

Computation of $\sin(x)$

Power Series

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} \dots$$

Finite number of terms of this infinite series may be used to compute an approximate value of $\sin(x)$, where x is in radian.

$$\begin{aligned}\sin x &= x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} \dots \\ &= \sum_{i \geq 0} (-1)^i \frac{x^{2i+1}}{(2i+1)!} \\ &= \sum_{i \geq 0} t_i, \text{ where } t_i = (-1)^i \frac{x^{2i+1}}{(2i+1)!}\end{aligned}$$

Inductive Definition of t_i

$$t_i = \begin{cases} x & \text{if } i = 0, \\ -t_{i-1} \frac{x^2}{2i(2i+1)} & \text{if } i > 0. \end{cases}$$

This is also called **recurrence relation** or **recursive definition**.

Approximation of $\sin(x)$

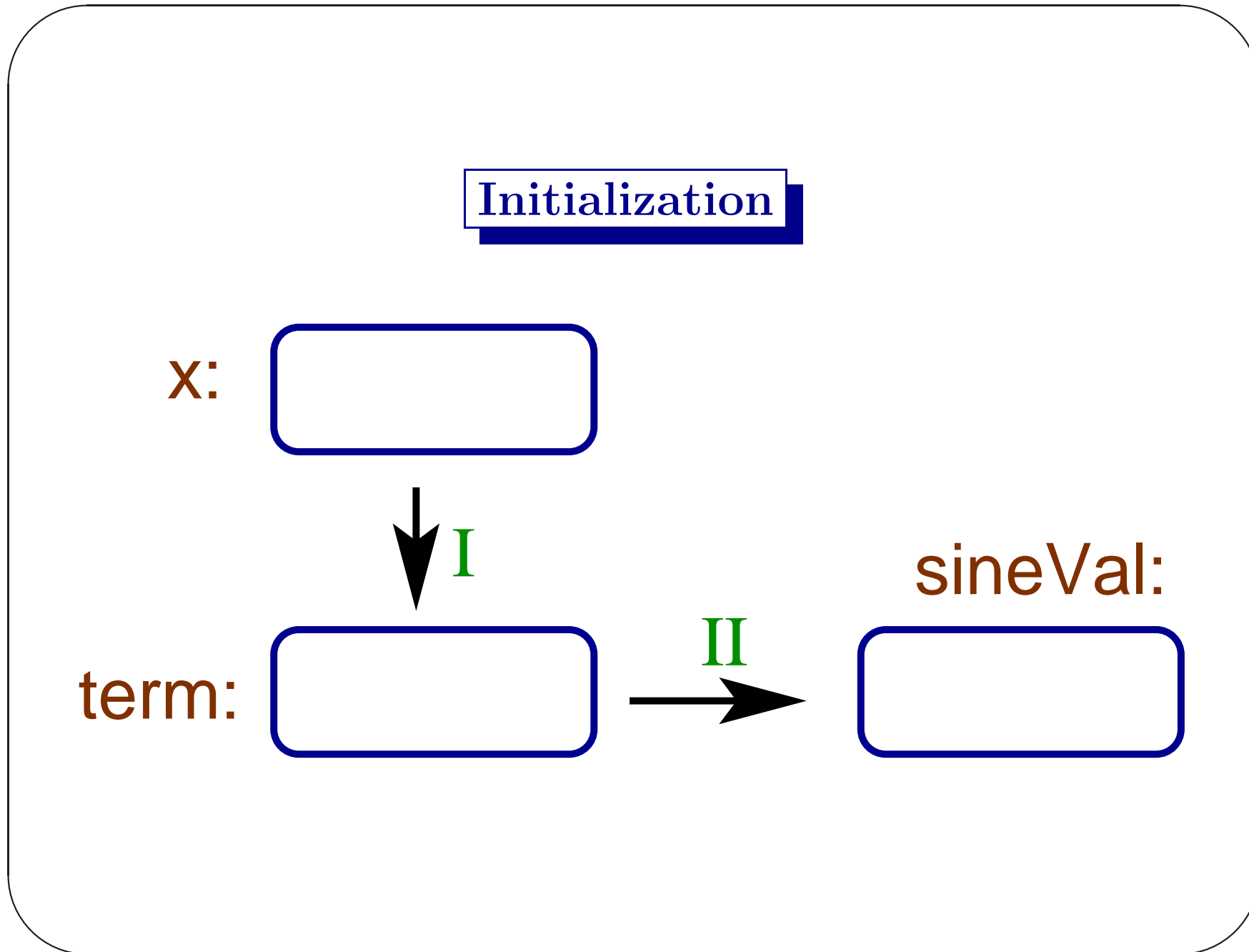
The sum upto the n^{th} term (S_n) of the series gives an approximate value of $\sin(x)$. The inductive definition of S_n is

$$S_n = \begin{cases} t_0 & \text{if } n = 0, \\ s_{n-1} + t_n & \text{if } n > 0. \end{cases}$$

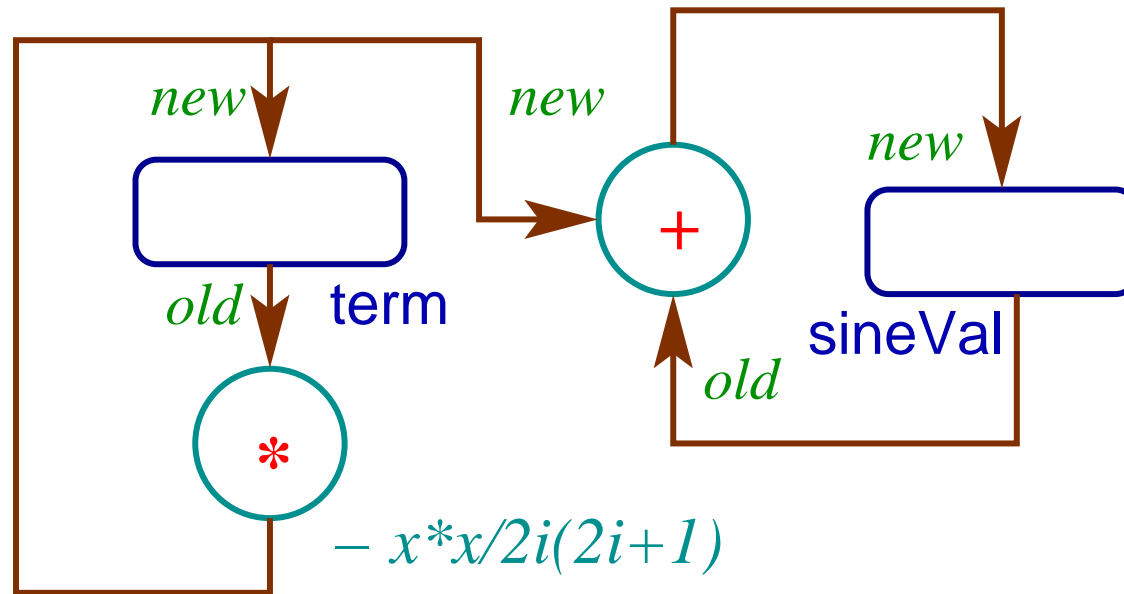
From Inductive Definition to Iterative Process

An iterative process of computation can be obtained from the inductive definition.

1. Start from **initial values** of t_i and S_i .
2. Repeat the following two steps.
 - (a) Compute the **next term**, t_{i+1} .
 - (b) Compute the next approximate value of $\sin(x)$ by computing S_{i+1} .



Iteration



Termination of Iteration

The process is to be terminated after a finite number of iterations. The termination may be

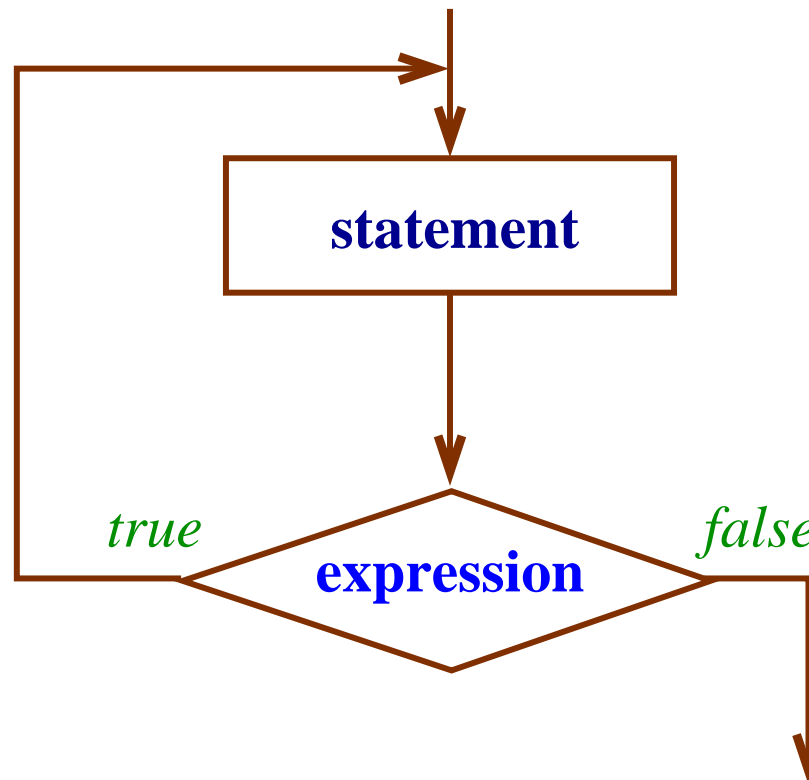
1. after a **fixed number** of iterations, or
2. after achieving a **pre-specified accuracy**.

do-while Statement

We shall use another iterative construct of C language to solve this problem.

```
do statement while (expression) ;
```

do-while Loop



```
/*
 * sin(x) using do-while
 * cc -Wall sin.c -lm
 */
#include <stdio.h>
#include <math.h>
int main()
{
    float x, precError, compError, xRadian, term, sineVal ;
    int termNo = 1 ;

    printf("Enter the value of an angle in Degree: ") ;
    scanf("%f", &x) ;
    printf("\nEnter the Percentage Error: ") ;
```

```
scanf("%f", &precError) ;

xRadian = M_PI*x/180.0 ;
term = xRadian ; // Initialization
sineVal = term ; //

do { // Iteration
    float factor ;

    factor = 2.0 * termNo++ ;
    factor *= (factor + 1.0) ;
    factor = - xRadian * xRadian / factor ;
    sineVal += (term = factor * term) ;
    compError = 100.0*fabs(term/sineVal) ;
```

```
} while (compError >= precError) ;  
  
printf("\nsin(%f) = %f\nNo. of Iterations = %d\n",  
       x, sineVal, termNo - 1) ;  
  
return 0 ;  
} // sin.c
```

Statement in do-while

```
{  
    float factor ;  
  
    factor = 2.0 * termNo++ ;  
    factor *= (factor + 1.0) ;  
    factor = - xRadian * xRadian / factor ;  
    sineVal += (term = factor * term) ;  
    compError = 100.0*fabs(term/sineVal) ;  
}
```

Expression in `do-while`

```
compError >= precError
```

Error

The program does not work for large value of angle due to error propagation (?).

```
$ ./a.out
```

```
Enter the value of an angle in Degree: 60
```

```
Enter the Percentage Error: 0.001
```

```
sin(60.000000) = 0.866026
```

```
No. of Iterations = 4
```

```
$
```


Error

```
$ ./a.out
```

```
Enter the value of an angle in Degree: 720
```

```
Enter the Percentage Error: 0.001
```

```
sin(720.000000) = 0.002326
```

```
No. of Iterations = 23
```

```
$ ./a.out
```

```
Enter the value of an angle in Degree: 1440
```

```
Enter the Percentage Error: 0.001
```

```
sin(1440.000000) = -225.040436
```

```
No. of Iterations = 36
```

```
$
```