Not entirely stable

Consider the function

$$f(x) = \frac{e^x - 1}{x}.$$

A computation of its relative condition number κ shows that $|\kappa(x)| < 1$ for all $|x| < 1$. In that sense, f is "easy" to compute accurately. In practice, though, it's not so simple.

An obvious sequence of steps to compute f is as follows:

$$y_1 = e^x, \qquad y_2 = y_1 - 1, \qquad y_3 = y_2/x. \tag{1}$$

The operations for y_1 and y_3 are well conditioned for $|x| < 1$, but the subtraction to get y_2 will suffer from cancellation error if $y_1 \approx 1$, or $x \approx 0$. That error makes this sequence of operations unstable for $x \approx 0$.

Now consider a power series expansion of f,

$$f(x) = 1 + \frac{1}{2!}x + \frac{1}{3!}x^2 + \cdots. \tag{2}$$

For $x > 0$, every term in the series is positive, leaving no possibility of cancellation error. (The analysis for negative x is more subtle.) If $x \approx 0$, then we should be able to find n such that $x^n/(n+1)!$ is smaller than machine precision, so that adding it to the terms before it will not change the result numerically. Thus a truncated form of the series can serve as a stable method for $x \approx 0$.

Goals

You will experiment with the two methods of computing f and observe their relative accuracy.

Preparation

Read section 1.3.

Procedure

Download the template script and complete it to perform the following steps.

1. MATLAB has a stable way of computing y_2 in (1) without using subtraction. You will use it to get reference "exact" values of f. Let

   ```
   x = logspace(-16,0,500);
   ```

   Create a vector y the same size as x such that y_j is the result of `expm1(x(j))/x(j)`.

2. Create a vector z such that z_j is computed using the three steps in (1) for x_j.

3. Compute a vector of relative differences between z_j and y_j. Make a log-log plot of the result as a function of x. You will see a loss of accuracy as $x \to 0$.

4. By trial and error, find a value of n such that $0.01^n/(n+1)!$ is less than `eps` (machine epsilon). This defines the last term to keep in truncating the series (2).

5. Create a vector `w` such that w_j is computed using the truncated form of (2). Repeat step 3 for `w` in place of `z`, using `semilogx` instead of `loglog`. This time you should see accuracy maintained as $x \to 0$. (If more terms of the series were kept, the accuracy could be maintained as $x \to 1$ as well, but the direct method is more efficient there.)