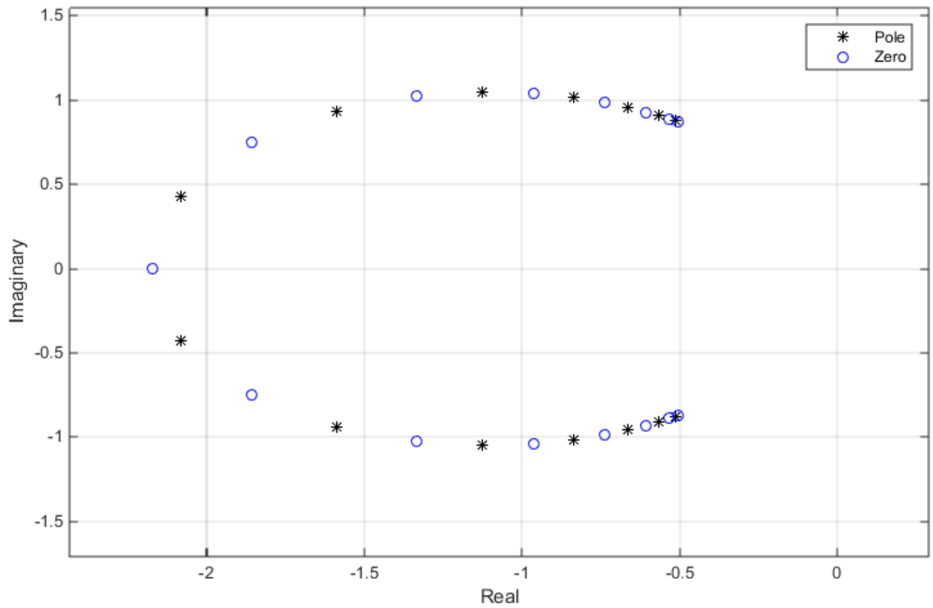


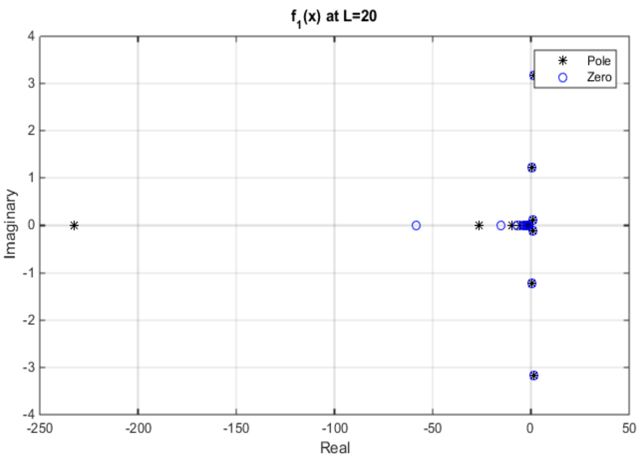
Figure 11: poles and zeros of f_6(x) at L=15

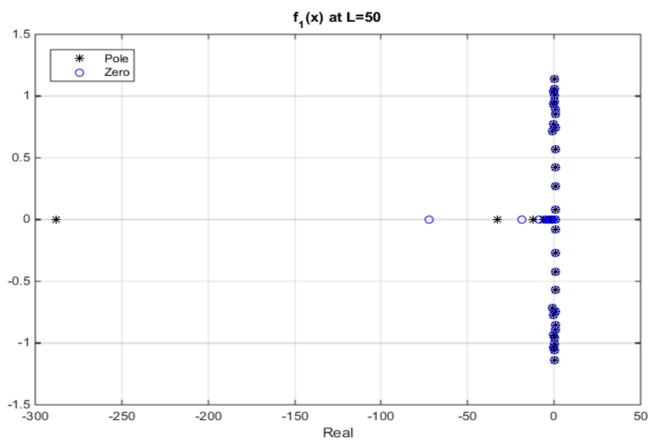
In addition, for L larger than 15, the position of the poles and zeros change completely and it follows another pattern as shown above. We may interpret the above change due to the change of the choice of branch cut. If we join the zeros and poles up, we can spot that it creates a branch cut directly between the two branch points. We will need this result for the second part of this question.

$\mathrm{f}_1(x) = \sqrt{1+x}$

$\mathrm{f}_3(x) = \frac{1}{\sqrt{1+x}}$

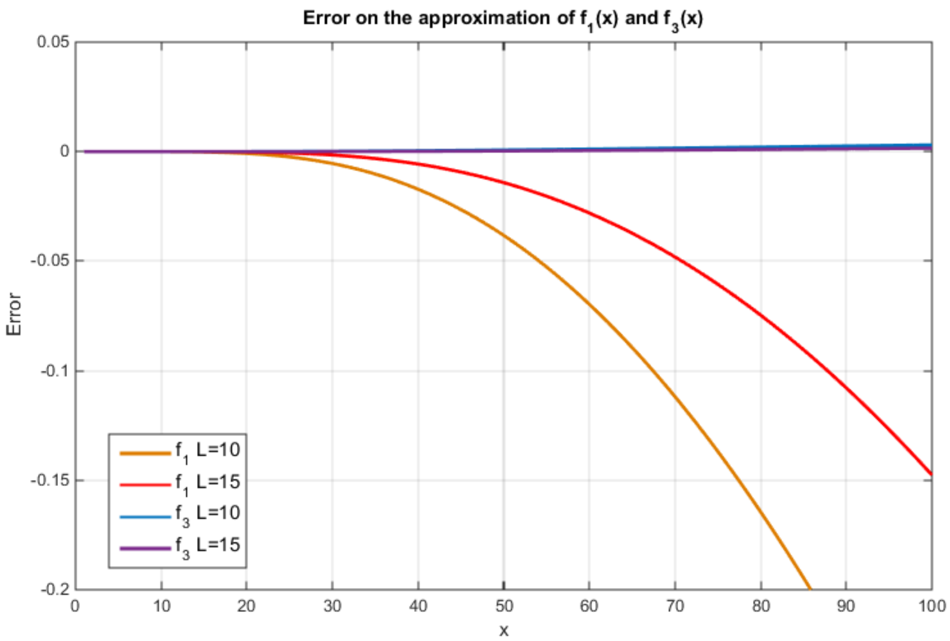
Analytically, we can deduce that there is a branch point at x=-1 and x=∞ in $\mathrm{f}_1(x)$ for which we may introduce a branch cut on the real axis from x=-1 to negative infinity. From the graph, we can see majority of the poles and zeros are lying on the branch cut. As L increases, the position of the poles and zeros tends to infinity with increasing spacing between the poles and zeros and however, the total number of poles and zeros does not increase significantly. On the other hand, at L=15, we can see two outstanding poles and zeros lying on the imaginary axis which are classified as anomalous poles and zeros. For L larger than 15, we can see there are more anomalous poles and zeros appearing outside the branch cut on the imaginary axis. However, the poles and zeros are always appear to be overlapping each other which essentially making the poles become removable singularities/ poles at order 0. This eased the effect of those poles and zeros on the approximant's radius of convergence and also the behaviour of the approximant. (However, from reference [3], it suggests the poles and zeros do not overlap exactly and cancelled out as I think. The existence of these "defect" points slows the rate of convergence on any other points of the complex plane. Further investigation is needed for the result.)





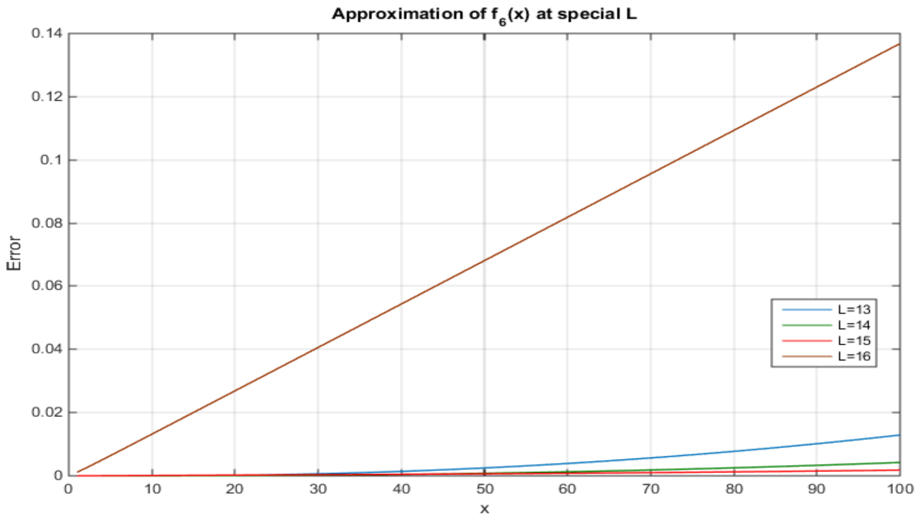
Pade approximation allows us to factor out some of the poles and points on the branch cuts before we do our power series approximation. As there are more overlaps of zeros and poles with the approximated function's branch cuts and poles, the more terms that we can factor out before obtaining the power series approximation. As we are factoring out the singularities, our radius of convergence increase. In most cases we investigated in this project, for larger L, there are more singularities we are able to factor out which means our approximation will be better as the radius of convergence increases. However, in $f_1(x) = \sqrt{1+x}$, the number of the poles and zeros does not increase when L is greater than 15. This limited our approximation accuracy as no further poles and zeros that we can factor out after L=15. Therefore, in question 2, we can see the magnitude of the error remains stable no matter how large L gets after L=15 at x=1.

In $f_3(x)$, we can see there is a pole at x=-1 and two branch points at x=∞ and x=-1 which is similar to our $f_1(x)$. We can construct the same branch cut here and from the graph, we can see $f_3(x)$ behaves exactly the same except the positions of poles and zeros swapped. We could expect $f_3(x)$ shares the same property of the convergence as $f_1(x)$. However, because the closest singularity to $f_3(x)$ is a pole rather than a branch point in $f_1(x)$, we can factor out the singularity(pole) completely for $f_3(x)$. This pushes the radius of convergence to the nearest branch point which means the Pade approximation for $f_3(x)$ diverges less as the approximation for $f_1(x)$.

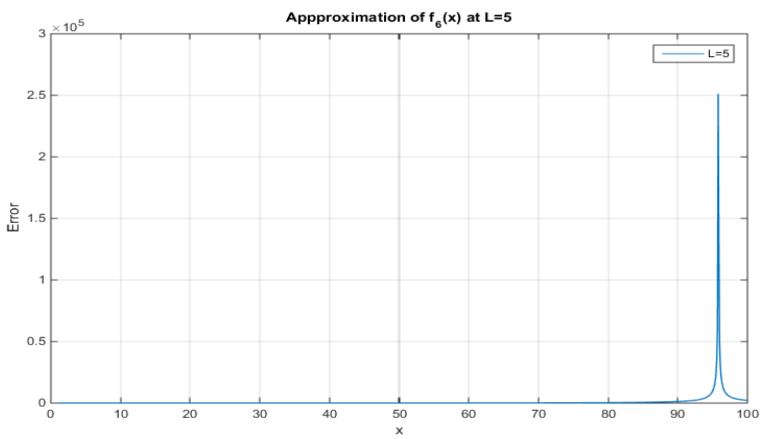


2. Problems encountered when approximating $f_6(x)$?

When I tried to approximate $f_6(x)$ with Pade approximation, as L increases from 6 to 15, the error reduces for the approximation at all x from 1 to 100. However, L greater than 15, the trend changes and the error increases exponentially. This could be because when L becomes greater than 15, the branch cut has suddenly changed which joins the two branch points directly together. This change caused the error diverges as L increases from because all the poles and zeros are no longer on the same branch cuts anymore as before therefore, no factorisation of the poles and zeros can be done by the Pade approximation. Therefore, taking L=15 shall give the most accurate approximation for $f_6(x)$ with the use of Pade Approximation.



In addition, for L=5, the error diverges at exactly x=95, which is due to the existence of pole at x=95 for the Pade approximation function when L=5.



At any poles on the real axis, the corresponding Pade approximation will diverge if we evaluating at that point.

Reference:

[1] Hinch, E.J., Perturbation Methods

[2] Bender, C. & Orszag, S.A., Advanced Mathematical Methods for Scientists and Engineers.

[3] George A. Baker Jr. http://www.scholarpedia.org/article/Pad%C3%A9_approximant, Pade Approximant

[4] M. Vajta., Some Remarks on Pade-Approximation, 3rd Tempus-Intcom Symposium (September 2000)

Program:

1.

```
function Q1(Nstart,Nend)
%sketch graph for Q1 error and N
H=zeros(Nend-Nstart+1,2);
    for x=Nstart:Nend
    A=productinner(x);
    B=sum(A);
    scatter(x,(B-(2)^0.5));
    H(x-Nstart+1,2)=B-(2)^0.5;
    H(x-Nstart+1,1)=x;
    hold on
    end
    %plot(fit(H(:,1),H(:,2),'exp2'))
    hold off
end
```

2.

```
function [H]=Q2padegraphfp(x,Lstart,Lend)
format long
%plotting absolute value of pade
H=zeros(Lend-Lstart+1,2);
    for L=Lstart:1:Lend
        L
    B=padeinner(L,L,x)
    scatter(L,((B-(x+1)^0.5)));
    H(L-Lstart+1,2)=(B-(x+1)^0.5);
    H(L-Lstart+1,1)=L;
    hold on
    end
    hold off
    plot(H(:,1),H(:,2));
end
function [R]=padeinner(L,M,x)
%1st Programming task for Proj 7.5
%Solve equations (4) and (5) given the coefficients c_k
%and values of L and M. Also evaluate the resulting Pade approximant
%for set of values x
format long
%remark: c_0 at first place
C=productinner(1000);
C=C';
A_1=zeros(M, M);
C_back=zeros(M,1);
for a=(L+2):(L+M+1),
    C_back(a-L-1)=C(1,a);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%
for r=(L+1):(L+M)
    min_rm=min(r, M);
    for k=1:min_rm,
        A_1((r-L),k)=C(1,r-k+1);
    end
end
Q=-A_1\C_back;

P=zeros(1,L+1);
P(1,1)=C(1);
for k=1:L
    T=0;
    min_km=min(k,M);
```

```
    for s=1:min_km,
        T=T+Q(s)*C(k-s+1);
    end
    P(1,k+1)=C(k+1)+T;
end
P=P';
X_M=zeros(M,1);
for r=1:M,
    X_M(r,1)=x^r;
end
X_L=zeros(L+1,1);
for r=0:L,
    X_L(r+1,1)=x^r;
end
numer=P'*X_L;
denum=1+Q'*X_M;
R=numer/denum;
end
function [A]=productinner(N)
format long
A=zeros(N+1,1);
A(1,1)=1;
A(2,1)=0.5;
%generating the coefficient of the power series of (1+x)^0.5 in a
column
%vector
for k=2:N,
    A(k+1,1)=((A(k,1))/2)*(-1)*(2*(k-1)-1)/(k);
end
end
```

3.

```
function [Dy]=Q2iterative(L,M,n)
C=productinner(100);
C=C';
T=padeqcoefficientinner(L,M);
for i=1:n

A_1=zeros(M, M);
C_back=zeros(M,1);
for a=(L+2):(L+M+1),
    C_back(a-L-1)=C(1,a);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%
for r=(L+1):(L+M)
    min_rm=min(r, M);
    for k=1:min_rm,
        A_1((r-L),k)=C(1,r-k+1);
    end
end
Q=-A_1\C_back;
Dy=A_1\(-C_back-A_1*T);
T=T+Dy;
end
end
function [Q]=padeqcoefficientinner(L,M)
%Solve equations (4) and (5) given the coefficients c_k
%and values of L and M. Also evaluate the resulting Pade approximant
%for set of values x
format long
%remark: c_0 at first place
```

```
C=productinner(100);
C=C';
A_1=zeros(M, M);
C_back=zeros(M,1);
for a=(L+2):(L+M+1),
    C_back(a-L-1)=C(1,a);
end
C_back;
%%%%%%%%%%%%%%%%%%%%%%%%%
for r=(L+1):(L+M)
    min_rm=min(r, M);
    for k=1:min_rm,
        A_1((r-L),k)=C(1,r-k+1);
    end
end
A_1;
Q=-A_1\C_back;
P=zeros(1,L+1);
P(1,1)=C(1);
for k=1:L
    T=0;
    min_km=min(k,M);
    for s=1:min_km,
        T=T+Q(s)*C(k-s+1);
    end
    P(1,k+1)=C(k+1)+T;
end
P=P';
end
function [A]=productinner(N)
format long
A=zeros(N+1,1);
A(1,1)=1;
A(2,1)=0.5;
%generating the coefficient of the power series of (1+x)^0.5 in a
column
%vector
for k=2:N,
    A(k+1,1)=((A(k,1))/2)*(-1)*(2*(k-1)-1)/(k);
end
end
```

4.

```
function [Q]=powergraph(xstart,xend,N)
format long
%plotting the error change for power series and actual answer
A=productinner(N);
B=flip(A);
B=B';
syms t
syms y
for x=xstart:0.1:xend,
    Q(int16((x-xstart)/0.1+1),1)=x;
    I=poly2sym(B,y);
    Q(int16((x-xstart)/0.1+1),2)=(subs(I,y,x)-(1+x)^0.5);
end
Q
ezplot((poly2sym(B,t)-(1+t)^0.5),[xstart,xend]);

end
function [A]=productinner(N)
```

```
format long
A=zeros(N+1,1);
A(1,1)=1;
A(2,1)=0.5;
%generating the coefficient of the power series of (1+x)^0.5 in a
column
%vector
for k=2:N,
    A(k+1,1)=((A(k,1))/2)*(-1)*(2*(k-1)-1)/(k);
end
end
```

5.

```
function [Q]=Q3padegraph(xstart,xend,M,L)
Q=zeros(10*(xend-xstart)+1,2);
for x=xstart:0.1:xend,
    Q(int16((x-xstart)/0.1+1),1)=x;
    Q(int16((x-xstart)/0.1+1),2)=(padeinner(L,M,x)-(1+x)^0.5);
end
plot(Q(:,1),Q(:,2));
%plotting error between pade approx and actual curve
end
function [R]=padeinner(L,M,x)
%1st Programming task for Proj 7.5
%Solve equations (4) and (5) given the coefficients c_k
%and values of L and M. Also evaluate the resulting Pade approximant
%for set of values x
format long
%remark: c_0 at first place
C=productinner(1000);
C=C';
A_1=zeros(M, M);
C_back=zeros(M,1);
for a=(L+2):(L+M+1),
    C_back(a-L-1)=C(1,a);
end
C_back;
%%%%%%%%%%%%%%%%%%%%%%%%%%

for r=(L+1):(L+M)
    min_rm=min(r, M);
    for k=1:min_rm,
        A_1((r-L),k)=C(1,r-k+1);
    end
end
Q=-A_1\C_back;

P=zeros(1,L+1);
P(1,1)=C(1);
for k=1:L
    T=0;
    min_km=min(k,M);
    for s=1:min_km,
        T=T+Q(s)*C(k-s+1);
    end
    P(1,k+1)=C(k+1)+T;
end
P=P';

X_M=zeros(M,1);
for r=1:M,
```

```
    X_M(r,1)=x^r;
end
X_L=zeros(L+1,1);
for r=0:L,
    X_L(r+1,1)=x^r;
end
numer=P'*X_L;
denum=1+Q'*X_M;
R=numer/denum;
end

function [A]=productinner(N)
format long
A=zeros(N+1,1);
A(1,1)=1;
A(2,1)=0.5;
%generating the coefficient of the power series of (1+x)^0.5 in a
column
%vector
for k=2:N,
    A(k+1,1)=((A(k,1))/2)*(-1)*(2*(k-1)-1)/(k);
end
end
```

6.

```
function [Q]=Q4padegraph(xstart,xend,M,L)
Q=zeros(10*(xend-xstart)+1,2);
for x=xstart:0.1:xend,
    Q(int16((x-xstart)/0.1+1),1)=x;
    Q(int16((x-xstart)/0.1+1),2)=((Q4padeinner(L,M,x)));
end
plot(Q(:,1),Q(:,2));

%plotting error between pade approx and actual curve
end

function [A]=Q4productinner(N)

A=zeros(N+1,1);

for k=0:N,
    P=factorial(k);
    A(k+1,1)=((-1)^k)*P;
end
end

function [R]=Q4padeinner(L,M,x)
%Q4 Programming task for Proj 7.5
%Solve equations (4) and (5) given the coefficients c_k
%and values of L and M. Also evaluate the resulting Pade approximant
%for set of values x
format long
%remark: c_0 at first place
C=Q4productinner(100);
C=C';
A_1=zeros(M, M);
C_back=zeros(M,1);
for a=(L+2):(L+M+1),
    C_back(a-L-1)=C(1,a);
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%

for r=(L+1):(L+M)
    min_rm=min(r, M);
    for k=1:min_rm,
        A_1((r-L),k)=C(1,r-k+1);
    end
end
Q=-A_1\C_back;

P=zeros(1,L+1);
P(1,1)=C(1);
for k=1:L
    T=0;
    min_km=min(k,M);
    for s=1:min_km,
        T=T+Q(s)*C(k-s+1);
    end
    P(1,k+1)=C(k+1)+T;
end
P=P';

X_M=zeros(M,1);
for r=1:M,
    X_M(r,1)=x^r;
end
X_L=zeros(L+1,1);
for r=0:L,
    X_L(r+1,1)=x^r;
end
numer=P'*X_L;
denum=1+Q'*X_M;
R=numer/denum;
end
```

7.

```
function [Q]=Q4powergraph(xstart,xend,N)
format long
%plotting for power series
A=Q4productinner(N);
B=flip(A);
B=B';
syms t
syms y
for x=xstart:0.1:xend,
    Q(int16((x-xstart)/0.1+1),1)=x;
    I=poly2sym(B,y);
    Q(int16((x-xstart)/0.1+1),2)=(subs(I,y,x));
end
ezplot((poly2sym(B,t)),[xstart,xend]);
end
```

8.

```
function [P]=programB(COE)
%COE be the coefficient of the polynomial starting from the highest
order
%of the polynomial

P=roots(COE);
end
```

9.

```
function [PO,ZE]=Q5f_1(L,M)
%Solve equations (4) and (5) given the coefficients c_k
%and values of L and M. Also evaluate the resulting Pade approximant
%for set of values x
format long
%remark: c_0 at first place
C=productinner(100);
C=C';
A_1=zeros(M, M);
C_back=zeros(M,1);
for a=(L+2):(L+M+1),
    C_back(a-L-1)=C(1,a);
end
C_back;
%%%%%%%%%%%%%%%%%%%%%%%%%%%

for r=(L+1):(L+M)
    min_rm=min(r, M);
    for k=1:min_rm,
        A_1((r-L),k)=C(1,r-k+1);
    end
end
A_1;
Q=-A_1\C_back;

P=zeros(1,L+1);
P(1,1)=C(1);
for k=1:L
    T=0;
    min_km=min(k,M);
    for s=1:min_km,
        T=T+Q(s)*C(k-s+1);
    end
    P(1,k+1)=C(k+1)+T;
end
P=flip(P');
ZE=programB(P);
Q=[1;Q];
Q=flip(Q');
PO=programB(Q);
disp('Poles occur at ');
PO
disp('Zeros occur at ');
ZE
p=plot(real(PO),imag(PO),'g*', real(ZE),imag(ZE),'bo');
p(1).MarkerEdgeColor='k';
p(1).MarkerFaceColor='k';
p(2).MarkerEdgeColor='b';

xlabel('Real') % x-axis label
ylabel('Imaginary') % y-axis label
legend('Pole','Zero')
end
function [A]=productinner(N)
format long
A=zeros(N+1,1);
A(1,1)=1;
A(2,1)=0.5;
```

```
%generating the coefficient of the power series of (1+x)^0.5 in a
column
%vector
for k=2:N,
    A(k+1,1)=((A(k,1))/2)*(-1)*(2*(k-1)-1)/(k);
end
end
```

10.

```
function [PO,ZE]=Q5f_3(L,M)
%Solve equations (4) and (5) given the coefficients c_k
%and values of L and M. Also evaluate the resulting Pade approximant
%for set of values x
format long
%remark: c_0 at first place
C=f_3productinner(100);
C=C'
A_1=zeros(M, M);
C_back=zeros(M,1);
for a=(L+2):(L+M+1),
    C_back(a-L-1)=C(1,a);
end
C_back;
%%%%%%%%%%%%%%%%%%%%%%%%%%%

for r=(L+1):(L+M)
    min_rm=min(r, M);
    for k=1:min_rm,
        A_1((r-L),k)=C(1,r-k+1);
    end
end
A_1;
Q=-A_1\C_back;

P=zeros(1,L+1);
P(1,1)=C(1);
for k=1:L
    T=0;
    min_km=min(k,M);
    for s=1:min_km,
        T=T+Q(s)*C(k-s+1);
    end
    P(1,k+1)=C(k+1)+T;
end
P=flip(P');
ZE=programB(P);
Q=[1;Q];
Q=flip(Q');
PO=programB(Q);
disp('Poles occur at ');
PO
disp('Zeros occur at ');
ZE
p=plot(real(PO),imag(PO),'g*', real(ZE),imag(ZE),'bo');
p(1).MarkerEdgeColor='k';
p(1).MarkerFaceColor='k';
p(2).MarkerEdgeColor='b';
xlabel('Real') % x-axis label
ylabel('Imaginary') % y-axis label
legend('Pole','Zero')
end
```

```
function [A]=f_3productinner(N)
A=zeros(N+1,1);
A(1,1)=1;
for k=1:N,
    A(k+1,1)=A(k,1)*(-1)*(1/k)*(0.5)*(2*(k-1)+1);
end
end
```

11.

```
function [PO,ZE]=Q5f_4(L,M)
%Solve equations (4) and (5) given the coefficients c_k
%and values of L and M. Also evaluate the resulting Pade approximant
%for set of values x
format long
%remark: c_0 at first place
C=f_4productinner(100);
C=C'
A_1=zeros(M, M);
C_back=zeros(M,1);
for a=(L+2):(L+M+1),
    C_back(a-L-1)=C(1,a);
end
C_back;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for r=(L+1):(L+M)
    min_rm=min(r, M);
    for k=1:min_rm,
        A_1((r-L),k)=C(1,r-k+1);
    end
end
A_1;
Q=-A_1\C_back;

P=zeros(1,L+1);
P(1,1)=C(1);
for k=1:L
    T=0;
    min_km=min(k,M);
    for s=1:min_km,
        T=T+Q(s)*C(k-s+1);
    end
    P(1,k+1)=C(k+1)+T;
end
P=flip(P');
ZE=programB(P);
Q=[1;Q];
Q=flip(Q');
PO=programB(Q);
disp('Poles occur at ');
PO
disp('Zeros occur at ');
ZE
p=plot(real(PO),imag(PO),'g*', real(ZE),imag(ZE),'bo');
p(1).MarkerEdgeColor='k';
p(1).MarkerFaceColor='k';
p(2).MarkerEdgeColor='b';
```

```
xlabel('Real') % x-axis label
ylabel('Imaginary') % y-axis label
legend('Pole','Zero')
end
function [A]=f_4productinner(N)
A=zeros(N+1,1);
A(1,1)=1;
for k=1:N,
    A(k+1,1)=A(k,1)/k;
end
end
```

12.

```
function [C]=Q5f_5(L,M)
%Solve equations (4) and (5) given the coefficients c_k
%and values of L and M. Also evaluate the resulting Pade approximant
%for set of values x
format long
%remark: c_0 at first place
C=f_5productinner;
C=C';
A_1=zeros(M, M);
C_back=zeros(M,1);
for a=(L+2):(L+M+1),
    C_back(a-L-1)=C(1,a);
end
C_back;
%%%%%%%%%%%%%%%%%%%%%%%%%%%

for r=(L+1):(L+M)
    min_rm=min(r, M);
    for k=1:min_rm,
        A_1((r-L),k)=C(1,r-k+1);
    end
end
A_1;
Q=-A_1\C_back;

P=zeros(1,L+1);
P(1,1)=C(1);
for k=1:L
    T=0;
    min_km=min(k,M);
    for s=1:min_km,
        T=T+Q(s)*C(k-s+1);
    end
    P(1,k+1)=C(k+1)+T;
end
P=flip(P');
ZE=programB(P);
Q=[1;Q];
Q=flip(Q');
PO=programB(Q);
disp('Poles occur at ');
PO
disp('Zeros occur at ');
ZE
p=plot(real(PO),imag(PO),'g*', real(ZE),imag(ZE),'bo');
p(1).MarkerEdgeColor='k';
p(1).MarkerFaceColor='k';
p(2).MarkerEdgeColor='b';
xlabel('Real') % x-axis label
```

```
ylabel('Imaginary') % y-axis label
legend('Pole','Zero')
end

function [C]=f_5productinner
A=zeros(101,1);
B=zeros(101,1);
C=zeros(101,1);
A(1,1)=1;
B(1,1)=1;
for k=1:100,
    A(k+1,1)=A(k,1)/k;
    B(k+1,1)=-1*B(k,1);
end
for i=1:101,
    sum=0;
    for j=1:i,
        sum=sum+A(j)*B(i-j+1);
    end
    C(i,1)=sum;
end
end
```

13.

```
function []=Q5f_6(L,M)
%Solve equations (4) and (5) given the coefficients c_k
%and values of L and M. Also evaluate the resulting Pade approximant
%for set of values x
format long
%remark: c_0 at first place
C=f_6productinner;
C=C';
A_1=zeros(M, M);
C_back=zeros(M,1);
for a=(L+2):(L+M+1),
    C_back(a-L-1)=C(1,a);
end
C_back;
%%%%%%%%%%%%%%%%%%%%%%%%%%%

for r=(L+1):(L+M)
    min_rm=min(r, M);
    for k=1:min_rm,
        A_1((r-L),k)=C(1,r-k+1);
    end
end
A_1;
Q=-A_1\C_back;

P=zeros(1,L+1);
P(1,1)=C(1);
for k=1:L
    T=0;
    min_km=min(k,M);
    for s=1:min_km,
        T=T+Q(s)*C(k-s+1);
    end
    P(1,k+1)=C(k+1)+T;
end
P=flip(P');
```

```
ZE=programB(P);
Q=[1;Q];
Q=flip(Q');
PO=programB(Q);
disp('Poles occur at ');
PO
disp('Zeros occur at ');
ZE
p=plot(real(PO),imag(PO),'g*', real(ZE),imag(ZE),'bo');
p(1).MarkerEdgeColor='k';
p(1).MarkerFaceColor='k';
%p(2).MarkerEdgeColor='b';
xlabel('Real') % x-axis label
ylabel('Imaginary') % y-axis label
legend('Pole','Zero')
end

function [C]=f_6productinner
A=zeros(101,1);
B=zeros(101,1);
C=zeros(101,1);
a_1=complex(-0.5,((3^0.5)/2));
a_2=complex(-0.5,-((3^0.5)/2));
A(1,1)=1;
A(2,1)=-0.5/a_1;
B(1,1)=1;
B(2,1)=-0.5/a_2;
for k=2:100,
    A(k+1,1)=((2*k-3)/(2*a_1))*A(k,1)/k;
    B(k+1,1)=((2*k-3)/(2*a_2))*B(k,1)/k;
end
for i=1:101,
    sum=0;
    for j=1:i,
        sum=sum+A(j)*B(i-j+1);
    end
    C(i,1)=sum;
end
end
```

14.

```
function [Q]=Q5f_6padegraph(xstart,xend,M,L)
C=f_6productinner';
Q=zeros(10*(xend-xstart)+1,2);
for x=xstart:0.1:xend,
    Q(int16((x-xstart)/0.1+1),1)=x;
    Q(int16((x-xstart)/0.1+1),2)=(abs(Q5f_6padeinner(L,M,x)-
(1+x+x^2)^0.5));
end
plot(Q(:,1),Q(:,2));
%plotting error between pade approx and actual curve
end

function [R]=Q5f_6padeinner(L,M,x)
%1st Programming task for Proj 7.5
%Solve equations (4) and (5) given the coefficients c_k
%and values of L and M. Also evaluate the resulting Pade approximant
```

```
%for set of values x
format long
%remark: c_0 at first place
C=f_6productinner;
C=C';
A_1=zeros(M, M);
C_back=zeros(M,1);
for a=(L+2):(L+M+1),
    C_back(a-L-1)=C(1,a);
end
C_back;
%%%%%%%%%%%%%%%%%%%%%%%%%%%

for r=(L+1):(L+M)
    min_rm=min(r, M);
    for k=1:min_rm,
        A_1((r-L),k)=C(1,r-k+1);
    end
end
Q=-A_1\C_back;

P=zeros(1,L+1);
P(1,1)=C(1);
for k=1:L
    T=0;
    min_km=min(k,M);
    for s=1:min_km,
        T=T+Q(s)*C(k-s+1);
    end
    P(1,k+1)=C(k+1)+T;
end
P=P';

X_M=zeros(M,1);
for r=1:M,
    X_M(r,1)=x^r;
end
X_M;
X_L=zeros(L+1,1);
for r=0:L,
    X_L(r+1,1)=x^r;
end
X_L;
P;
Q;
numer=P'*X_L;
denum=1+Q'*X_M;
R=numer/denum;
end
function [C]=f_6productinner
A=zeros(101,1);
B=zeros(101,1);
C=zeros(101,1);
a_1=complex(-0.5,((3^0.5)/2));
a_2=complex(-0.5,-((3^0.5)/2));
A(1,1)=1;
A(2,1)=-0.5/a_1;
B(1,1)=1;
B(2,1)=-0.5/a_2;
for k=2:100,
    A(k+1,1)=((2*k-3)/(2*a_1))*A(k,1)/k;
```

```
    B(k+1,1)=((2*k-3)/(2*a_2))*B(k,1)/k;
end
for i=1:101,
    sum=0;
    for j=1:i,
        sum=sum+A(j)*B(i-j+1);
    end
    C(i,1)=sum;
end
end
```