Successive Substitution Method

OGUNLEYE OLAOLUWA,

UGWU ODIRACHUKWUNMA

Numerical Analysis, Clarkson University, NY

7th December, 2022

Contents

Abstract

This project report tends to summarize and explain Successive Substitution method in one variable as a means of solving non-linear equations.

Chapter 1

INTRODUCTION

1.1 Background

Solution of systems of linear equations are one of the fundamental problems in numerical analysis. It is also one of the easiest, in the sense that there are efficient algorithms to solve them, and that it is relatively straightforward to evaluate the results produced by them. Nonlinear equations, however, are more difficult, even when the number of unknowns is small. This calls for the invention of numerical techniques that approximates the solutions of nonlinear systems.

Numerical solution of a system of nonlinear equations is one of the more challenging tasks in numerical analysis. To start with, consider what seems at first-sight to be the solution of a single equation in one variable:

$$f(x) = 0 \tag{1.1}$$

The values of x that satisfy this equation are called the zeros or roots of the function f.

1.2 What is Successive Substitution?

Successive Substitution is a numerical method for solving a nonlinear equation for the unknown.

Algorithm:

Step 1: Rewrite $f(x) = 0$ to the form $x = g(x)$

Step 2: Evaluate $x_1 = g(x_0)$ with an initial guess x_0,

Step 3: Continue the iteration $x_{k+1} = g(x_k) \qquad k = 1, 2, \ldots$

Step 4: if $|x_{n+1} - x_n| \geq \epsilon$ GOTO **Step** 3 *else* GOTO **Step** 5

Step 5: return $x^* = x_{n+1}$ as numerical solution and **STOP**

1.3 Conditions to Consider

1. **Test for Convergence**

$$\left|\frac{df(x)}{dx}\right| < 1$$

2. **Test Vicinity of Root**

   If $f(a)f(b) < 0$, then there exists a root of $f(x)$ in the interval $[a, b]$

3. **Terminating Criteria**

   - $|x_{n+1} - x_n| \leq \epsilon \qquad$ (Absolute Error)
   - $\left|\frac{x_{n+1} - x_n}{x_{n+1}}\right| \leq \epsilon \qquad$ (Relative Error)

Remark 1.3.1. The convergence is highly dependent on how $f(x)$ is defined

Example 1.3.1. Given the equation $x^3 - 3x + 1 = 0$, our interest is to obtain a root of the given equation by method of Successive Substitution.

- **Step 1: Test Vicinity of Root**

  Consider the interval $[0, 1]$,

  $f(x) = x^3 - 3x + 1 = 0$

  $f(0) = 0 - 0 + 1 = 1$

  $f(1) = 1 - 3 + 1 = -1$

  $f(0)f(1) = 1(-1) = -1 < 0$

  Since $f(a)f(b) < 0$, then there exists a root of $f(x)$ in the interval $[0, 1]$

- **Step 2: Guess First Estimate:**

  Choose $x_0 \in [0, 1]$, say $x_0 = 0.25$

- **Step 3 Rewrite the nonlinear function** into a form given by $x = g(x)$

  $x = g(x) = \frac{1}{3}(x^3 + 1)$

- **Step 4: Test for Convergence**

  We check if $\left|\frac{dg(x)}{dx}\right|_{x_0=0.25} < 1$

  $g(x) = \dfrac{1}{3}(x^3 + 1)$

  $\dfrac{dg(x)}{dx} = x^2$

  $\dfrac{dg(x)}{dx}\Big|_{x_0=0.25} = x^2\Big|_{x_0=0.25} = |0.0625| < 1$

- **Inference**: Therefore, the iteration method with $x_0 = 0.25$ will converge to one of the roots of the equation.

Chapter 2

Convergence

2.1 Convergence Rates of Root-Finding Methods

Since there are several numerical methods to solving system of nonlinear equation, we need to develop a way of comparing the different methods by how well they work and how quickly they produce a desired level of accuracy.

Definition 2.1.1. Let $x_0, x_1, \ldots$ be a sequence of iterates produced by some root-finding method for the equation $f(x) = 0$. Then we say that the method produces convergent iterates (or just simply that it is convergent) if there exists an x^* such that

$$\lim_{n\to\infty} x_n = x^*, \tag{2.1}$$

with $f(x^*) = 0$.

Remark 2.1.1. If a method is convergent and there exists a constant L such that

$$|x_{n+1} - x^*| \leq L|x_n - x^*|^p, \tag{2.2}$$

for all n, then the method is said to have iterative order of convergence p.

Remark 2.1.2. For a first-order method, convergence can be guaranteed only if $L < 1$; for methods of higher order the error in the iterates will decrease as long as the starting guess x_0 is sufficiently close to the root.

Definition 2.1.2 (Lipschitz Condition)**.** Let $S \subset \mathbb{R}$. A function $g : S \longrightarrow \mathbb{R}$ satisfies a Lipschitz condition on S if there exists a constant $L > 0$ such that, for any two points $x, y \in S$,

$$|g(x) - g(y)| \leq L|x - y|$$

The greatest lower bound for such constants is the *Lipschitz constant* for g on S.

- If $L \in (0, 1)$ on S, then g is a **contraction** map.
- If $L = 1$, then g is a **nonexpansive** map

Example 2.1.1. A function $g : \mathbb{R} \to \mathbb{R}$ defined by

$$g(x) = \tfrac{1}{3}x + 1 \quad \forall\ x \in \mathbb{R}$$

is a contraction map with Lipschitz constant $L = \frac{1}{3}$, $L \in (0, 1)$

Example 2.1.2. A function $g(x) = x^2$ defined on $[0, \frac{1}{4}]$ is a contraction map with Lipschitz constant $L = \frac{1}{2}$, $L \in (0, 1)$

Example 2.1.3. A function $g : \mathbb{R} \to \mathbb{R}$ defined by

$$g(x) = x + 1 \quad \forall\ x \in \mathbb{R}$$

is a nonexpansive map. Here $L = 1$

Definition 2.1.3. Let $p \geq 1$. An iterative method that produces sequences $\{x_n\}$ converging to x^* converges with **order p** if there exists a constant $C > 0$ and an integer $M \geq 0$ such that

$$|x^* - x_{n+1}| \leq C|x^* - x_n|^p$$

whenever $n \geq M$

Remark 2.1.3. Observe that:

- If $p = 1$, convergence occurs when $0 < C < 1$, and the method **converges linearly.**
- If $p = 2$, the method **converges quadratically**.
- If

$$\lim_{n\to\infty} \frac{|x^* - x_{n+1}|}{|x^* - x_n|^p} = C$$

  Then we call C the **asymptotic error constant**.

Definition 2.1.4 (Superlinear Convergence)**.** An iterative method producing sequences $\{x_n\}_{n=1}^{\infty}$ converging to x^* converges **superlinearly** if there exists a sequence $\{C_n\}$ of positive real numbers such that $C_n \to 0$ and

$$|x^* - x_{n+1}| \leq C_n|x^* - x_n|$$

Theorem 2.1.1 (Convergence of Successive Substitution)**.** If the iteration function g is a contraction on some interval containing the iterates

$$x_0, x_1, x_2, \cdots$$

then g has a fixed point x^*, $x_n \to x^*$, and the successive substitution method (**??**): $x_{n+1} = g(x_n)$ converges at least linearly.

Proof

We want to show that $\{x_n\}_{n=1}^{\infty}$ converges to x^*

It suffices to show that $|x^* - x_{n+1}| \to 0$ as $n \to \infty$

$$\begin{aligned}
|x^* - x_{n+1}| &= |g(x^*) - g(x_n)| \\
&\leq L|x^* - x_n| \quad L \in (0,1) \\
&\leq L|g(x^*) - g(x_{n-1}| \\
&\leq L \cdot L|x^* - x_{n-1}| \\
&\leq L^2|g(x^*) - g(x_{n-2})| \\
&\leq L^2 \cdot L|x^* - x_{n-2}| \\
&\vdots \\
&\leq L^{n+1}|x^* - x_0| \\
|x^* - x_{n+1}| &\leq L^{n+1}|x^* - x_0| \\
\lim_{n\to\infty} |x^* - x_{n+1}| &\leq \lim_{n\to\infty} L^{n+1}|x^* - x_0| = 0 \quad \textit{since } L \in (0,1) \\
|x^* - x_{n+1}| &= 0 \quad \textit{since} \quad |x^* - x_{n+1}| \geq 0
\end{aligned}$$

Theorem 2.1.2. Let m, n be positive integers, and let $\{x_k\}_{k=1}^{\infty}$ be a sequence of iterates generated by the successive substitution method, where the iteration function g has Lipschitz constant $L \in (0,1)$ on some interval containing every iterate x_k. For $j = 0, 1, ..., m$,

$$\left|x_{m+n} - x_m\right| \leq L^{m-j} \frac{1-L^n}{1-L} \left|x_{j+1} - x_j\right|$$

Proof:

In order to prove the above theorem, we first of all need to prove the following claims:

Claim 1: $(1-L)\sum_{i=0}^{n-1} L^i = 1 - L^n$

Proof:

$$\begin{aligned}
(1-L)\sum_{i=0}^{n-1} L^i &= \sum_{i=0}^{n-1}(L^i - L^{i+1}) \\
&= 1 - L + L - L^2 + L^2 - L^3 + \cdots + L^{n-1} + L^n \\
&= 1 - L^n \\
(1-L)\sum_{i=0}^{n-1} L^i &= 1 - L^n \\
\sum_{i=0}^{n-1} L^i &= \frac{1-L^n}{1-L}
\end{aligned}$$

Claim 2: $\left|x_{i+1} - x_i\right| \leq L^{i-j}\left|x_{j+1} - x_j\right|$

$$\begin{aligned}
\left|x_{i+1} - x_i\right| &\leq |g(x_i) - g(x_{i-1})| \\
&\leq L^1 |x_i - x_{i-1}| \\
&\leq L^2 |x_{i-1} - x_{i-2}| \\
&\vdots \\
\left|x_{i+1} - x_i\right| &\leq L^{i-j} |x_{j+1} - x_j|
\end{aligned}$$

Now, we prove the actual theorem, 2.1.2,

$$\begin{aligned}\big|x_{m+n} - x_m\big| = \big|&x_{m+n} - x_{m+n-1} + x_{m+n-1} - x_{m+n-2} + x_{m+n-2} - \cdots - x_{m+n-(n-2)} \\ &+ x_{m+n-(n-2)} - x_{m+n-(n-1)} + x_{m+n-(n-1)} - x_m\big|\end{aligned}$$

$$\begin{aligned}\big|x_{m+n} - x_m\big| = \big|&x_{m+n} - x_{m+n-1} + x_{m+n-1} - x_{m+n-2} + x_{m+n-2} - \cdots - x_{m+2} \\ &+ x_{m+2} - x_{m+1} + x_{m+1} - x_m\big|\end{aligned}$$

$$\begin{aligned}
&= \Big|\sum_{i=m}^{m+n-1} x_{i+1} - x_i\Big| \\
&\leq \sum_{i=m}^{m+n-1} \big|x_{i+1} - x_i\big| \qquad (By\ triangular\ inequality) \\
&\leq \sum_{i=m}^{m+n-1} L^{i-j}|x_{j+1} - x_j| \quad (From\ Claim\ 2) \\
&= L^{-j}|x_{j+1} - x_j| \sum_{i=m}^{m+n-1} L^i \\
&= L^{-j}|x_{j+1} - x_j| \sum_{i=0}^{n-1} L^{i+m} \\
&= L^{-j} \cdot L^m |x_{j+1} - x_j| \sum_{i=0}^{n-1} L^i \\
&\leq L^{m-j}|x_{j+1} - x_j| \frac{1-L^n}{1-L} \quad (From\ Claim\ 1)
\end{aligned}$$

Remark 2.1.4. From theorem 2.1.2 above, as $n \to \infty$,

when $j = 0$:

$$|x^* - x_m| \leq \frac{L^m}{1-L}|x_1 - x_0| \quad a \ \ priori \tag{2.3}$$

when $j = m - 1$:

$$|x^* - x_m| \leq \frac{L}{1-L}|x_m - x_{m-1}| \quad a \ \ posteriori$$

Example 2.1.4. Given the following equations, our interest is to obtain the root of the given equations by method of successive substitution.

1. $x^3 - 3x + 1 = 0$

2. $x^3 + 2x + 2 - 10e^{-2x^2} = 0$

The solutions are described in the Matlab live code output below.

2.2 Implementation in Matlab

The codes are attached in the following pages.

Name: **Odirachukwunma Ugwu, Olaoluwa Ogunleye**

Topic: Successive Substitution

Course: Numerical Analysis MA 578

Date: 16th Nov 2022

Instructor: Dr Olaniyi Iyiola

Example 1 : $x^3 - 3x + 1 = 0$ on $(0, 0.5)$

define : $x_{n+1} = \frac{(x_n^3 + 1)}{3}$

Error tolerance : 0.0001

```
x_0 = 0.25;                                     % initial guess

error_tolerance = 0.0001;                       % error tolerance
prev_x = x_0;
curr_x = (prev_x^3 + 1)/3;                      % objective function
succesive_error = abs(curr_x - prev_x);         % initialize error
no_iteration = 0;
while succesive_error > error_tolerance
    prev_x                                      % for aesthetics
    curr_x = (prev_x^3 + 1)/3
    succesive_error = abs(curr_x - prev_x)
    prev_x = curr_x;
    no_iteration = no_iteration + 1;
    fprintf("    ")                             % for aesthetics
end
```

```
prev_x = 0.2500
curr_x = 0.3385
succesive_error = 0.0885

prev_x = 0.3385
curr_x = 0.3463
succesive_error = 0.0077

prev_x = 0.3463
curr_x = 0.3472
succesive_error = 9.0574e-04

prev_x = 0.3472
curr_x = 0.3473
succesive_error = 1.0888e-04

prev_x = 0.3473
curr_x = 0.3473
```

```
succesive_error = 1.3128e-05
```

```
fprintf("root is: %f, after %i iteration(s)", curr_x, no_iteration)
```

```
root is: 0.347295, after 5 iteration(s)
```

```
fprintf("\n \n \n \n ")                                    % adding space knows
```

Example 2 : $x^3 + 2x + 2 - 10e^{-2x^2} = 0$ on $(0, 1)$

define : $x_{n+1} = \sqrt{-\frac{1}{2}\ln(\frac{x^3 + 2x + 2}{10})}$

Error tolerance : 0.0001

```
x_0 = 0.5;                                                 % initial guess

error_tolerance = 0.0001;                                  % error tolerance
prev_x = x_0;
curr_x = sqrt(-log((prev_x^3 + 2*prev_x +2)/10)/2);        % objective function
succesive_error = abs(curr_x - prev_x);                    % initialize error
no_iteration = 0;
while succesive_error > error_tolerance
    prev_x                                                 % for aesthetics
    curr_x = sqrt(-log((prev_x^3 + 2*prev_x +2)/10)/2)
    succesive_error = abs(curr_x - prev_x)
    prev_x = curr_x;
    no_iteration = no_iteration + 1;
    fprintf("    ")                                        % for aesthetics
end
```

```
prev_x = 0.5000
curr_x = 0.7626
succesive_error = 0.2626

prev_x = 0.7626
curr_x = 0.6798
succesive_error = 0.0829

prev_x = 0.6798
curr_x = 0.7076
succesive_error = 0.0278

prev_x = 0.7076
curr_x = 0.6984
succesive_error = 0.0092
```

```
prev_x = 0.6984
curr_x = 0.7015
succesive_error = 0.0030

prev_x = 0.7015
curr_x = 0.7005
succesive_error = 0.0010

prev_x = 0.7005
curr_x = 0.7008
succesive_error = 3.3226e-04

prev_x = 0.7008
curr_x = 0.7007
succesive_error = 1.0991e-04

prev_x = 0.7007
curr_x = 0.7007
succesive_error = 3.6362e-05
```

```
fprintf("root is: %f, after %i iteration(s)", curr_x, no_iteration)
```

```
root is: 0.700733, after 9 iteration(s)
```

2.3 Applications

Real life application of the method of successive substitution is endless, ranging from engineering to physics, computer science and mathematics. It forms the backbone of recursion, backtracking, and dynamic programming in computer science. Essentially, any modelling problem in physics that can be transcribe into a nonlinear equation can easily make use of the tools that successive substitution provides to obtain a numerical solution (if it exist).

2.4 References

[1] Burden, R. L., Burden, A. M., Faires, J. D. (2015). Numerical Analysis. United States: Cengage Learning.

[2] Allen, M. B., Isaacson, E. L. (2019). Numerical Analysis for Applied Science. United Kingdom: Wiley.

[3] https://pages.mtu.edu/~tbco/cm3450/succ_subs.pdf

[4] https://samples.jbpub.com/9780763714994/Linz07.pdf