2. e^x

$f(x) = e^x = 1 + x + x^2/2! + x^3/3! + \dots$

We are given a mantissa of 5 digits.

a. It was quite painful, but after rounding each operation at 5 digits, I find that

$e^{5.5} \approx 244.71$

After fewer than 30 terms, the series stops increasing our accuracy because all terms are smaller than 10^{-2}, and we only have room for 5 digits in the mantissa. I've attached a page at the end with these terms.

b. The partial sums are:

k	S_k	"	" "
0	1.0000	12	243.62
1	6.5000	13	244.30
2	21.625	14	244.57
3	49.355	15	244.67
4	87.484	16	244.70
5	129.43	17	244.71
6	167.88	18	244.71
7	198.09	19	244.71
8	218.86	20	244.71
9	231.55	21	"
10	238.53	22	"
11	242.02	23	"

" " " "

So we see the S_k don't change past 17, so we gain nothing in going beyond this.

To 5 digit precision, the built in exp() function in Mathematica gives:

$e^{5.5} \approx 244.69$

The error is roughly 0.02 or 0.82%

This is quite a large error.

c.

Now let's try out going from right to left.
The partial sums are:

k	S_k	k	S_k
0	1.0000	11	242.02
1	6.5000	12	243.62
2	21.625	13	244.29
3	49.355	14	244.56
4	87.484	15	244.66
5	129.43	16	244.69
6	167.88	17	244.69
7	198.09	18	"
8	218.86	19	"
9	231.55	20	"
10	238.53	21	"

We see that, rounded to 5 digits, summing back-to-front has produced no more than an eps of error. We still converge approximately at the same term (16, rather than 17)!

~~at the~~ relative error is only:

$$\left| \frac{e^{5.5} - \text{Machine}(e^{5.5})}{e^{5.5}} \right| = 0.00079\%$$

Much better.

d)

(i)

k	S_k	k	S_k	k	S_k
1	-4.5000	8	8.0890	15	-0.021394
2	10.625	9	-4.6030	16	0.012116
3	-17.105	10	2.3775	17	0.001274
4	21.024	11	-1.1127	18	0.0045868
5	-20.918	12	0.4870	19	0.0036278
6	17.529	13	-0.18979	20	0.0038915
7	-12.679	14	0.07609	21	0.0038224

(Next page)

K	S_K
22	0.0038397
23	0.0038356
24	0.0038366
25	0.0038364
26	0.0038364
27	0.0038364
28	0.0038364
29	0.0038364
30	0.0038364

We see that not only does this series converge very slowly, but it is wrong by about

$$\left|\frac{e^{-5.5} - M(e^{-5.5})}{e^{-5.5}}\right| = 6.13\%$$

(ii)

K	S_K				
1	-4.5000	11	-1.1140	21	0.0040000
2	10.625	12	0.048700	22	0.0040000
3	-17.105	13	-0.19000	23	"
4	21.024	14	0.076000	24	
5	-20.918	15	-0.021000	25	
6	17.529	16	0.012000	26	
7	-12.679	17	0.0010000	27	
8	8.0890	18	0.0050000	28	
9	-4.6030	19	0.0040000	29	
10	2.3780	20	0.0030000	30	

For this we severely lose precision. This is because of subtraction of many close numbers, and our largest numbers having only 10^{-3} accuracy in the sum - the same magnitude as our answer. This is a horrible way to compute this.

(iii)

This should go a bit better.

K	S_K		
1	-4.5000	11	-1.1100
2	10.625	12	0.49000
3	-17.105	13	-0.19000
4	21.024	14	0.080000
5	-20.918	15	-0.020000
6	17.529	16	0.010000
7	-12.679	17	0.000000
8	8.0900		"
9	-4.60000		"
10	2.38		

We lose all digits this way, because the sum of the terms in each side becomes large compared to 10^{-3}.

(iv)

K	S_K				
1	-4.5000	9	-4.5900	17	0.00000
2	10.625	10	2.3900		"
3	-17.105	11	-61000		
4	21.024	12	0.49000		
5	-20.918	13	-0.19000		
6	17.529	14	0.08000		
7	-12.679	15	-0.02000		
8	8.1000	16	0.01000		

The 3rd and 4th methods converge fastest, but are horrible. The lowest error was slowest to converge (i).

(e) To compute e^{-x} for $x > 0$, it could be better to compute e^{x} first, right to left, and then divide 1.0000 by e^x. In this case this yields:

$$\frac{1.0000}{244.69} = 0.0040868$$

The error on this is just 0.00070%
This seems acceptable and is of order eps.