Exercise Set 1.3, page 36

1. (a) $\frac{1}{1}+\frac{1}{4}\ldots+\frac{1}{100}=1.53;\quad \frac{1}{100}+\frac{1}{81}+\ldots+\frac{1}{1}=1.54.$

   The actual value is 1.549. Significant round-off error occurs much earlier in the first method.

   (b) The following algorithm will sum the series $\sum_{i=1}^{N} x_i$ in the reverse order.

   INPUT $N; x_1, x_2, \ldots, x_N$
   OUTPUT SUM

   STEP 1 Set $SUM = 0$
   STEP 2 For $j = 1, \ldots, N$ set $i = N - j + 1$
   $SUM = SUM + x_i$
   STEP 3 OUTPUT(SUM);
   STOP.

2.

	Approximation	Absolute Error	Relative Error
(a)	2.715	3.282×10^{-3}	1.207×10^{-3}
(b)	2.716	2.282×10^{-3}	8.394×10^{-4}
(c)	2.716	2.282×10^{-3}	8.394×10^{-4}
(d)	2.718	2.818×10^{-4}	1.037×10^{-4}

3. (a) 2000 terms (b) 20,000,000,000 terms

4. 4 terms

5. 3 terms

6. (a) $O\left(\frac{1}{n}\right)$ (b) $O\left(\frac{1}{n^2}\right)$ (c) $O\left(\frac{1}{n^2}\right)$ (d) $O\left(\frac{1}{n}\right)$

7. The rates of convergence are:

   (a) $O(h^2)$ (b) $O(h)$ (c) $O(h^2)$ (d) $O(h)$

8. (a) $n(n+1)/2$ multiplications; $(n+2)(n-1)/2$ additions.

   (b) $\sum_{i=1}^{n} a_i \left(\sum_{j=1}^{i} b_j\right)$ requires n multiplications; $(n+2)(n-1)/2$ additions.

9. The following algorithm computes $P(x_0)$ using nested arithmetic.

   INPUT $n, a_0, a_1, \ldots, a_n, x_0$
   OUTPUT $y = P(x_0)$

   STEP 1 Set $y = a_n$.
   STEP 2 For $i = n-1, n-2, \ldots, 0$ set $y = x_0 y + a_i$.
   STEP 3 OUTPUT (y);
   STOP.

10. The following algorithm uses the most effective formula for computing the roots of a quadratic equation.

INPUT A, B, C.
OUTPUT x_1, x_2.

STEP 1 If $A = 0$ then
if $B = 0$ then OUTPUT ('NO SOLUTIONS');
STOP.
else set $x_1 = -C/B$;
OUTPUT ('ONE SOLUTION',x_1);
STOP.

STEP 2 Set $D = B^2 - 4AC$.

STEP 3 If $D = 0$ then set $x_1 = -B/(2A)$;
OUTPUT ('MULTIPLE ROOTS', x_1);
STOP.

STEP 4 If $D < 0$ then set
$b = \sqrt{-D}/(2A)$;
$a = -B/(2A)$;
OUTPUT ('COMPLEX CONJUGATE ROOTS');
$x_1 = a + bi$;
$x_2 = a - bi$;
OUTPUT (x_1, x_2);
STOP.

STEP 5 If $B \geq 0$ then set
$d = B + \sqrt{D}$;
$x_1 = -2C/d$;
$x_2 = -d/(2A)$
else set
$d = -B + \sqrt{D}$;
$x_1 = d/(2A)$;
$x_2 = 2C/d$.

STEP 6 OUTPUT (x_1, x_2);
STOP.

11. The following algorithm produces the product $P = (x - x_0), \ldots, (x - x_n)$.

INPUT $n, x_0, x_1, \cdots, x_n, x$
OUTPUT P.

STEP 1 Set $P = x - x_0$;
$i = 1$.

STEP 2 While $P \neq 0$ and $i \leq n$ set
$P = P \cdot (x - x_i)$;
$i = i + 1$

STEP 3 OUTPUT (P);
STOP.

12. The following algorithm determines the number of terms needed to satisfy a given tolerance.

    INPUT number x, tolerance TOL, maximum number of iterations M.
    OUTPUT number N of terms or a message of failure.

    STEP 1 Set $SUM = (1-2x)/(1-x+x^2)$;
    $S = (1+2x)/(1+x+x^2)$;
    $N = 2$.

    STEP 2 While $N \le M$ do Steps 3–5.

    STEP 3 Set $j = 2^{N-1}$;
    $y = x^j$
    $t_1 = \frac{jy}{x}(1-2y)$;
    $t_2 = y(y-1)+1$;
    $SUM = SUM + t_1/t_2$.

    STEP 4 If $|SUM - S| < TOL$ then
    OUTPUT (N);
    STOP.

    STEP 5 Set $N = N+1$.

    STEP 6 OUTPUT('Method failed');
    STOP.

    When $TOL = 10^{-6}$, we need to have $N \ge 4$.

13. (a) If $|\alpha_n - \alpha|/(1/n^p) \le K$, then $|\alpha_n - \alpha| \le K(1/n^p) \le K(1/n^q)$ since $0 < q < p$. Thus, $|\alpha_n - \alpha|/(1/n^p) \le K$ and $\{\alpha_n\}_{n=1}^{\infty} \to \alpha$ with rate of convergence $O(1/n^p)$.

    (b)

n	$1/n$	$1/n^2$	$1/n^3$	$1/n^5$
5	0.2	0.04	0.008	0.0016
10	0.1	0.01	0.001	0.0001
50	0.02	0.0004	8×10^{-6}	1.6×10^{-7}
100	0.01	10^{-4}	10^{-6}	10^{-8}

    The most rapid convergence rate is $O(1/n^4)$.

14. (a) If $F(h) = L + O(h^p)$, there is a constant $k > 0$ such that

$$|F(h) - L| \le kh^p,$$

    for sufficiently small $h > 0$. If $0 < q < p$ and $0 < h < 1$, then $h^q > h^p$. Thus, $kh^p < kh^q$, so

$$|F(h) - L| \le kh^q \quad \text{and} \quad F(h) = L + O(h^q).$$

    (b) For various powers of h we have the entries in the following table.

h	h^2	h^3	h^4
0.5	0.25	0.125	0.0625
0.1	0.01	0.001	0.0001
0.01	0.0001	0.00001	10^{-8}
0.001	10^{-6}	10^{-9}	10^{-12}

The most rapid convergence rate is $O\left(h^4\right)$.

15. Suppose that for sufficiently small $|x|$ we have positive constants k_1 and k_2 independent of x, for which

$$|F_1(x) - L_1| \leq K_1|x|^\alpha \quad \text{and} \quad |F_2(x) - L_2| \leq K_2|x|^\beta.$$

Let $c = \max(|c_1|, |c_2|, 1)$, $K = \max(K_1, K_2)$, and $\delta = \max(\alpha, \beta)$.

(a) We have

$$\begin{aligned} |F(x) - c_1L_1 - c_2L_2| &= |c_1(F_1(x) - L_1) + c_2(F_2(x) - L_2)| \\ &\leq |c_1|K_1|x|^\alpha + |c_2|K_2|x|^\beta \\ &\leq cK[|x|^\alpha + |x|^\beta] \\ &\leq cK|x|^\gamma[1 + |x|^{\delta-\gamma}] \\ &\leq \tilde{K}|x|^\gamma, \end{aligned}$$

for sufficiently small $|x|$ and some constant $\tilde{K}$. Thus, $F(x) = c_1L_1 + c_2L_2 + O(x^\gamma)$.

(b) We have

$$\begin{aligned} |G(x) - L_1 - L_2| &= |F_1(c_1x) + F_2(c_2x) - L_1 - L_2| \\ &\leq K_1|c_1x|^\alpha + K_2|c_2x|^\beta \\ &\leq Kc^\delta[|x|^\alpha + |x|^\beta] \\ &\leq Kc^\delta|x|^\gamma[1 + |x|^{\delta-\gamma}] \\ &\leq \tilde{K}|x|^\gamma, \end{aligned}$$

for sufficiently small $|x|$ and some constant $\tilde{K}$. Thus, $G(x) = L_1 + L_2 + O(x^\gamma)$.

16. Since $\lim_{n\to\infty} x_n = \lim_{n\to\infty} x_{n+1} = x$ and $x_{n+1} = 1 + \frac{1}{x_n}$, we have $x = 1 + \frac{1}{x}$. This implies that $x = \left(1 + \sqrt{5}\right)/2$. This number is called the *golden ratio.* It appears frequently in mathematics and the sciences.

17. (a) 354224848179261915075 (b) $0.3542248538 \times 10^{21}$

(c) The result in part (a) is computed using exact integer arithmetic, and the result in part (b) is computed using 10-digit rounding arithmetic.

(d) The result in part (a) required traversing a loop 98 times.

(e) The result is the same as the result in part (a).

18. (a) $n = 50$ (b) $n = 500$