Practical and Efficient Evaluation of Inverse Functions

Jeffrey C. Hayen
Oregon Institute of Technology
(Jeffrey.Hayen@oit.edu)

ORMATYC 2017

PRELIMINARY EXERCISE

- **Evaluation of a Recurrence Relation (for a Mystery Function)**

$$x_{k+1} = x_k\,(2 - b\,x_k)$$

Based upon the relation above, with $b = 7$ and $x_0 = 0.2$, complete the table below:

b	k	x_k	x_{k+1}	$x = g(b)$
7.00000	0	0.20000		0.14286
	1			
	2			
	3			

Express the results entered (in columns 3 and 4) to 5 decimal places of precision.

A SET-THEORETIC APPROACH TO RELATIONS AND FUNCTIONS

Definition – (A Cartesian Product)

Let A and B be sets. The set of all ordered pairs, with the first element in A and the last element in B, is called the *Cartesian product* of A with B, which is denoted by $A \times B$. Expressed in set-builder notation,

$$A \times B = \{(x, y) : x \in A, y \in B\}$$

Definition – (A Relation)

Let A and B be sets. A *relation* from A to B is a subset of $A \times B$. Let R denote this relation, whereby $R \subseteq A \times B$. Suppose that $(x, y) \in R$. Then this association between x and y may be conveniently expressed as $x \, R \, y$.

Definition – (The Domain and Range of a Relation)

The *domain* and *range* of a relation $R \subseteq A \times B$, respectively denoted by $\operatorname{Dom}(R)$ and $\operatorname{Rng}(R)$, are the sets defined by

$$\text{Dom}(R) = \{x \in A : \exists\, y \in B \text{ with } x\,R\,y\} \quad , \quad \text{Rng}(R) = \{y \in B : \exists\, x \in A \text{ with } x\,R\,y\}$$

Definition – (A Function)

A *function* from A to B is a <u>relation</u> from A to B which has the essential properties identified below. Let f denote this relation, and suppose that $(a,b) \in f$. Then this association between a and b is customarily expressed as $f(a) = b$. Properties of f:

1. $\text{Dom}(f) = A$ **2.** $(a,b) \in f \text{ and } (a,c) \in f \;\Rightarrow\; c = b$

Also, the aggregate of f along with its domain A and *co-domain* B is often denoted by $f : A \to B$, and (a,b) is routinely called an *input-output* pair (associated by f).

Definition – (A 1-to-1 Function)

A function $f : A \to B$ is said to be *one-to-one* (or, equivalently, 1-to-1) if, and only if,

$$(a,b) \in f \text{ and } (d,b) \in f \;\Rightarrow\; d = a$$

Alternatively expressed, $f(d) = f(a) \;\Rightarrow\; d = a$**.**

Definition – (An Onto Function)

A function $f : A \to B$ is said to be *onto* if, and only if, $\operatorname{Rng}(f) = B$.

Definition – (A 1-to-1 Correspondence)

A function $f : A \to B$ is said to be a *one-to-one correspondence* if, and only if,

1. f **is 1-to-1** **2.** f **is onto**

Theorem – (An Inverse Function)

Suppose that $f : A \to B$ is a 1-to-1 correspondence. Then $\forall\, y \in B$, $\exists\, x \in A$ (which is unique) such that $f(x) = y$. This property induces a function $g : B \to A$ for which

$$g(y) = x \iff f(x) = y$$

g **is called the *inverse function* for f, and it is a 1-to-1 correspondence as well.**

Remark: g may be denoted by f^{-1}; alternatively, f may be denoted by g^{-1}. Thus, g or f may be regarded as the inverse of (or the original for) f or g, respectively.

INTERPRETATIONS AND PROPERTIES OF 1-TO-1 CORRESPONDENCES

- **A 'Mapping' Diagram for an Invertible Function (Figure 1)**
  - **Emphasis: Association of each input with a unique output.**
- **A 'Process' Diagram for an Invertible Function (Figure 2)**
  - **Emphasis: Conversion of each input into a unique output.**
  - **Remark:** $y = f(x)$ **and** $x = g(y)$ **yield the same graphs.**
  - **Remark:** $y = f(x)$ **and** $y = g(x)$ **yield different graphs.**
- **An Expanded Diagram based upon** f **as the 'Original' Function (Figure 3)**
  - **Cancellation Property 1:** $g(f(x)) = x$
- **An Expanded Diagram based upon** g **as the 'Original' Function (Figure 4)**
  - **Cancellation Property 2:** $f(g(y)) = y$

INPUT-OUTPUT TABLES FOR ORIGINAL/INVERSE FUNCTIONS

- **Example:** $f(x) = x^2$ **and** $g(y) = \sqrt{y}$ **with** $A = \mathbb{R}^+ \cup \{0\}$ **and** $B = \mathbb{R}^+ \cup \{0\}$

Table 1. Output = f(Input)

Input	1	2	3	4	5
Output	1	4	9	16	25

Table 2. Output = g(Input)

Input	1	4	9	16	25
Output	1	2	3	4	5

- **Issue – Arbitrary Inputs – for instance:** $g(3) = \mathbf{?} \Rightarrow 1 < g(3) < 2$ $[g(3) \approx 1.732]$

INPUT-OUTPUT TABLES FOR ORIGINAL/INVERSE FUNCTIONS

- **Example:** $f(x) = 2^x$ **and** $g(y) = \log_2(y)$ **with** $A = \mathbb{R}$ **and** $B = \mathbb{R}^+$

Table 3. Output = f(Input)

Input	-2	-1	0	1	2
Output	$\frac{1}{4}$	$\frac{1}{2}$	1	2	4

Table 4. Output = g(Input)

Input	$\frac{1}{4}$	$\frac{1}{2}$	1	2	4
Output	-2	-1	0	1	2

- **Issue – Arbitrary Inputs – for instance:** $g(3) = \textbf{?} \Rightarrow 1 < g(3) < 2 \quad [g(3) \approx 1.585]$

ANALYTICAL DETERMINATION OF INVERSE FUNCTIONS

- **Example: Inverse Hyperbolic Cosine Function**

$$f(x) = \cosh(x) \equiv \tfrac{1}{2}\left(e^{x} + e^{-x}\right) \quad ; \quad x \geq 0$$

$$\begin{aligned} y = \tfrac{1}{2}\left(e^{x} + e^{-x}\right) \quad &\Rightarrow \quad e^{x} + e^{-x} - 2y = 0 \\ &\Rightarrow \quad \left(e^{x}\right)^{2} - 2y\left(e^{x}\right) + 1 = 0 \\ &\Rightarrow \quad x = g(y) \end{aligned}$$

$$g(y) \triangleq \cosh^{-1}(y) = \ln\left[\, y + \sqrt{y^{2} - 1}\, \right] \quad ; \quad y \geq 1$$

- **Example: Inverse Hyperbolic Sine Function**

$$f(x) = \sinh(x) \equiv \tfrac{1}{2}\left(e^{x} - e^{-x}\right) \quad ; \quad |\, x \,| < \infty$$

$$g(y) \triangleq \sinh^{-1}(y) = \ln\left[\, y + \sqrt{y^{2} + 1}\, \right] \quad ; \quad |\, y \,| < \infty$$

STANDARD NEWTON-RAPHSON METHOD

- **Introduction to the Method**

The *Newton-Raphson method* is a numerical algorithm utilized to obtain an accurate but approximate solution or 'root' to an equation of the specific form

$$F(x) = 0 \tag{1}$$

where $F(x)$ represents any expression that involves a single variable, presumed to be denoted by x in this instance. In advanced courses on numerical analysis, it can be shown that this method (a) has desirable *convergence properties*, and (b) yields a desired root to Eq. (1) of suitable precision if such a solution exists. The algorithm inherently employs an iterative process.

Typically, the `solve` utility or equivalent feature available on scientific calculators is based upon a practical implementation of the Newton-Raphson method, which is commonly known as the *Secant method*.

An important caveat of both these methods is that a 'sufficiently-close' initial guess for the solution of interest must be provided in order for the iterative process to be successful. However, techniques to obtain an appropriate initial guess are available.

- **Derivation of the Algorithm**

Essential Concept: Locally approximate the function $F(x)$ with a linear function in order to forecast a desired root p for which $F(p)=0$. Accordingly, consider

$$y_{\mathrm{a}} = b(x-\bar{x})+c \quad , \quad y_{\mathrm{e}} = F(x) \tag{2}$$

where $\bar{x}$ is assumed to be 'close to' p. This linear function should serve to fulfill its intended purpose if the conditions stipulated below are satisfied:

$$\begin{aligned} y_{\mathrm{a}} &= y_{\mathrm{e}} \quad @ \quad x=\bar{x} \\ \frac{dy_{\mathrm{a}}}{dx} &= \frac{dy_{\mathrm{e}}}{dx} \quad @ \quad x=\bar{x} \end{aligned} \tag{3}$$

After application of these conditions, it is determined that

$$b = F'(\bar{x}) \quad , \quad c = F(\bar{x}) \tag{4}$$

As a result, the relation for y_a in Eqs. (2) becomes

$$y_\mathrm{a} = F'(\bar{x})(x - \bar{x}) + F(\bar{x}) \tag{5}$$

Next, suppose that p is a root to Eq. (1), and that $\bar{x}$ is 'nearby' p, so $\bar{x}$ is regarded as an estimate for p. Then $y_\mathrm{e} = 0$ since $F(p) = 0$. But y_a is presumed to acceptably approximate y_e. As a result, let $\bar{x} \triangleq x_k$ and $x \triangleq x_{k+1}$ for which $y_\mathrm{a} = 0$, so that Eq. (5) yields the relation

$$0 = F'(x_k)(x_{k+1} - x_k) + F(x_k) \tag{6}$$

It is desirable and effective to convert this relation into a mechanism to generate an increasingly accurate sequence of values that estimate the root p:

$$\boxed{x_{k+1} = x_k - \frac{F(x_k)}{F'(x_k)}} \tag{7}$$

where x_k is a previous estimate for p while x_{k+1} is an updated estimate for p. This relation is recursively applied to refine and improve each estimate for p, and it can be proven that the sequence of estimates $\{x_k\}$ 'rapidly' converges to the desired p under modest conditions on the function $F(x)$ and the initial guess x_0. In practice, the iterative process (and the sequence itself) is terminated when a specified degree of precision in the accuracy of the most recent estimate for p is achieved.

See Figure 5 for an insightful depiction of the iterative process and its effectiveness.

EVALUATION OF INVERSE FUNCTIONS (VIA ORIGINAL FUNCTIONS)

In applied sciences, it is often convenient to express certain formulas via an inverse function g that is based upon an original function f. It then becomes necessary to evaluate g at a particular input b. Let x denote the output of g in this instance:

$$x = g(b) \tag{8}$$

From the unique-output property of functions along with the cancellation properties

of invertible functions, Eq. (8) can be recast in terms of the <u>original function</u> f **as**

$$f(x) = f(g(b)) \quad \Rightarrow \quad f(x) = b \tag{9}$$

In order to utilize the method of Eq. (7), Eq. (9) must be realigned with Eq. (1):

$$F(x) = 0 \quad ; \quad F(x) \equiv f(x) - b \tag{10}$$

- **Example – Evaluate** $\sqrt{b}$ **for (a)** $b = \frac{1}{3}$**, and (b)** $b = 7$**.**

$$x = \sqrt{b} \quad \Rightarrow \quad x^2 = b \quad \Rightarrow \quad \begin{array}{l} F(x) = x^2 - b \\ F'(x) = 2x \end{array}$$

Based upon Eq. (7), the algorithm for the evaluation of $\sqrt{b}$ **becomes**

$$x_{k+1} = \tfrac{1}{2}\left(x_k + b/x_k\right)$$

- **Graphical Estimates and Spreadsheet Results**

- $b = \frac{1}{3} \quad \Rightarrow \quad x_0 = 0.5 \quad \Rightarrow \quad \sqrt{b} \approx x_3 = 0.57735$

- $b = 7 \quad \Rightarrow \quad x_0 = 2.5 \quad \Rightarrow \quad \sqrt{b} \approx x_2 = 2.64575$

- **Example – Evaluate $\ln(b)$ for (a) $b = \frac{1}{3}$, and (b) $b = 7$.**

$$x = \ln(b) \quad \Rightarrow \quad e^x = b \quad \Rightarrow \quad \begin{aligned} F(x) &= e^x - b \\ F'(x) &= e^x \end{aligned}$$

Based upon Eq. (7), the algorithm for the evaluation of $\ln(b)$ becomes

$$x_{k+1} = x_k - 1 + b\,e^{-x_k}$$

- **Graphical Estimates and Spreadsheet Results**

  - $b = \frac{1}{3} \quad \Rightarrow \quad x_0 = -1.0 \quad \Rightarrow \quad \ln(b) \approx x_3 = -1.09861$

  - $b = 7 \quad \Rightarrow \quad x_0 = 2.0 \quad \Rightarrow \quad \ln(b) \approx x_2 = 1.94591$

- **Example – Evaluate** $\sin^{-1}(b)$ **for (a)** $b=\frac{4}{5}$**, and (b)** $b=\frac{6}{5}$**.**

$$x=\sin^{-1}(b) \quad \Rightarrow \quad \sin(x)=b \quad \Rightarrow \quad \begin{array}{l} F(x)=\sin(x)-b \\ F'(x)=\cos(x) \end{array}$$

Based upon Eq. (7), the algorithm for the evaluation of $\sin^{-1}(b)$ **becomes**

$$x_{k+1}=x_k-\tan(x_k)+b\sec(x_k)$$

- **Graphical Estimates and Spreadsheet Results**

  - $b=\frac{4}{5} \quad \Rightarrow \quad x_0=1.0 \quad \Rightarrow \quad \sin^{-1}(b)\approx x_3=0.92730$

  - $b=\frac{6}{5} \quad \Rightarrow \quad x_0=1.0 \quad \Rightarrow \quad \sin^{-1}(b)\text{ is undefined}$

EXTENDED NEWTON-RAPHSON METHOD

- **Explanation of the Method**

This method is developed in almost the same manner as in the case of the standard method, except that the relations indicated below are utilized in an effort to forecast a desired root p for which $F(p)=0$:

$$y_{\mathrm{a}} = a(x-\overline{x})^2 + b(x-\overline{x}) + c \quad , \quad y_{\mathrm{e}} = F(x) \tag{11}$$

where $\overline{x}$ is assumed to be 'close to' p. It is anticipated that the quadratic function will fulfill its intended purpose even better than the linear function employed for the standard method if the conditions stipulated below are satisfied:

$$\begin{aligned} y_{\mathrm{a}} &= y_{\mathrm{e}} \qquad @ \quad x=\overline{x} \\ \frac{dy_{\mathrm{a}}}{dx} &= \frac{dy_{\mathrm{e}}}{dx} \qquad @ \quad x=\overline{x} \\ \frac{d^2y_{\mathrm{a}}}{dx^2} &= \frac{d^2y_{\mathrm{e}}}{dx^2} \qquad @ \quad x=\overline{x} \end{aligned} \tag{12}$$

After application of these conditions, it is determined that

$$a = \tfrac{1}{2} F''(\bar{x}) \quad , \quad b = F'(\bar{x}) \quad , \quad c = F(\bar{x}) \tag{13}$$

- **Presentation of the Algorithm**

By proceeding with the same approach as in the case of the standard method, again with $\bar{x} \triangleq x_k$ and $x \triangleq x_{k+1}$ for which $y_\mathrm{a} = 0$, the relation for y_a in Eqs. (11) yields

$$x_{k+1} - x_k = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{14}$$

where a, b, and c are evaluated as indicated in Eqs. (13), except that $\bar{x}$ is replaced by x_k. At this stage, for reasons to be discussed, it is advantageous to rationalize the numerator of Eq. (14), which then becomes

$$x_{k+1} - x_k = \frac{-2c}{b \pm \sqrt{b^2 - 4ac}} \tag{15}$$

It is now apparent that an ambiguity exists for the selection of the proper sign so as

to develop a useful recurrence relation for the algorithm of interest. This ambiguity can be resolved by first performing a simplification:

$$x_{k+1} - x_k = \frac{-2c}{b \pm |b| \sqrt{1 - 4ac/b^2}} \tag{16}$$

Next, by carefully examining representative cases (see diagrams in Figure 6), it can be concluded that the proper sign to select is $\operatorname{sgn}(b)$**. As a result,**

$$x_{k+1} - x_k = \frac{-2c}{b + \operatorname{sgn}(b) |b| \sqrt{1 - 4ac/b^2}} \tag{17}$$

or, since $\operatorname{sgn}(b)|b| = b$ **for any real number** b**,**

$$x_{k+1} = x_k - \frac{2c/b}{1 + \sqrt{1 - 4ac/b^2}} \tag{18}$$

Last, substituting the actual expressions for a**,** b**, and** c **into Eq. (18) reveals**

$$\boxed{x_{k+1} = x_k - \gamma(x_k)\,\frac{F(x_k)}{F'(x_k)}} \tag{19}$$

for which $\gamma(x_k)$ acts as a *modulating factor* that accelerates convergence, where

$$\gamma(x_k) = \frac{2}{1+\sqrt{1-\mu(x_k)}} \quad , \quad \mu(x_k) = \frac{2\,F''(x_k)\,F(x_k)}{[F'(x_k)]^2} \tag{20}$$

These particular functions possess some important properties:

1. $\mu(x_k) \to 0$ **and** $\gamma(x_k) \to 1$ **as** $F(x_k) \to 0$

2. $\mu(x_k) = 0$ **and** $\gamma(x_k) = 1$ **if** $F''(x_k) = 0$

EVALUATION OF INVERSE FUNCTIONS (VIA ORIGINAL FUNCTIONS)

The previous examples are now reconsidered via the extended method.

(See Progression of Calculations in Spreadsheet Results)

IMPROVED NEWTON-RAPHSON METHOD

- **Outline of the Method**
  - **This method is still based upon the extended method presented above.**
  - **This method utilizes a different technique to estimate the initial guess.**

The initial guess is 'bracketed' via supplying reasonable upper and lower bounds on the value of x for which $f(x) = b$. This procedure is demonstrated in a spreadsheet to be presented. An important element of this improved method is the formation of a 'locally' interpolating quadratic polynomial for $f(x)$:

$$y_{\mathrm{a}} = \bar{a}\,x^2 + \bar{b}\,x + \bar{c} \quad , \quad y_{\mathrm{e}} = f(x) \tag{21}$$

where y_{a} is assumed to 'closely follow' y_{e} between these bounds on x. But the most important characteristic of this quadratic polynomial is that, for a particular y_{a}, it <u>can be solved</u> for the required x. Thus, in the case of $y_{\mathrm{a}} = b$, the initial guess x_0 will be selected as this solution for x.

- **Procedure for the Method**

Two bounds on x **must be supplied:** x_{LB} **and** x_{UB}**, with** $x_{\mathrm{LB}} < x_{\mathrm{UB}}$**, such that**

$$\begin{array}{lll} f(x_{\mathrm{LB}}) \le b \le f(x_{\mathrm{UB}}) & \text{if} & f \text{ is strictly increasing} \\ & \text{or} & \\ f(x_{\mathrm{LB}}) \ge b \ge f(x_{\mathrm{UB}}) & \text{if} & f \text{ is strictly decreasing} \end{array} \qquad \textbf{(22)}$$

Such bounds exist because the function f **must be strictly monotonic in the vicinity of the value of** x **for which** $f(x) = b$**, a necessary condition for invertible functions.**

Based upon these bounds, let

$$\begin{array}{lcl} x_1 = x_{\mathrm{LB}} & & y_1 = f(x_1) \\ x_2 = x_{\mathrm{MP}} \equiv \frac{1}{2}(x_{\mathrm{LB}} + x_{\mathrm{UB}}) & , & y_2 = f(x_2) \\ x_3 = x_{\mathrm{UB}} & & y_3 = f(x_3) \end{array} \qquad \textbf{(23)}$$

The interpolating quadratic polynomial is then determined by the conditions

$$\begin{aligned} \overline{a}\,x_1^2 + \overline{b}\,x_1 + \overline{c} &= y_1 \\ \overline{a}\,x_2^2 + \overline{b}\,x_2 + \overline{c} &= y_2 \\ \overline{a}\,x_3^2 + \overline{b}\,x_3 + \overline{c} &= y_3 \end{aligned} \tag{24}$$

By means of the *rule of Cramer*, this system of linear equations can be solved for the variables $(\overline{a}, \overline{b}, \overline{c})$:

$$\begin{aligned} \overline{a} &= \frac{y_1(x_3 - x_2) + y_2(x_1 - x_3) + y_3(x_2 - x_1)}{(x_1 - x_2)(x_2 - x_3)(x_3 - x_1)} \\ \overline{b} &= \frac{y_1(x_2^2 - x_3^2) + y_2(x_3^2 - x_1^2) + y_3(x_1^2 - x_2^2)}{(x_1 - x_2)(x_2 - x_3)(x_3 - x_1)} \\ \overline{c} &= \frac{y_1\,x_2\,x_3(x_3 - x_2) + y_2\,x_3\,x_1(x_1 - x_3) + y_3\,x_1\,x_2(x_2 - x_1)}{(x_1 - x_2)(x_2 - x_3)(x_3 - x_1)} \end{aligned} \tag{25}$$

Finally, the initial guess x_0 is determined from the proper solution to the equation

$$\overline{a}\,x^2 + \overline{b}\,x + \overline{c} = b \tag{26}$$

whose well-known candidate solutions are

$$x_{+,-} = \frac{-\overline{b} \pm \sqrt{\overline{b}^2 - 4\,\overline{a}\,(\overline{c} - b)}}{2\,\overline{a}} \qquad \textbf{(27)}$$

The sign ambiguity apparent above is resolved as indicated below:

$$\begin{aligned} x_0 &= x_+ \quad \text{if} \quad x_{\text{LB}} < x_+ < x_{\text{UB}} \\ &\quad\;\; \text{or} \\ x_0 &= x_- \quad \text{if} \quad x_{\text{LB}} < x_- < x_{\text{UB}} \end{aligned} \qquad \textbf{(28)}$$

With x_0 determined, the *extended method* is then applied exactly as before, by means of Eq. (19), to complete the implementation of the *improved method*.

EVALUATION OF INVERSE FUNCTIONS (VIA ORIGINAL FUNCTIONS)

The previous examples are now reconsidered via the improved method.

(See Progression of Calculations in Spreadsheet Results)

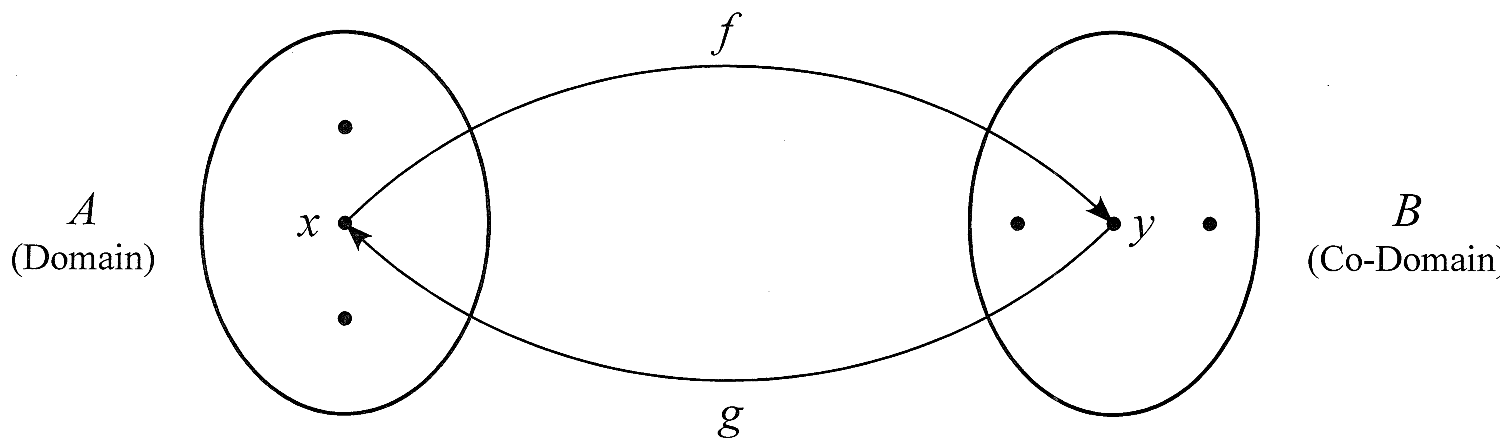


Figure 1 – A ‘Mapping’ Diagram for an Invertible Function

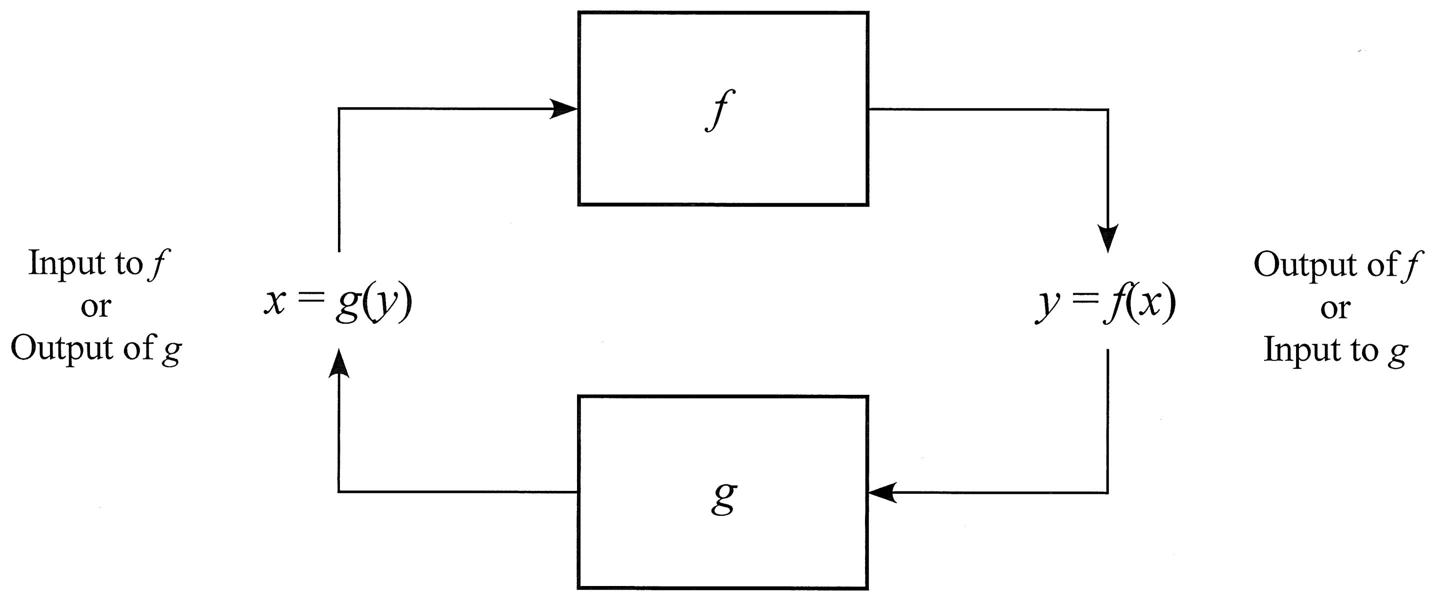
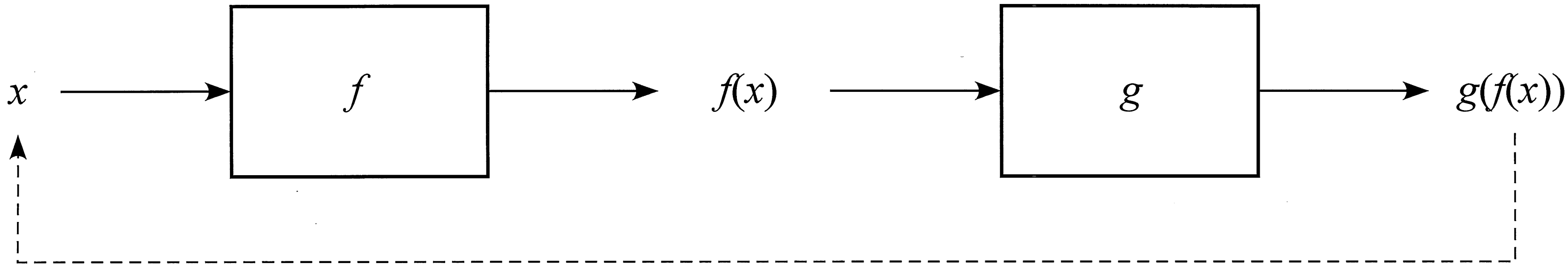
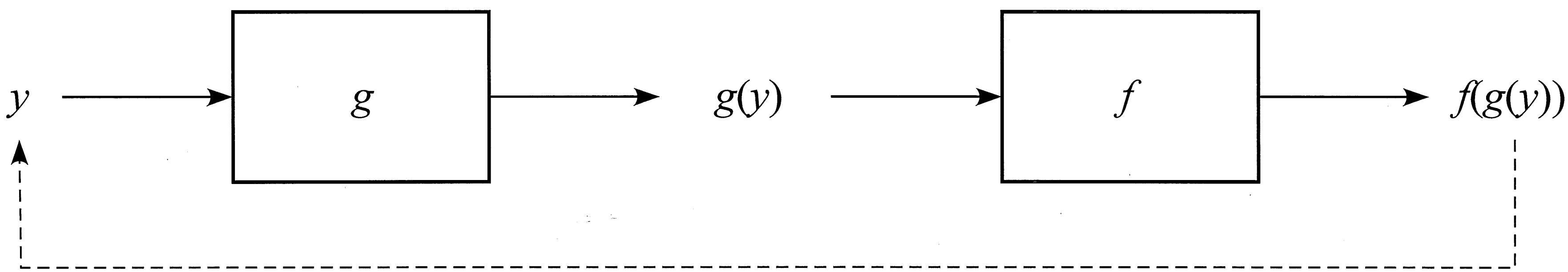


Figure 2 – A ‘Process’ Diagram for an Invertible Function



$$\Rightarrow g(f(x)) = x$$

Figure 3 – Function Notation, Composition, and Cancellation ('Original' Function: f)



$$\Rightarrow f(g(y)) = y$$

Figure 4 – Function Notation, Composition, and Cancellation ('Original' Function: g)

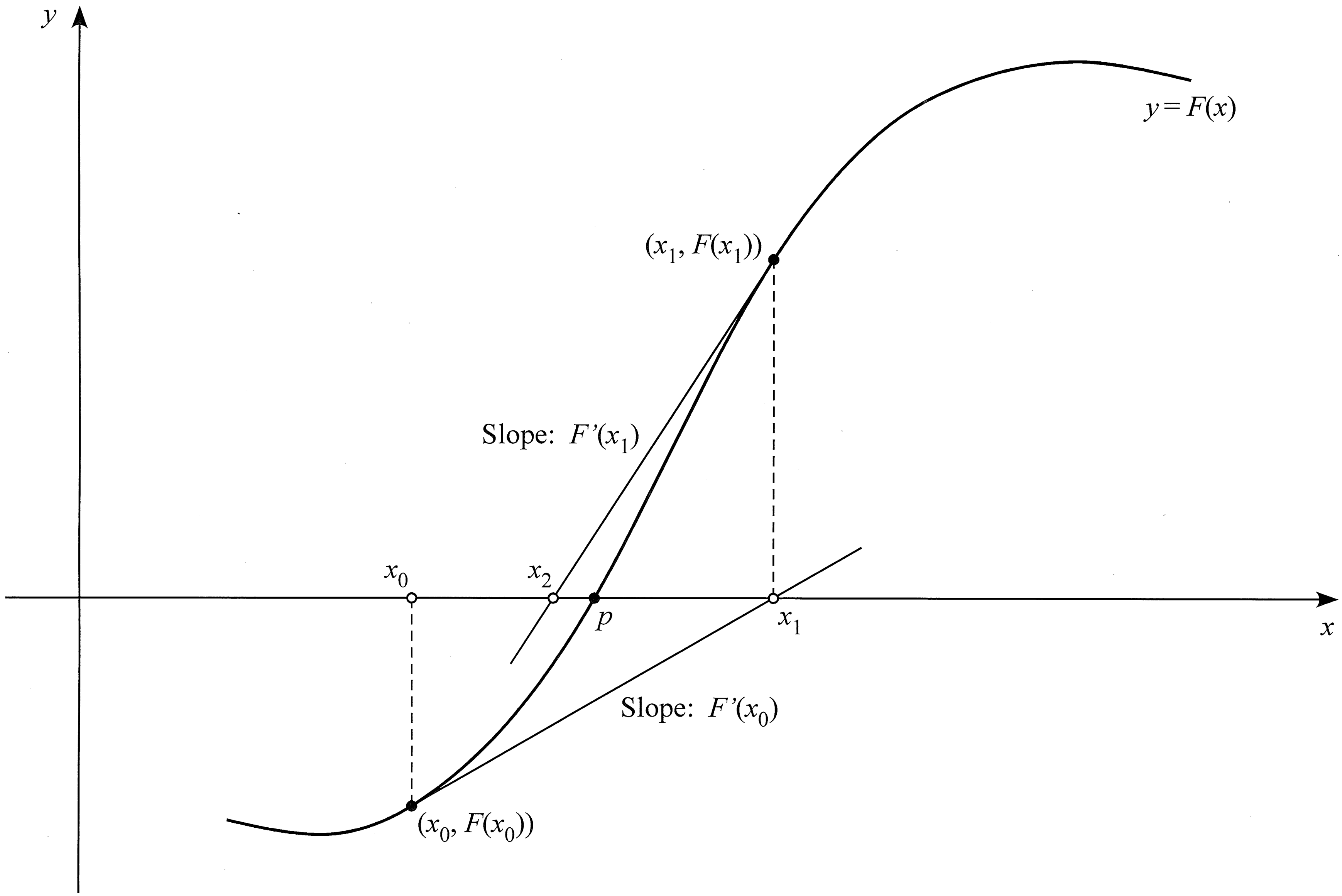


Figure 5 – Standard Newton-Raphson Method Procedure

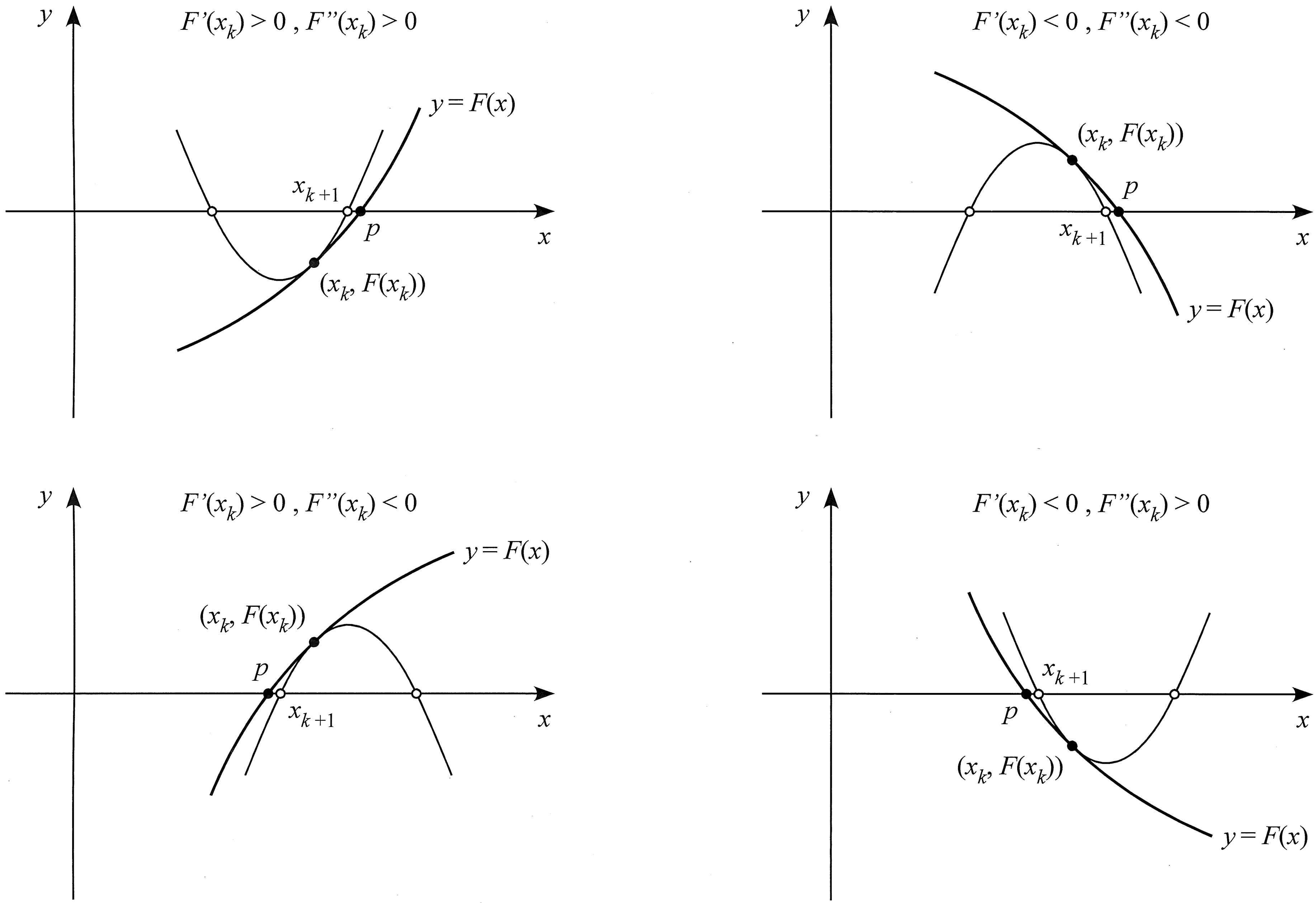


Figure 6 – Some Extended Newton-Raphson Method Cases

Reciprocal Function – Standard Method

b	k	x_k	x_{k+1}	$x = g(b)$
7.00000	0	0.20000	0.12000	0.14286
	1	0.12000	0.13920	
	2	0.13920	0.14276	
	3	0.14276	0.14286	
	4	0.14286	0.14286	
	5	0.14286	0.14286	

Square-Root Function – Standard Method

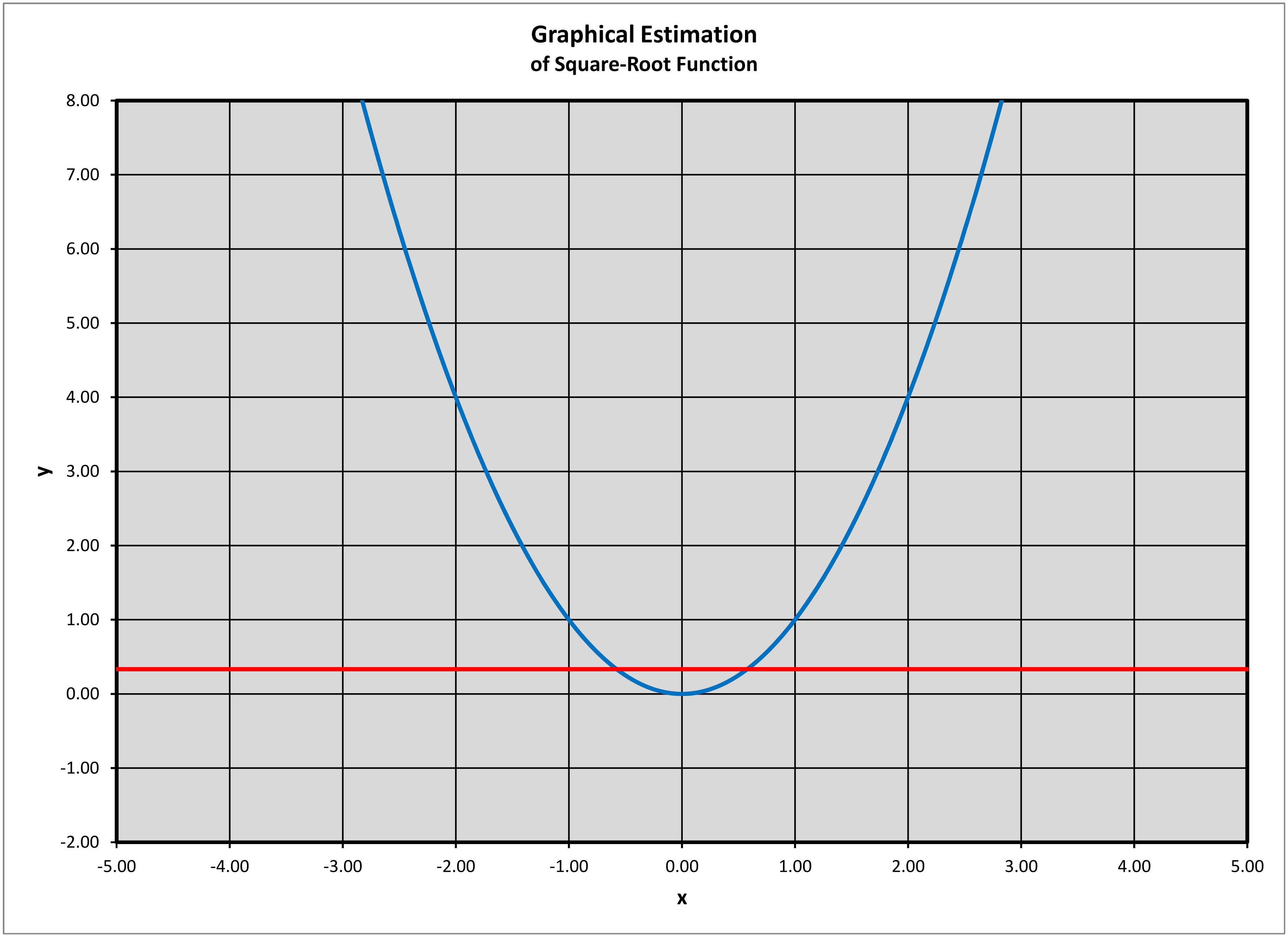
b	k	x_k	x_{k+1}	$x = g(b)$
0.33333	0	0.50000	0.58333	0.57735
	1	0.58333	0.57738	
	2	0.57738	0.57735	
	3	0.57735	0.57735	
	4	0.57735	0.57735	
	5	0.57735	0.57735	

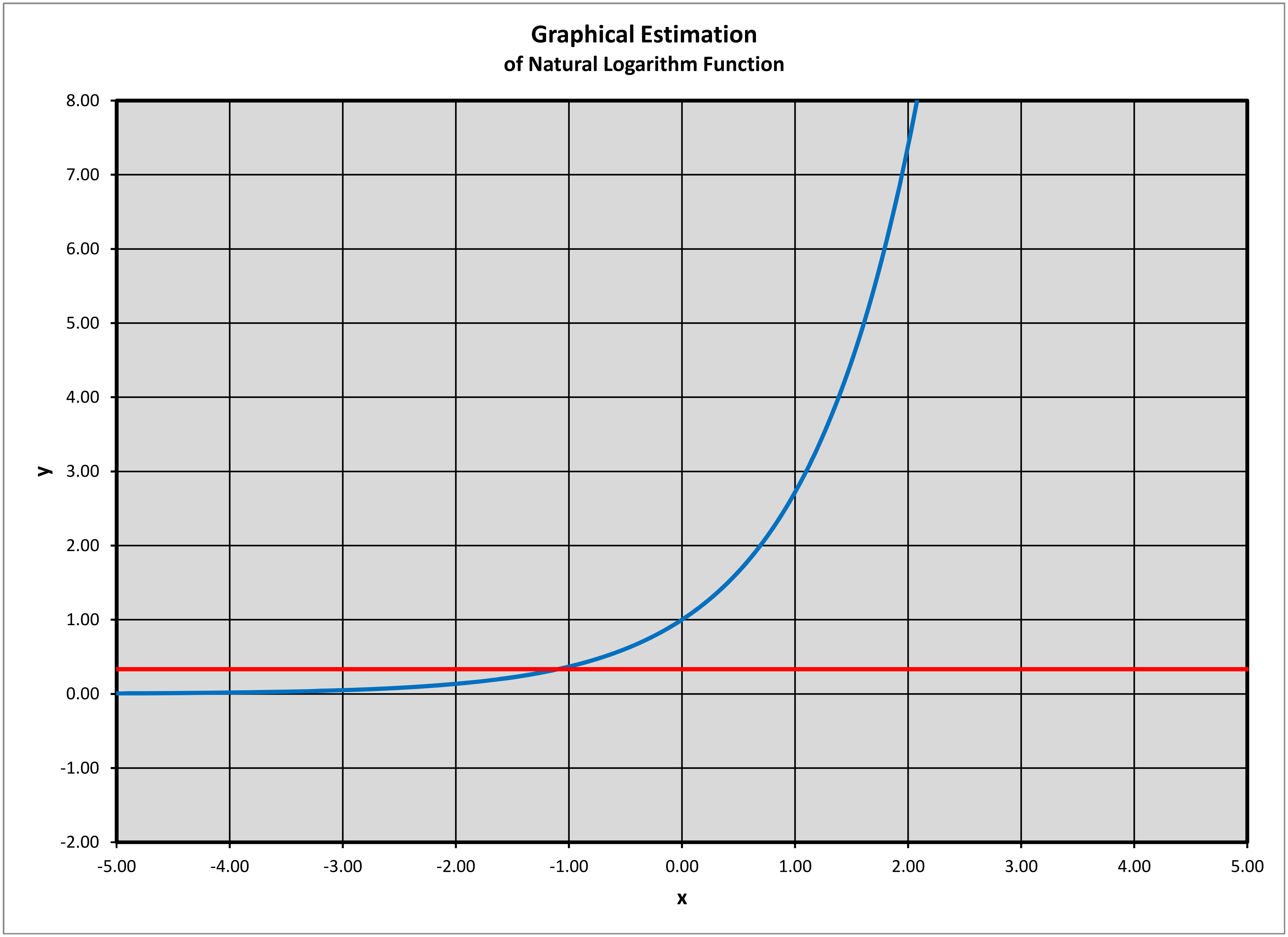
Natural Logarithm Function – Standard Method

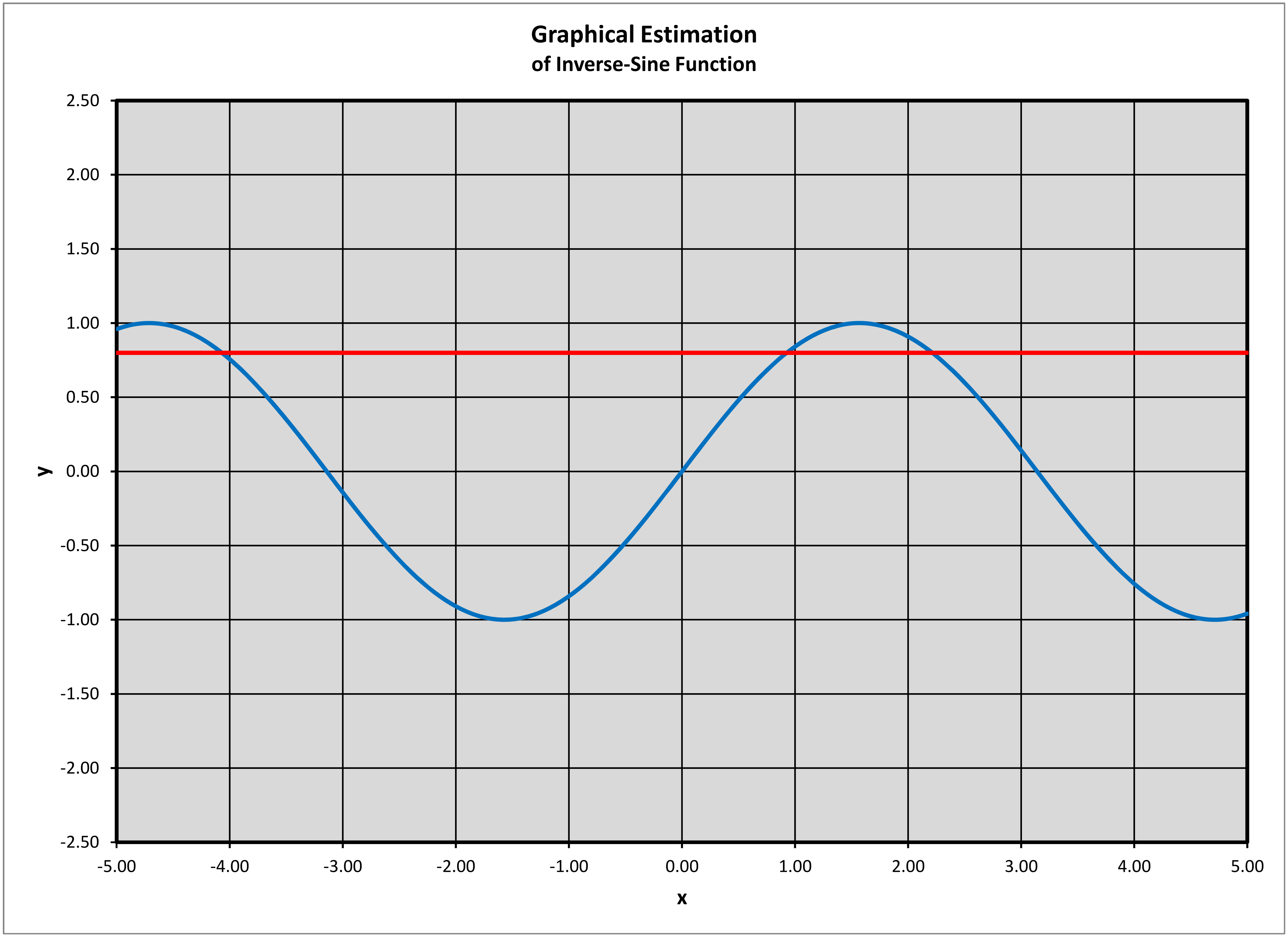
b	k	x_k	x_{k+1}	$x = g(b)$
0.33333	0	-1.00000	-1.09391	-1.09861
	1	-1.09391	-1.09860	
	2	-1.09860	-1.09861	
	3	-1.09861	-1.09861	
	4	-1.09861	-1.09861	
	5	-1.09861	-1.09861	

Inverse-Sine Function – Standard Method

b	k	x_k	x_{k+1}	$x = g(b)$
0.80000	0	1.00000	0.92324	0.92730
	1	0.92324	0.92728	
	2	0.92728	0.92730	
	3	0.92730	0.92730	
	4	0.92730	0.92730	
	5	0.92730	0.92730	


Graphical Estimation
of Square-Root Function
8.00
7.00
6.00
5.00
4.00
3.00
2.00
1.00
0.00
-1.00
-2.00
y
-5.00
-4.00
-3.00
-2.00
-1.00
0.00
1.00
2.00
3.00
4.00
5.00
x


Graphical Estimation
of Natural Logarithm Function
8.00
7.00
6.00
5.00
4.00
3.00
2.00
1.00
0.00
-1.00
-2.00
y
-5.00
-4.00
-3.00
-2.00
-1.00
0.00
1.00
2.00
3.00
4.00
5.00
x



Graphical Estimation
of Inverse-Sine Function
2.50
2.00
1.50
1.00
0.50
0.00
-0.50
-1.00
-1.50
-2.00
-2.50
y
-5.00
-4.00
-3.00
-2.00
-1.00
0.00
1.00
2.00
3.00
4.00
5.00
x

Square-Root Function – Extended Method

b	k	x_k	$F(x_k)$	$F'(x_k)$	$F''(x_k)$	x_{k+1}	$x = g(b)$
0.33333	0	0.50000	-0.08333	1.00000	2.00000	0.57735	0.57735
	1	0.57735	0.00000	1.15470	2.00000	0.57735	
	2	0.57735	0.00000	1.15470	2.00000	0.57735	
	3	0.57735	0.00000	1.15470	2.00000	0.57735	
	4	0.57735	0.00000	1.15470	2.00000	0.57735	
	5	0.57735	0.00000	1.15470	2.00000	0.57735	

Natural Logarithm Function – Extended Method

b	k	x_k	$F(x_k)$	$F'(x_k)$	$F''(x_k)$	x_{k+1}	$x = g(b)$
0.33333	0	-1.00000	0.03455	0.36788	0.36788	-1.09879	-1.09861
	1	-1.09879	-0.00006	0.33328	0.33328	-1.09861	
	2	-1.09861	0.00000	0.33333	0.33333	-1.09861	
	3	-1.09861	0.00000	0.33333	0.33333	-1.09861	
	4	-1.09861	0.00000	0.33333	0.33333	-1.09861	
	5	-1.09861	0.00000	0.33333	0.33333	-1.09861	

Inverse-Sine Function – Extended Method

b	k	x_k	$F(x_k)$	$F'(x_k)$	$F''(x_k)$	x_{k+1}	$x = g(b)$
0.80000	0	1.00000	0.04147	0.54030	-0.84147	0.92735	0.92730
	1	0.92735	0.00004	0.59995	-0.80004	0.92730	
	2	0.92730	0.00000	0.60000	-0.80000	0.92730	
	3	0.92730	0.00000	0.60000	-0.80000	0.92730	
	4	0.92730	0.00000	0.60000	-0.80000	0.92730	
	5	0.92730	0.00000	0.60000	-0.80000	0.92730	

Natural Logarithm Function – Improved Method

b	k	x_k	$F(x_k)$	$F'(x_k)$	$F''(x_k)$	x_{k+1}	$x = g(b)$
0.33333	0	-1.08310	0.00521	0.33855	0.33855	-1.09861	-1.09861
	1	-1.09861	0.00000	0.33333	0.33333	-1.09861	

x_{LB}	x_{MP}	x_{UB}
-2.00000	-1.00000	0.00000
$f(x_{LB})$	$f(x_{MP})$	$f(x_{UB})$
0.13534	0.36788	1.00000

Inverse-Sine Function – Improved Method

b	k	x_k	$F(x_k)$	$F'(x_k)$	$F''(x_k)$	x_{k+1}	$x = g(b)$
0.80000	0	0.94695	0.01164	0.58416	-0.81164	0.92730	0.92730
	1	0.92730	0.00000	0.60000	-0.80000	0.92730	

x_{LB}	x_{MP}	x_{UB}
0.00000	0.75000	1.50000
$f(x_{LB})$	$f(x_{MP})$	$f(x_{UB})$
0.00000	0.68164	0.99749

Inverse-Tangent Function – Improved Method

b	k	x_k	$F(x_k)$	$F'(x_k)$	$F''(x_k)$	x_{k+1}	$x = g(b)$
1.20000	0	0.87603	-0.00007	2.43983	5.85524	0.87606	0.87606
	1	0.87606	0.00000	2.44000	5.85600	0.87606	

x_{LB}	x_{MP}	x_{UB}
0.75000	0.87500	1.00000
$f(x_{LB})$	$f(x_{MP})$	$f(x_{UB})$
0.93160	1.19742	1.55741

Dynamics II (MECH 312)
Explicit Time Solutions – An Example Problem
Vertical Projectile Motion with a Drag Force

Vertical Projectile Motion Formulas

The formulas given below exactly describe the motion of a projectile which may be modeled as a particle, has been *vertically* projected in a uniform gravitational field, and is subjected to a drag force that is proportional to the projectile speed squared: $F_{\mathrm{d}} = k\,v^2$

Note: In the formulas below, k, v_{o}, m, and g_{o} should be regarded as known parameters.

- Ascent Relations: $0 < t < t_{\mathrm{a}}$

$$y = \frac{m}{k}\ln\left\{\sqrt{1 + k\,v_{\mathrm{o}}^2/(m\,g_{\mathrm{o}})}\,\cos\left[\sqrt{k\,g_{\mathrm{o}}/m}\,(t_{\mathrm{a}} - t)\right]\right\}$$

$$v = \sqrt{m\,g_{\mathrm{o}}/k}\,\tan\left[\sqrt{k\,g_{\mathrm{o}}/m}\,(t_{\mathrm{a}} - t)\right]$$

$$t_{\mathrm{a}} = \sqrt{m/(k\,g_{\mathrm{o}})}\,\tan^{-1}\left[\sqrt{k\,v_{\mathrm{o}}^2/(m\,g_{\mathrm{o}})}\right] \quad , \quad h = \frac{m}{k}\ln\left[\sqrt{1 + k\,v_{\mathrm{o}}^2/(m\,g_{\mathrm{o}})}\right]$$

- Descent Relations: $t_{\mathrm{a}} < t < t_{\mathrm{i}}$

$$y = \frac{m}{k}\ln\left\{\sqrt{1 + k\,v_{\mathrm{o}}^2/(m\,g_{\mathrm{o}})}\,\mathrm{sech}\left[\sqrt{k\,g_{\mathrm{o}}/m}\,(t - t_{\mathrm{a}})\right]\right\}$$

$$v = \sqrt{m\,g_{\mathrm{o}}/k}\,\tanh\left[\sqrt{k\,g_{\mathrm{o}}/m}\,(t - t_{\mathrm{a}})\right]$$

$$t_{\mathrm{d}} = \sqrt{m/(k\,g_{\mathrm{o}})}\,\sinh^{-1}\left[\sqrt{k\,v_{\mathrm{o}}^2/(m\,g_{\mathrm{o}})}\right] \quad , \quad v_{\mathrm{i}} = \frac{v_{\mathrm{o}}}{\sqrt{1 + k\,v_{\mathrm{o}}^2/(m\,g_{\mathrm{o}})}}$$

- Mathematical Definitions

$$\mathrm{sech}\,(u) = \frac{2}{e^u + e^{-u}}$$

$$\tanh\,(u) = \frac{e^u - e^{-u}}{e^u + e^{-u}}$$

$$\sinh^{-1}(u) = \ln\left[u + \sqrt{1 + u^2}\right]$$

Note: All of the relations above can be simplified via a parameter α, which is defined below.

- Notation/Terminology

t : elapsed time (from the projection instant)

y : projectile position (vertical displacement)

v : projectile speed

k : drag coefficient

v_o : projection speed (initial speed)

m : particle mass

g_o : gravity constant

α : medium resistance intensity, defined by $\alpha \equiv \sqrt{k\,v_o^2/(m\,g_o)}$

t_a : ascent time

h : maximum height

t_d : descent time

v_i : impact speed

t_i : impact instant, defined by $t_i = t_a + t_d$

Note: t_a and t_d denote the durations of the ascent and descent phases, respectively, whereas t_i denotes the terminal value of t or (equivalently) the duration of the entire projectile motion.

Ascent Phase: $0 < t < t_a$

Governing Equation: $m\dfrac{dv}{dt} = -m\,g_o - k\,v^2$

Kinematic Relation: $\dfrac{dv}{dt} = v\,\dfrac{dv}{dy}$

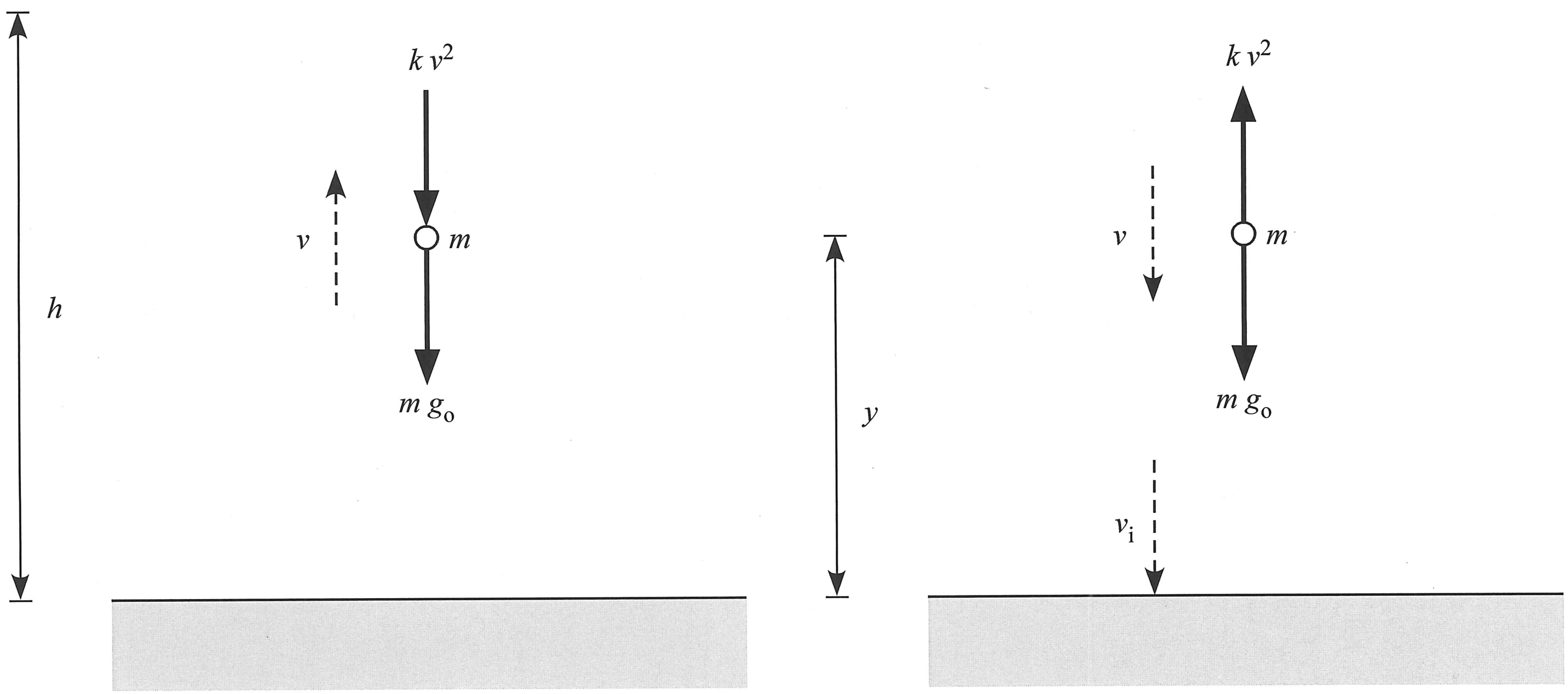
Initial Conditions: $v = v_o$, $y = 0$ @ $t = 0$

Descent Phase: $t_a < t < t_i$

Governing Equation: $m\dfrac{dv}{dt} = m\,g_o - k\,v^2$

Kinematic Relation: $\dfrac{dv}{dt} = -v\,\dfrac{dv}{dy}$

Initial Conditions: $v = 0$, $y = h$ @ $t = t_a$



Reference Diagram – Vertical Projectile Motion with a Drag Force