PS 3

Due: Weds, Feb 11

1: 2.5.10

1. f_1 and f_2 have the same values, so we only need to consider one. In the case of the first formula, we want to show

$$-\sin(x) \approx f_1(x, \delta) = \frac{\cos(x+\delta) - \cos(x)}{\delta};$$

the right hand side is a divided difference that converges to $\frac{d}{dx}\cos(x) = -\sin(x)$ as the step size δ goes to zero. We can be more precise via Taylor's theorem with remainder:

$$\cos(x+\delta) = \cos(x) - \sin(x)\delta - \frac{1}{2}\cos(\hat{x})\delta^2$$

for $\hat{x} \in [x, x+\delta]$, so the absolute error in the approximation is bounded by

$$|f_1(x,\delta) + \sin(x)| \leq \frac{1}{2}|\cos(\hat{x})||\delta| \leq \frac{1}{2}\delta.$$

2. This is just algebra:

$$f_2(x, \delta) = -2\sin(x + \delta/2)\sin(\delta/2).$$

3. The code is

```
x = 3;
d = 1e-11;
f1 = cos(x+d)-cos(x);
f2 = -2*sin(x+d/2)*sin(d/2);
g1 = f1/d + sin(x)
g2 = f2/d + sin(x)
```

The output is

```
g1 =   -4.4060e-07
g2 =    4.9500e-12
```

4. The first formula suffers cancellation. The computation of $\cos(x)$ and $\cos(x+\delta)$ is contaminated by roundoff at the absolute error level of at most 10^{-16} (given that $\cos(x)$ is so close to -1, relative error and absolute error roughly coincide in this case). Therefore, we expect to see error of at most about $10^{-16}/\delta$ or 10^{-5} in the computation of f_1/δ. What we observe is a bit below this, but not much. In contrast, the second formula is numerically nice, and we see error consistent with the error in approximating the series ($4.95 \times 10^{-12} \approx \cos(3)/2 \cdot \delta$).

1: 2.5.11 Using properties of logs, along with the fact that $(x-\sqrt{x^2-1})(x+\sqrt{x^2-1}) = 1$, we have

$$\ln(x-\sqrt{x^2-1}) = -\ln\left(\frac{1}{x-\sqrt{x^2-1}}\right) = \ln(x+\sqrt{x^2-1}).$$

The latter formula is more suitable in practice, since the first formula suffers cancellation for large x. For example, for $x = 10^{16}$, the first formula gives `-Inf` (since the argument evaluates to zero in floating point), while the second gives the correct answer of around -37.5.

1: 4.6.16 If A is orthogonal, the singular values are all one. In this case, the singular value matrix is $\Sigma = I$, and we can set $U = AW$ and $V = W^T$ for any orthogonal matrix W; that is, the singular vectors are completely non-unique.

2: Definitions Let $\hat{x} = 32$ be regarded as an approximation to the positive solution for $f(x_*) = x_*^2 - 1000 = 0$. What are the absolute error, the relative error, and the residual error?

Answer: Absolute error:

$$|\hat{x} - x_*| = |\sqrt{1000} - 32| \approx 0.377$$

Relative error:

$$|\hat{x} - x_*|/|x_*| \approx 0.377/\sqrt{1000} \approx 0.0119$$

Residual error:

$$f(\hat{x}) = 32^2 - 1000 = 24.$$

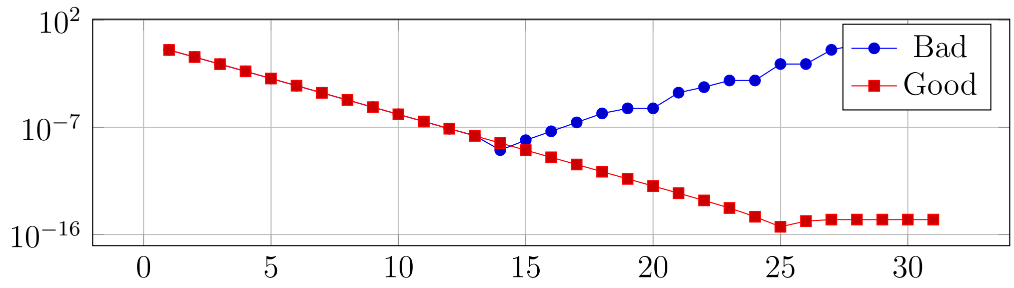

Figure 1: Error in a recurrence for π.

3: Pi, see! The following routine estimates π by recursively computing the semiperimeter of a sequence of 2^{k+1}-gons embedded in the unit circle:

```
N = 4;
L(1) = sqrt(2);
s(1) = N*L(1)/2;
for k = 1:30
  N = N*2;
  L(k+1) = sqrt( 2*(1-sqrt(1-L(k)^2/4)) );
  s(k+1) = N*L(k+1)/2;
end

fp = fopen('ps3pibad.dat', 'w');
for k = 1:length(s)
  fprintf(fp, '%d %e\n', k, abs(s(k)-pi));
end
fclose(fp);
```

Plot the absolute error $|s_k - \pi|$ against k on a semilog plot. Explain why the algorithm behaves as it does, and describe a reformulation of the algorithm that does not suffer from this problem.

Answer: We plot the error for the original and the modified codes in Figure 1. As L_k (the side length) becomes small, the expression to compute L_{k+1} suffers extreme cancellation. Eventually, we compute $L_k = 0$, at which point we are approximating π by zero. We fix by changing one line:

```
  L(k+1) = sqrt( 2*(1-sqrt(1-L(k)^2/4)) );      % Bad original
  L(k+1) = sqrt( L(k)^2/2/(1+sqrt(1-L(k)^2/4)) ); % Good replacement
```