**Problem 1.** Since \(f^{(n)}(x) = n!/(1-x)^{n+1}\) we have the Taylor representation

\[
f(x) = \sum_{m=0}^{n-1} x^m + R_n(x).
\]

Taylor’s integral theorem for the remainder gives us

\[
R_n(x) = \int_0^x f^{(n-1)}(z) (x-z)^{n-1} \, dz
\]

\[
= \frac{n}{x-1} \int_0^x \frac{x-1}{1-z} \frac{1}{1-z} \, dz
\]

\[
= \frac{1}{x-1} \left[ \frac{x-1}{1-z} \right]_{z=0}^{z=x}
\]

\[
= x^n
\]

Another expression is the Cauchy remainder formula \(R_n(x) = \frac{f^{(n)}(\xi)}{n!} x^n = \frac{x^n}{(1-\xi)^{n+1}}\) for some \(\xi\) between 0 & \(x\).

**Problem 2.** (a) Base case:

\[
\left( \begin{array}{c} \alpha \\ 0 \end{array} \right) = \left( \begin{array}{c} \alpha \\ 0 \end{array} \right) = A.
\]

Induction step:

\[
A^n = AA^{n-1} = \left( \begin{array}{c} \alpha \\ 0 \end{array} \right) \left( \begin{array}{c} \alpha \frac{\alpha-\beta}{\alpha+\beta-1} \\ 0 \end{array} \right) = \left( \begin{array}{c} \alpha \frac{\alpha^{n-1}-\beta^{n-1}}{\alpha+\beta-1} + \beta^n \\ 0 \end{array} \right) = \left( \begin{array}{c} \alpha^n \frac{\alpha^{n-1}-\beta^{n-1}}{\alpha+\beta-1} \\ 0 \end{array} \right).
\]

(b) Since \(|\alpha|, |\beta| < 1\), we recall the analytic forms for geometric series from Problem 1 and write:

\[
\sum_{n=0}^{\infty} A^n = \sum_{n=0}^{\infty} \left( \begin{array}{c} \alpha^n \\ 0 \end{array} \right) = \left( \begin{array}{c} \frac{1}{1-\alpha} \\ 0 \end{array} \right) = \left( \begin{array}{c} \frac{1}{1-\alpha} \\ 0 \end{array} \right).
\]

Then we recall Cramer’s rule and with \(\det(I - A) = (1-\alpha)(1-\beta)\) we can write:

\[
I - A = \left( \begin{array}{cc} 1 - \alpha & -\beta \\ 0 & 1 - \beta \end{array} \right), \quad (I - A)^{-1} = \frac{1}{\det(I - A)} \left( \begin{array}{cc} 1 - \beta & \beta \\ 0 & 1 - \alpha \end{array} \right) = \left( \begin{array}{cc} 1 - \alpha & \beta \\ 0 & 1 - \alpha \end{array} \right).
\]

**Problem 3.** We find:

(a) 15852243, (b) 35483.29809570312, and (c) 228.3125. For (d) we recall geometric series and write

\[
(0.13)_{4} = 1 \times 4^{-1} + 3 \times 4^{-2} + 1 \times 4^{-3} + 3 \times 4^{-4} + \cdots = \frac{1}{4} \sum_{k=0}^{\infty} 4^{-2k} + \frac{3}{4^2} \sum_{k=0}^{\infty} 4^{-2k}
\]

\[
= \left( 1 + \frac{3}{16} \right) \frac{1}{1 - \frac{1}{16}} = \frac{7}{15} = 0.45
\]
Problem 4. The representation is justified as any hexadecimal number admits a binary representation:

\[ \sum_{k \in \mathbb{Z}} b_k 16^k = \sum_{k \in \mathbb{Z}} \left[ 2^3 a_k + 2^2 a'_k + 2a''_k + a'''_k \right] 16^k \]

\[ = \sum_{k \in \mathbb{Z}} \left[ 2^3 a_k + 2^2 a'_k + 2a''_k + a'''_k \right] 2^{4k} \]

\[ = \sum_{k \in \mathbb{Z}} \left[ 2^{4k+3} a_k + 2^{4k+2} a'_k + 2^{4k+1} a''_k + 2^{4k} a'''_k \right]. \]

Following the hint, note that \( a'' = (b - a''')/2 \mod 2 \), thus we set \( b'' \equiv (b' - a'')/2 = 2a + a' \), \( a' = (b' - a')/2 \mod 2 \), and \( b''' \equiv (b' - a')/2 = a \). This computes the binary digits \( a''', a'', a', a \) for a hexadecimal digit \( b \). In a hexadecimal number we iterate across all digits and replace with the 4 associated binary digits in this way.

Problem 5. Conversion code:

```matlab
a=10; b=11; c=12; d=13; e=14; f=15;
sgn=1;

SE=polyval([c 0 2],16)
E=SE-1023;
if E > 2048 substract 2048 and set sgn = -1
    E=E-2^11
else
    E=E
end

F=polyval(fliplr([3 b d 3 c 9 b e 4 5 d e]),1/16)/16
y=sgn*(1+F)*2^E
```

(a) We find \( \sigma = 1 \) and \( E = 1026 - 1023 = 3 \). (b) We find \( F = 0.233700550136170 \). (c) The full number is \(-9.869604401089358 \). This is a double precision approximation to \(-\pi^2 \).

Problem 6. Let \( x = \sigma(0.a_1a_2a_3 \cdots)_{\beta} \beta^e \). With rounding we have

\[ \hat{x} = \begin{cases} 
\sigma(0.a_1a_2a_3 \cdots a_k)_{\beta} \beta^e & \text{if } a_{k+1} < \beta/2 \\
\sigma(0.a_1a_2a_3 \cdots [a_k + 1])_{\beta} \beta^e & \text{else} 
\end{cases} \]

Case 1: \( 0 \leq a_{k+1} < \beta/2 \)

\[ x - fl(x) = \sigma(0.0\cdots0a_{k+1} \cdots)_{\beta} \beta^e \]

\[ x - fl(x) = \frac{0.0 \cdots 0a_{k+1}a_{k+2}a_{k+3} \cdots}{(0.a_1a_2 \cdots a_{k+1} \cdots)_{\beta}} = \frac{a_{k+1}a_{k+2}a_{k+3} \cdots}{(a_1a_2 \cdots a_k \cdots)_{\beta}} \beta^{-k} \]

Let’s bound the numerator and denominators (both are positive). Since \( \beta \) is an even integer, if \( a_{k+1} < \frac{\beta}{2} \) then \( a_{k+1} \leq \frac{\beta}{2} - 1 \). Let’s use this on the numerator

\[ (a_{k+1}a_{k+2}a_{k+3} \cdots)_{\beta} = a_{k+1} + (0.a_{k+2}a_{k+3} \cdots)_{\beta} \leq a_{k+1} + 1 \leq \frac{\beta}{2}. \]

We bound the denominator from below, \( (a_1.a_2.a_k \cdots)_{\beta} > 1 \). So we have

\[ \frac{x - fl(x)}{x} \leq \frac{1}{2} \beta^{-k+1}. \]
Case 2: $\frac{\beta}{2} \leq a_{k+1} < \beta$. Here, write down the relative error

$$\frac{fl(x) - x}{x} = \frac{(0.0\ldots011\ldots0a_{k+2}\ldots\beta)}{(0.a_1a_2\ldots a_k\ldots))_\beta} = 1 - \frac{(0.a_{k+1}a_{k+2}\ldots\beta)}{(0.a_1a_2\ldots a_k\ldots))_\beta} \beta^{-k}.$$

Again we bound the positive numerator and the positive denominator. To bound the numerator, we notice that since $\frac{\beta}{2} \leq a_{k+1} < \beta$, hence

$$0.a_{k+1}a_{k+2}\ldots\beta = a_{k+1}\beta + \ldots \geq a_{k+1}\beta \geq \frac{1}{2}.$$

or, equivalently, $1 - (0.a_{k+1}a_{k+2}\ldots\beta) \leq \frac{1}{2}$. For the denominator, using that $a_1 > 1$, we deduce that $(0.a_1a_2\ldots a_k) \geq \frac{1}{2}$. So our inequality becomes

$$\frac{fl(x) - x}{x} \leq \frac{1}{2}^{\beta-k+1}.$$

In either case, we prove the claim as required.

**Problem 7.**

(a) The relative error is $3.9175 \times 10^{-5} \div 10^{-5}$ so that there are 5 significant figures.

(b) The relative error is $0.7947 \div 10^0$ so that there are 0 significant figures.

(c) The relative error is $2.1012 \times 10^{-4} \div 10^{-4}$ so that there are 4 significant figures.

**Problem 8.** We have

$$\text{Rel}(\hat{x}/\hat{y}) = \frac{\hat{x}/\hat{y} - x/y}{x/y} = \frac{x/y - \hat{x}/\hat{y}}{1 + \text{Rel}(\hat{x})} - 1 = \frac{\text{Rel}(\hat{x}) - \text{Rel}(\hat{y})}{\text{Rel}(\hat{y}) + 1}.$$  

Note Rel($\hat{y}$) + 1 $\approx$ 1, which concludes our argument.

**Problem 9.**

(a) Rewrite: (i) $y^3-x^3 = (y-x)(y^2+xy+x^2)$, (ii) $\log y - \log x = \log(y/x) = \log(1+(y-x)/x) = (y-x)/x - (y-x)/x^2 + (y-x)/x^3 + O((y-x)^4)$, and (iii) as

$$\tan y - \tan x = \frac{\sin y \cos x - \sin x \cos y}{\cos x \cos y} = \frac{\sin(y-x)}{\cos x \cos y}.$$

(b) Evaluation:

```matlab
x=2; d=1e-5; y=x+d;
xs=single(x); ys=single(y); ds=single(d);
format long
bad1=ys^3-xs^3
good1=ds*(ys^2+xs*ys+xs^2)
true1=ds*(y^2+x*y+x^2)
relb1=double(bad1)/true1-1
relg1=double(good1)/true1-1
pause
bad2= log(ys)-log(xs)
good2= (ds*xs)^3/3;
good2= good2-(ds*xs)^2/2;
good2= good2+(ds*xs)
true2= (d/x)-(d/x)^2/2+(d/x)^3/3+(d/x)^4/4
relb2=double(bad2)/true2-1
relg2=double(good2)/true2-1
pause
```
bad3 = \tan(ys) - \tan(xs) \\
good3 = \sin(ds) / \cos(xs) / \cos(ys) \\
true3 = \sin(d) / \cos(x) / \cos(y) \\
relb3 = \text{double}(bad3) / true3 - 1 \\
relg3 = \text{double}(good3) / true3 - 1 \\

We find we gain respectively: (i) 5 significant digits, (ii)* 5 significant digits, and (iii) 4 significant digits. Note in example (ii) that you might imagine that it would suffice to write $\log y - \log x = \log(y/x)$ because division is always well-conditioned. However, the condition number of the logarithm function is given by $K(\ln z) = \frac{z}{\ln z} \cdot \frac{1}{\ln z}$ and this becomes large for $z \approx 1$.

**Problem 10.** (a) Since $f_n'(x) = nx^{n-1}$, we have $K(f_n, x) = x f_n'(x) / f_n(x) = n$.

(b) We check that $y_2(x) = x^2$. Next check that $(x - x^{-1}) x^{n-1} + x^{n-2} = x^n$. Then we're done.

(c) Code:

```matlab
x=1/3;
n=20;
a=x-1/x;
y(1)=1;
y(2)=x;
t(1)=1;
t(2)=x;
relerr(1)=0;
relerr(2)=0;
for k=2:n
    y(k+1)=a*y(k)+y(k-1);
    t(k+1)=t(k)*x;
    relerr(k+1)=y(k+1)/t(k+1)-1;
end
cond=abs(relerr)/eps;
condt=0:n;
semilogy(0:n,cond,'linewidth',2)
hold on
semilogy(0:n,condt,'linewidth',2)
xlabel('n','fontsize',15)
ylabel('cond','fontsize',15)
legend('algorithm','true','location','northwest')
```

Plot:
(d) Suppose solutions take the form $y_n = r^n$. Then:

$$r^n = \left(x - \frac{1}{x}\right) r^{n-1} + r^{n-2} \implies r^2 - \left(x - \frac{1}{x}\right) r - 1 = 0 \implies r = x, -\frac{1}{x}.$$ 

So in general $y_n = \alpha x^n + \beta (-x)^{-n}$ for some $\alpha, \beta$ determined by $y_0, y_1$. In infinite precision arithmetic, one finds with inputs $y_0 = 1, y_1 = x$ that $\alpha = 1$ and $\beta = 0$. However, in finite precision arithmetic, generally $\alpha \approx 1$ and $|\beta| \ll 1$ but non-zero. In that case, when $|x| < 1$, the second root $|\frac{1}{x}| > 1$ and $(-\frac{1}{x})^n$ grows faster with $n$ than the desired solution $x^n$. The algorithm is thus exponentially unstable, as shown by the error growth in the plot.

(e) Changing the initial condition to $x = 1/2$ in the code of part (c), we find now that the relative error is exactly zero! The reason is that $x = 1/2 = (0.1)_2$ is a machine number and so are all other numbers such as $a = x - 1/x = -3/2 = -(1.1)_2$ that appear in the algorithm. Thus, there is no round-off error and the code produces the exact result without any error.