Problem 1 figure:

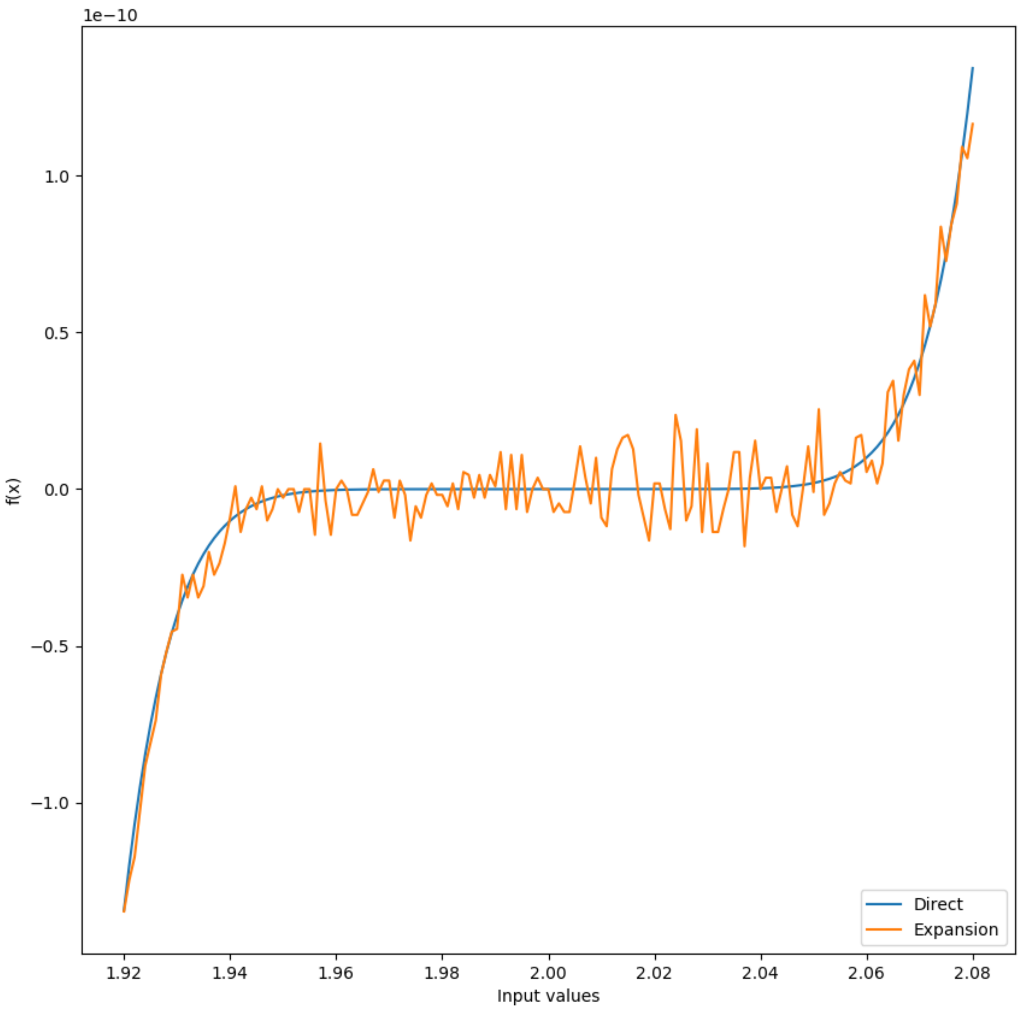

fig: Values for the direct evaluation and expansion of $(x - 2)^9$

Problem 2 figure:

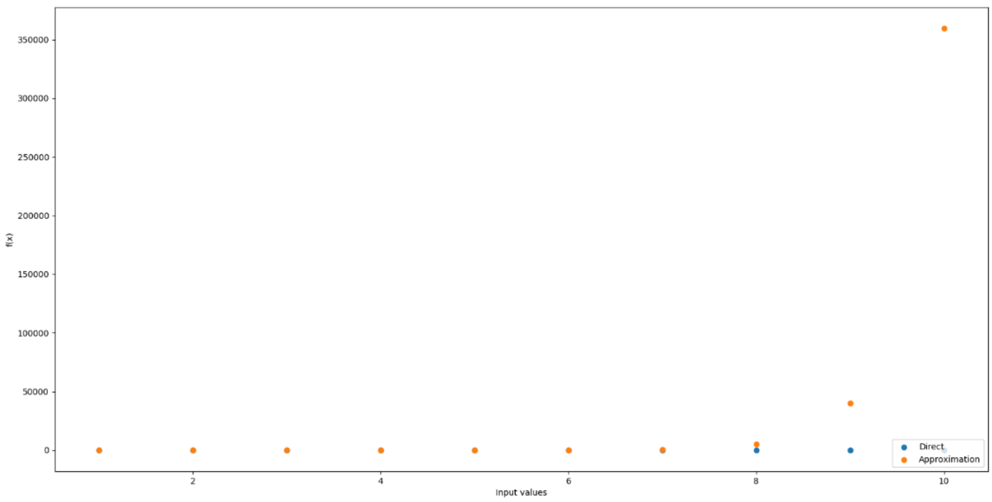

fig: Scatter plot comparing the factorial function and its approximation

Problem 3 (a):

From my understanding Relative Error would be a better measure for polynomial evaluation. We might have instances where our estimate and the true value are very small.

In this case absolute error may be less than the machine precision; giving an inaccurate value due to representation. Now if the true value is also small, we can express relative error as (x_estimate/x_true) - 1, which is numerically stable.

Problem 3 - part (b)

Absolute errors from Problem 2:	Relative errors from Problem 2:
-1.0	-1.0
-1.0778629911042108	-0.5389314955521054
-1.0809956485110168	-0.3603318828370056
1.836209591345864	0.459052397836466
18.506175132893294	3.701235026578659
112.0191679575901	18.669861326265018
703.078184642185	100.43974066316927
4972.395831612462	621.5494789515577
39893.39545265671	4432.599494739635
359526.87284194835	35952.687284194835

It is clear from the calculated values that both relative and absolute errors increases.

Problem 4 (a):

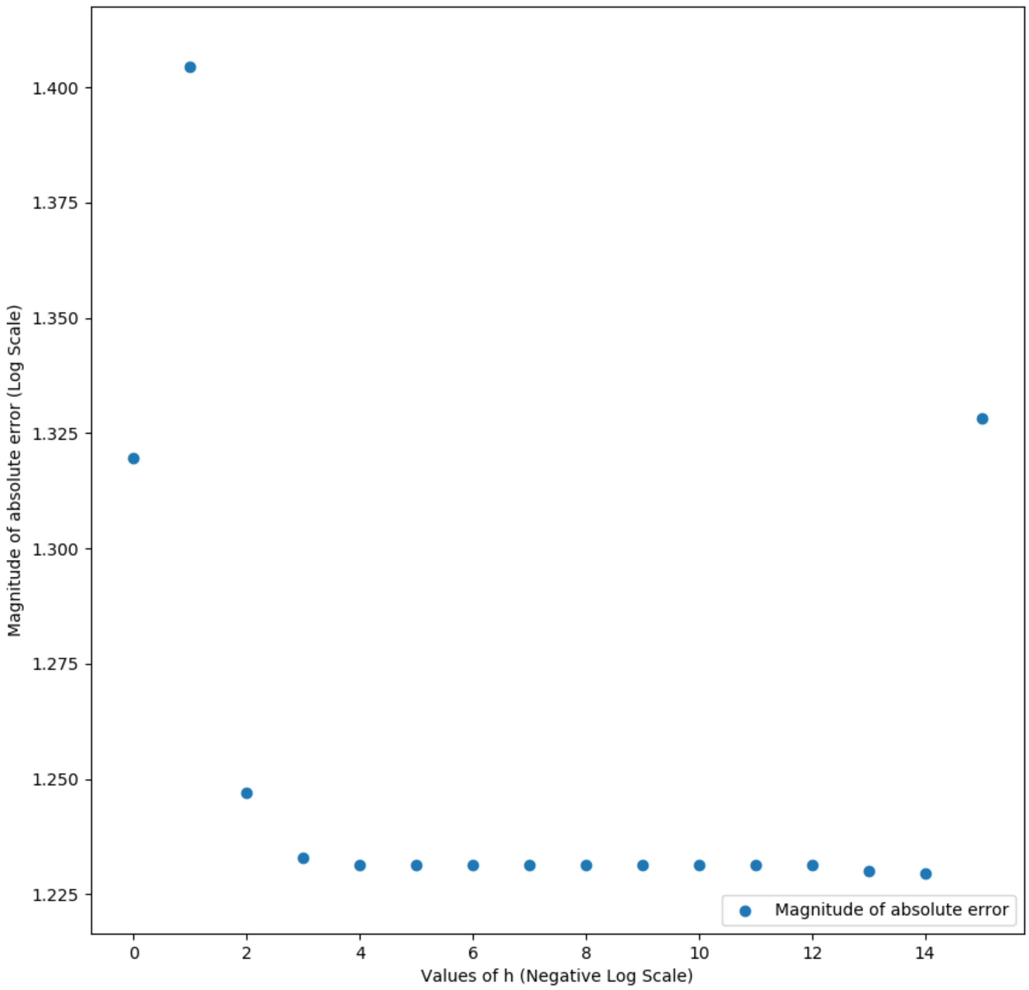

Fig: log scaled magnitudes of absolute errors

Yes, there is a minimum value for the magnitude of error near -log(h) = 14.

Machine precision for floating point values on 64-bit machine is 2.220446049250313e-16 (approx.) and the square root is 1.4901161193847656e-08 (approx.).

h corresponding to lowest magnitude of absolute error is h = 1e-14.

Problem 4 (b):

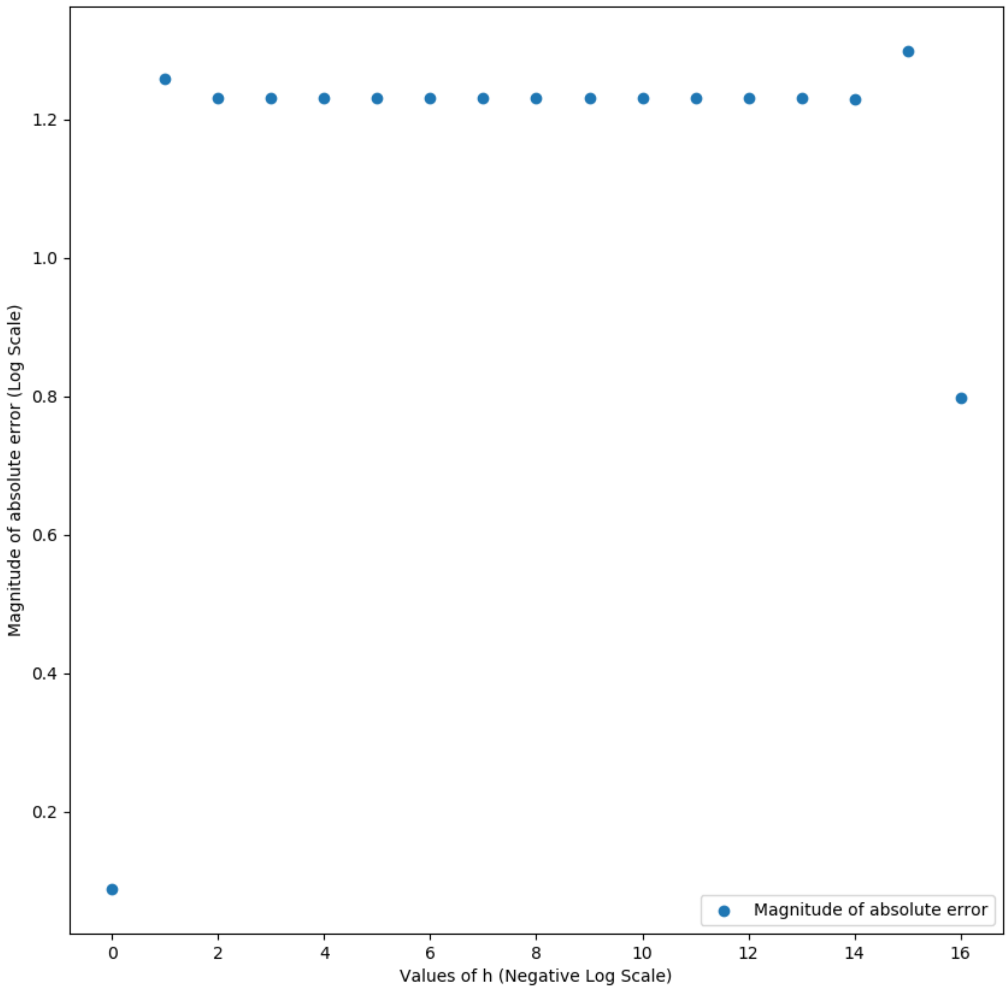

Fig: log scaled magnitudes of absolute errors

Yes, there is a minimum value for the magnitude of error near -log(h) = 0.

Machine precision for floating point values on 64-bit machine is 2.220446049250313e-16 (approx.) and the square root is 1.4901161193847656e-08 (approx.).

h corresponding to lowest magnitude of absolute error is h = 1e-0.