Problem 1 Fix integer $n \geq 1$, n points x_i with $|x_i| \leq 1$, n points y_j with $|y_j| \leq 1$, n coefficients f_j, and n coefficients g_j.

(a) Fix integer $k \geq 0$. Design an algorithm for evaluating

$$f(x) = \sum_{j=1}^{n} f_j (x y_j)^k$$

at n points x_i, in $O(n)$ operations.

(b) Find a polynomial $P(x)$ with complex coefficients such that

$$|P(x) - \mathrm{e}^{\mathrm{i}x}| \leq \epsilon$$

on the interval $|x| \leq 1$.

(c) Design an algorithm for approximating

$$g(x) = \sum_{j=1}^{n} g_j \mathrm{e}^{\mathrm{i}x y_j}$$

at n points x_i in $O(n)$ operations, with absolute error bounded by

$$\epsilon \sum_{j=1}^{n} |g_j|.$$

(d) Define the $n \times n$ matrix F by

$$F_{jk} = \mathrm{e}^{\mathrm{i}x_j y_k}.$$

Find a rank r independent of n and an $n \times n$ matrix B with elements

$$B_{jk} = \sum_{i=1}^{r} c_{ji} d_{ik}$$

such that B has rank at most r and absolute error

$$|F_{jk} - B_{jk}| \leq \epsilon$$

for all n.

Problem 2 Show that floating point arithmetic sums

$$s_n = \sum_{k=1}^{n} \frac{1}{k^2} = 1 + \frac{1}{2^2} + \frac{1}{3^2} + \cdots + \frac{1}{n^2}$$

with absolute error $\leq (2n+1)\epsilon$ from left to right, while summing from right to left gives absolute error $\leq (3 + \ln n)\epsilon$. Estimate the maximum accuracy achievable and the number of terms required in each case.

Problem 3 Suppose a and b are floating point numbers with $0 < a < b < \infty$. Show that

$$a \leq \mathrm{fl}\left(\sqrt{ab}\right) \leq b,$$

in IEEE standard floating point arithmetic if no overflow occurs.

Problem 4 Design an algorithm to evaluate

$$f(x) = \frac{\mathrm{e}^x - 1 - x}{x^2}$$

in IEEE double precision arithmetic, to 12-digit accuracy for all machine numbers $|x| \leq 1$.

Problem 5 Figure out exactly what sequence of intervals is produced by bisection with the *arithmetic* mean for solving $x = 0$ with initial interval $[a_0, b_0] = [-1, 2]$. How many steps will it take to get maximum accuracy in IEEE standard floating point arithmetic?

Problem 6 Implement a MATLAB function `bisection.m` of the form

```
function [r, h] = bisection(a, b, f, p, t)
% a: Beginning of interval [a, b]
% b: End of interval [a, b]
% f: function handle y = f(x, p)
% p: parameters to pass through to f
% t: User-provided tolerance for interval width
```

At each step $j = 1$ to n, carefully choose m as in bisection with the *geometric* mean (watch out for zeroes!). Replace $[a, b]$ by the smallest interval with endpoints chosen from a, m, b which keeps the root bracketed. Repeat until a f value exactly vanishes, $b - a \le t \min(|a|, |b|)$, or b and a are adjacent floating point numbers, whichever comes first. Return the final approximation to the root r and a $3 \times n$ history matrix `h[1:3,1:n]` with column `h[1:3,j]` $=$ $(a, b, f(m))$ recorded at step j. Try to make your implementation as foolproof as possible.

(a) (See BBF 2.1.7) Sketch the graphs of $y = x$ and $y = 2\sin x$.

(b) Use `bisection.m` to find an approximation to within ϵ to the first positive value of x with $x = 2\sin x$. Report the number of steps, the final result, and the absolute and relative errors.

(c) Use `bisection.m` as many times as needed to find approximations within ϵ to all solutions $x > 0$ of the equation

$$f(x) = \frac{1}{x} + \ln x - 2 = 0.$$

Report the number of steps, the final results, and the absolute and relative errors.

(d) Use `bisection.m` to solve the equation

$$f(x) = (x - \epsilon^3)^3 = 0$$

on the interval $[-1, 2]$. Report the number of steps, the final result, and the absolute and relative errors.

(e) Use `bisection.m` to solve the equation

$$f(x) = \arctan(x - \epsilon^2) = 0$$

on the interval $[-1, 2]$. Report the number of steps, the final result, and the absolute and relative errors.