Example for Problem 2 in Assignment #1

Use `format long g` in Matlab to see all computed digits.

1. Let $x = \frac{1}{7} \cdot 10^{-3}$. For the example

   \[ y = 1 - \cos x \]

   do the following:

   (a) Find the unavoidable error: find the expression for the condition $c_f(x)$ using pencil & paper. You can then evaluate it either by using Matlab or by using a Taylor approximation:

   \[ c_f(x) = \frac{x \cdot f'(x)}{f(x)} = \frac{x \cdot \sin x}{1 - \cos x} \bigg|_{x=10^{-3}/7} \approx 1.999999965986 \]

   \[ c_f(x) = \frac{x \cdot \sin x}{1 - \cos x} \approx \frac{x \cdot x}{1 - (1 - \frac{1}{2}x^2)} = 2 \]

   Hence the unavoidable error is $|c_f(x)| \varepsilon_M + \varepsilon_M \approx 3 \cdot 10^{-16}$.

   (b) **Compute the result using “naive evaluation” in Matlab** (use `format long g` to show all digits). Then use `vpa` to evaluate the result in high precision arithmetic. Then find the relative error. **Is the algorithm numerically stable?**

   We obtain $c_f(x) = \frac{x f'(x)}{f(x)} \approx 2$ and an unavoidable error $\approx 3\varepsilon_M \approx 3 \cdot 10^{-16}$. We then use in Matlab

   ```matlab
   x = 1e-3/7; y = 1 - cos(x)
xh = vpa('10^-3/7'); yh = vpa(1 - cos(xh))
relerror = double((y-yh)/yh)
```

   and obtain an error $\approx -3.5 \cdot 10^{-9}$. So the actual error is much larger than the unavoidable error, hence the algorithm is **numerically unstable**.

   (c) **Explain which step in the algorithm causes the large error.** Analyze this step to explain the size of the error.

   We use the algorithm $y_1 := \cos x$, $y := 1 - y_1$. The step $y := 1 - y_1$ causes the large error:

   For the function $g(y_1) = 1 - y_1$ we have the condition $c_g(y_1) = \frac{y_1}{1-y_1} \approx -9.8 \cdot 10^7$. Even if we compute the value $\hat{y}_1$ with full machine accuracy we can have $|\varepsilon_{\hat{y}_1}|$ as large as $\varepsilon_M$. Therefore for the value $\tilde{y} := 1 - \hat{y}_1$ we can have an error $|\varepsilon_{\tilde{y}}|$ as large as $|c_g(y_1)| \varepsilon_M \approx 10^{-8}$. As $\hat{y} = fI(\tilde{y})$ we have $|\varepsilon_{\hat{y}}| \leq |\varepsilon_{\tilde{y}}| + \varepsilon_M$ which can therefore also be $\approx 10^{-8}$.

   (d) **If the algorithm was numerically unstable: use a better way to compute the result in Matlab in machine arithmetic.** Check that the new error is not much larger than the unavoidable error.

   The “naive evaluation” algorithm was numerically unstable. Therefore we rewrite $f(x)$ using the formula $\cos(a + b) = \cos a \cdot \cos b - \sin a \cdot \sin b$ as follows:

   \[ 1 - \cos x = 1 - \cos\left(\frac{x}{2} + \frac{x}{2}\right) = 1 - \cos^2\left(\frac{x}{2}\right) + \sin^2\left(\frac{x}{2}\right) = 2 \sin^2\left(\frac{x}{2}\right) \]

   and evaluate the expression on the right hand side: We use the Matlab code

   ```matlab
   x = 1e-3/7; y = 2*sin(x/2)^2
   xh = vpa('10^-3/7'); yh = vpa(2*sin(xh/2)^2)
   relerror = double((y-yh)/yh)
   ```

   and obtain an error $\approx 2 \cdot 10^{-16}$ which is the same order of magnitude as the unavoidable error. Hence this algorithm is **numerically stable**.