1. Output comparation:

SLATEC arguments	values from SLATEC	values from my implementation
1.0 1.0 1e-10 0	1.435683098, 0.1351132291	1.4356830978, 0.1351135349
1.5 .5 1e-10 0	1.435683098, 0.1351132291	1.4356830898, 0.1351132564
1.4 .1 1e-10 0	1.435683098, 0.1351132291	1.4356830176, 0.1351131089
1.44 .14 1e-10 0	1.435683098, 0.1351132291	1.4356830888, 0.1351132060

From outputs we can see that SLATEC implementation is more stable when changing the initial guess. In all tested cases both agree on at least 6 first decimal places.

To compare performance, I edited both programs to run the interesting routine 1 000 000 times and printed the time the whole program took to run with "time" unix command. For SLATEC implementation I also changed the surrounding lines so that x-vector gets reset every time. After these changes the outputs of time were

```
real  0m3.210s
user  0m3.208s
sys   0m0.000s
```

for SLATEC and

```
real  0m3.015s
user  0m2.728s
sys   0m0.284s
```

for my implementation. There real is the wall clock time the user experiences, user is time spent in user-mode within process and sys is time spend in kernel mode.

Small differences can be seen between consecutive runs but nothing too big. My implementation is slightly faster but this difference is not very big.

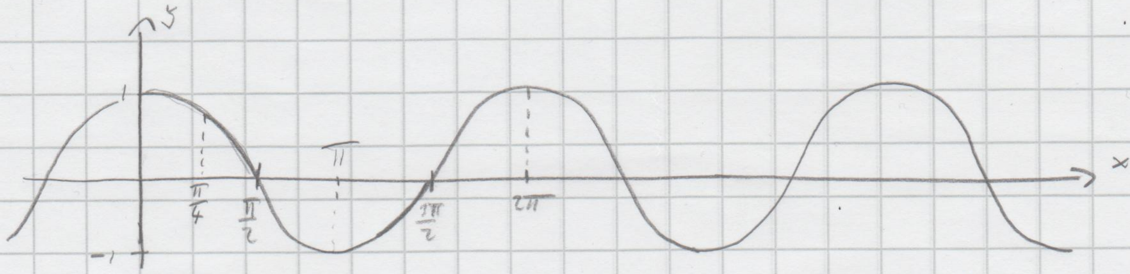
(2) (a) We'll have to find k and r so that $x = k\ln 2 + r$ and $|r| < \frac{1}{2}\log 2$ ie. $k = \text{round}\left(\frac{x}{\ln 2}\right)$ and $r = x - \text{round}\left(\frac{x}{\ln 2}\right)\cdot \ln 2$

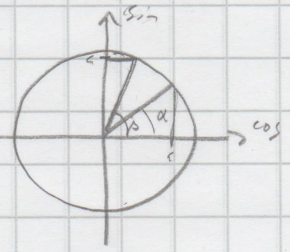
Then

$$e^x = e^{k\ln 2 + r}$$
$$= e^{k\ln 2} e^r$$
$$= 2^k e^r \quad \text{where} \quad \begin{cases} k = \text{round}\left(\frac{x}{\ln 2}\right) \\ r = x - \text{round}\left(\frac{x}{\ln 2}\right)\ln 2 \end{cases}$$

For own myexp we can set $x = k\ln 2 + r'\ln 2$ where k is same as earlierly but $r' = r/\ln 2$ and therefore $e^x = 2^k 2^{r'}$.

2 b



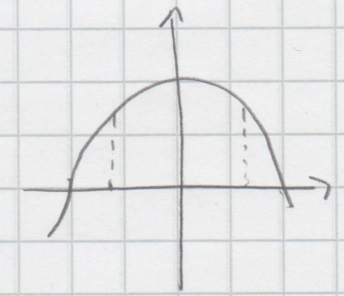


First we can "move" all x values to range $[0, 2\pi]$ by setting
$x' = |x - \lfloor \frac{x}{2\pi} \rfloor \cdot 2\pi|$

$\alpha = \frac{\pi}{2} - \beta$

$\sin \alpha = \cos \beta$

Now it is fairly easy to "mirror" the function to be defined in four parts:



$$\cos(x') = \begin{cases} \cos(x') & , \ 0 < x' \leq \frac{\pi}{2} \\ -\cos(-(x'-\pi)) & , \ \frac{\pi}{2} \leq x' < \pi \\ -\cos(x'-\pi) & , \ \pi \leq x < \frac{3\pi}{2} \\ \cos(-(x'-2\pi)) & , \ \frac{3\pi}{2} \leq x \leq 2\pi \end{cases}$$

$$= \begin{cases} \cos(x') \\ -\cos(\pi - x') \\ -\cos(x'-\pi) \\ \cos(2\pi - x') \end{cases}$$

Now we have reduced $\cos x$ to $[0, \frac{\pi}{2}]$. Now we have to reduce it further to $[0, \frac{\pi}{4}]$. Lets use the knowledge $\sin(\frac{\pi}{2} - x) = \cos x$ when $x \in [\frac{\pi}{4}, \frac{\pi}{2}]$ and $\sin x = \sqrt{1-\cos^2 x}$. Therefore
when $x' \in [0, \frac{\pi}{4}]$ $\cos(x') = \cos(x')$
when $x' \in [\frac{\pi}{4}, \frac{\pi}{2}]$ $\cos(x') = \sqrt{1-\cos^2(\frac{\pi}{2} - x')}$

Now combining this with earlier form:

$$\cos(x') \begin{cases} \cos(x') & , \ 0 \leq x' < \frac{\pi}{4} \\ \sqrt{1-\cos^2(\frac{\pi}{2} - x')} & , \ \frac{\pi}{4} \leq x' < \frac{\pi}{2} \\ -\sqrt{1-\cos^2(x - \frac{\pi}{2})} & , \ \frac{\pi}{2} \leq x' < \frac{3\pi}{4} \\ -\cos(\pi - x') & , \ \frac{3\pi}{4} \leq x' < \pi \\ -\cos(x'-\pi) & , \ \pi \leq x' < \frac{5\pi}{4} \\ -\sqrt{1-\cos^2(\frac{3\pi}{2} - x')} & , \ \frac{5\pi}{4} \leq x' < \frac{3\pi}{2} \\ \sqrt{1-\cos^2(x - \frac{3\pi}{2})} & , \ \frac{3\pi}{2} \leq x' < \frac{7\pi}{4} \\ \cos(2\pi - x') & , \ \frac{7\pi}{4} \leq x' < 2\pi \end{cases} \quad \text{where } x' = |x - \lfloor \tfrac{x}{2\pi} \rfloor \cdot 2\pi|$$

(4) Let's consider finding a polynomial $P_N(x) = \sum_{i=0}^{N} c_i e^{ix} = \sum_{i=0} c_i (e^x)^i$ which goes through points $(x_1, y_1), (x_2, y_2), \dots$ and (x_N, y_N). It is clear that if we performing change of variables $t = e^x \Rightarrow x = \ln t$. Now

$P_N(t) = \sum_{i=0}^{N} c_i t^i$ and points the polynomial goes through are $(\ln t_1, y_1), (\ln t_2, y_2), (\ln t_3, y_3), \dots$ and $(\ln t_N, y_N)$. We can recognize $P_N(t) = \sum c_i t^i$ as interpolating polynomial for given points $(\ln t_i, y_i)$. It is proven in lecture notes (interpolation, page 5-7) that there is a solution for these c_i and that solution is unique. $\square$