1.3 Algorithms and Convergence
Algorithm & Pseudocode

• **Algorithm** is an ordered sequence of unambiguous and well-defined instructions that performs some tasks.

• **Pseudocode** is an artificial and informal high-level language that describes the operating principle of a computer program or algorithm.
  – Pseudocode allows ones to focus on the logic of the algorithm without being distracted by details of language syntax.
  – The pseudo-code is a "text-based" detail (algorithmic) design tool and is complete. It describes the entire logic of the algorithm so that implementation is a task of translating line by line into source code.
  – Pseudocode also uses structured programming design.
• **Rules of pseudocode**

1. Three categories of algorithmic operations
   a) sequential operations (Sequence) - instructions are executed in order.
      • Example: "variable" = "expression".
   b) conditional operations (If-Then-Else) - a control structure that asks a true/false question and then selects the next instruction based on the answer.
      • Example:
        if "condition" then
          (subordinate) statement 1
        else
          (subordinate) statement 2
   c) iterative (loop) operations (While) - a control structure that repeats the execution of a block of instructions
      • Example:
        while "condition"
          (subordinate) statement 1
          (subordinate) statement 2

2. All statements showing "dependency" are to be indented.

3. A period (.) indicates the termination of a step.

4. A semicolon (;) separates tasks within a step.
Pseudocode Structure

INPUT:
OUTPUT:

Step1:
Step2:

etc...

etc...
Example. Compute $\sum_{i=1}^{N} x_i$

INPUT $N, x_1, x_2, \ldots, x_N$.

OUTPUT $SUM = \sum_{i=1}^{N} x_i$

Step 1 Set $SUM = 0$. // Initialize accumulator

Step 2 For $i = 1, 2, \ldots N$ do

set $SUM = SUM + x_i$. // add next term

Step 3 OUTPUT(SUM);

STOP.
Characterizing Algorithms

Error Growth

Suppose $E_0 > 0$ denotes an initial error, and $E_n$ is the error after $n$ subsequent operations.

1. If $E_n \approx CnE_0$, where $C$ is a const. independent of $n$: the growth of error is **linear**.

2. If $E_n \approx C^n E_0$, where $C > 1$: the growth of error is **exponential**.

**Remark:** linear growth is unavoidable; exponential growth must be avoided.

Stability

• Stable algorithm: small changes in the initial data produce small changes in the final result

• Unstable or conditionally stable algorithm: small changes in all or some initial data produce large errors
Example a. For any $c_1$ and $c_2$, $p_n = c_1 \left(\frac{1}{3}\right)^n + c_2 3^n$ is the solution to the recursive equation

$$p_n = \frac{10}{3} p_{n-1} - p_{n-2}, \quad \text{for } n = 2, 3, \ldots.$$  

Suppose $p_0 = 1$ and $p_1 = \frac{1}{3}$. Use 5-digit rounding arithmetic to compute $\{p_n\}$. Is the procedure stable?
Definition 1.18 Rate of convergence for sequences

Suppose $\{\beta_n\}_{n=1}^{\infty}$ is a sequence converging to 0, and $\{\alpha_n\}_{n=1}^{\infty}$ converges to a number $\alpha$. If a positive constant $K$ exists with

$$|\alpha_n - \alpha| \leq K|\beta_n|,$$

for large $n$, then $\{\alpha_n\}_{n=1}^{\infty}$ is said to converge to $\alpha$ with rate of convergence $O(\beta_n)$, indicated by $\alpha_n = \alpha + O(\beta_n)$.

Typical $\{\beta_n\}_{n=1}^{\infty}$:

$$\beta_n = \frac{1}{np} \quad \text{for some } p > 0$$
Example 2. Suppose that, for $n \geq 1$, $\alpha_n = \frac{n+1}{n^2}$ and $\hat{\alpha}_n = \frac{n+3}{n^3}$. Determine rates of convergence for these two sequences.
Definition 1.19 Rate of convergence for functions

Suppose that \( \lim_{h \to 0} G(h) = 0 \) and \( \lim_{h \to 0} F(h) = L \). If a positive constant \( K \) exists with
\[ |F(h) - L| \leq K|G(h)|, \quad \text{for sufficiently small } h, \]
then \( F(h) = L + O(G(h)) \).

Typical \( G(h) \):
\[ G(h) = h^p \quad \text{for some } p > 0 \]
Example 3. Use the third Taylor polynomial about $h = 0$ to show that \( \cosh + \frac{1}{2}h^2 = 1 + O(h^4) \).