Numerical Methods

Lecture 1.

Introduction to numerical methods

dr hab.inż. Katarzyna Zakrzewska, prof. AGH,
Department of Electronics, AGH
e-mail: zak@agh.edu.pl
http://home.agh.edu.pl/~zak

Introduction to numerical methods

Numerical methods belong to applied mathematics focused on the development of approximate methods for solving mathematical problems that cannot be solved by exact methods or because of their large computational complexity.

Numerical methods are involved in constructing algorithms in which the input data, intermediate results and final results are represented by **numbers.**

Characteristics of numerical methods:

- calculations are performed on approximate numbers
- solutions are expressed as approximate numbers
- error in the numerical calculation should be always controlled

References:

- Z. Fortuna, B. Macukow, J. Wąsowski, Metody numeryczne, Podręczniki Akademickie EIT, WNT Warszawa,1982, 2005
- L.O. Chua, P-M. Lin, Komputerowa analiza układów elektronicznych-algorytmy i metody obliczeniowe, WNT, Warszawa, 1981
- G.Dahlquist, A.Björck, Metody matematyczne, PWN Warszawa, 1983
- Autar Kaw, Luke Snyder

  http://numericalmethods.eng.usf.edu

Additional references:

- M.Wciślik, Wprowadzenie do systemu Matlab, Wydawnictwo Politechniki Świętokrzyskiej, Kielce, 2000
- S. Osowski, A. Cichocki, K.Siwek, Matlab w zastosowaniu do obliczeń obwodowych i przetwarzania sygnałów, Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa 2006
- W.H. Press, et al., Numerical recipes, Cambridge University Press, 1986

Outline

- Solving engineering problems
- Overview of typical mathematical procedures
- Fixed and floating-point representation of numbers

How to solve an engineering problem?

Problem Description

↓

Mathematical Model

↓

Solution of Mathematical Model

↓

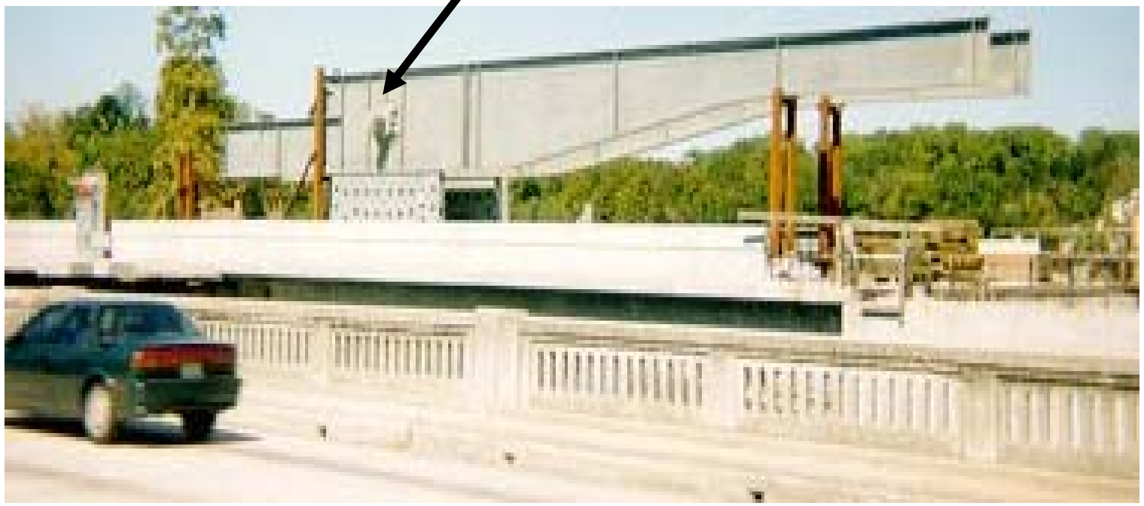
Application of the Solution

Example of Solving an Engineering Problem

the Bridge of Lions in
St. Augustine, Florida

[Autar Kaw](http://numericalmethods.eng.usf.edu)
http://numericalmethods.eng.usf.edu

Bascule Bridge THG



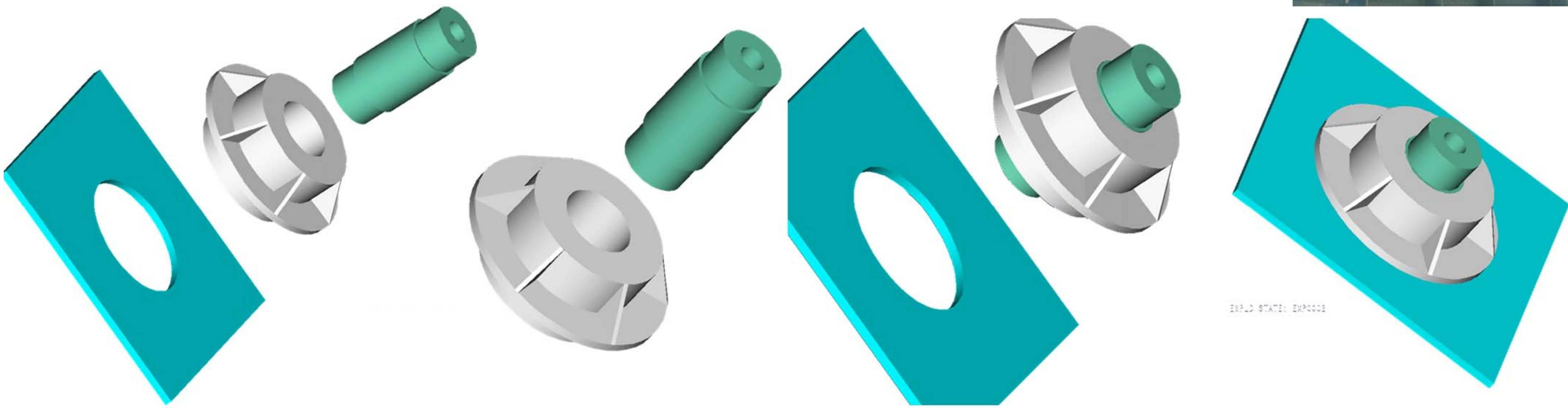
Bascule Bridge THG

Hub

Trunnion

Girder

Trunnion-Hub-Girder Assembly Procedure



Step 1. Trunnion immersed in dry-ice/alcohol (- -108 F, around -80 C)

Step 2. Trunnion warm-up in hub

Step 3. Trunnion-Hub immersed in dry-ice/alcohol

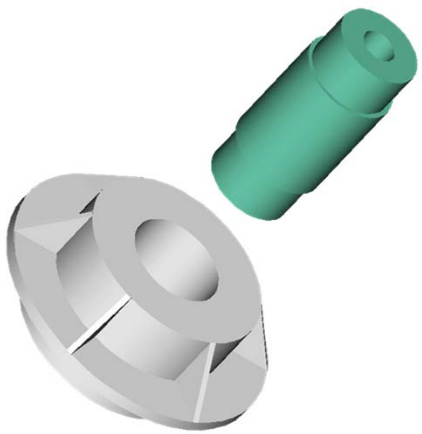
Step 4. Trunnion-Hub warm-up into girder

A problem occurred!

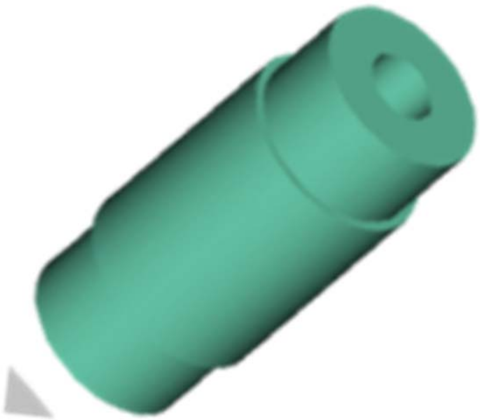
After cooling, the trunnion got stuck in the hub

Why did it get stuck?

Magnitude of contraction of the trunnion was expected to be 0.015" or more. Did it contract enough?



Calculations



$$\boxed{\Delta D = D \times \alpha \times \Delta T}$$

$$D = 12.363''$$

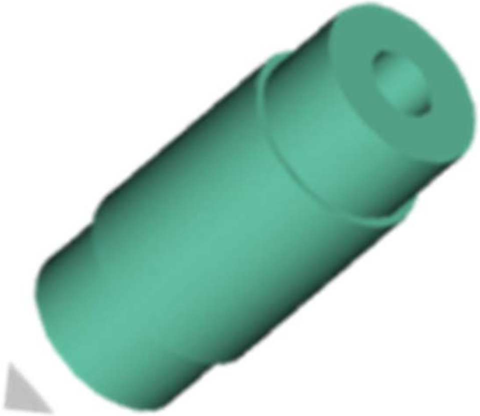
$$\alpha = 6.47 \times 10^{-6} in / in /^{o} F$$

$$\Delta T = -108 - 80 = -188^{o} F$$

$$\Delta D = (12.363)(6.47 \times 10^{-6})(-188)$$

$$= -0.01504''$$

Units



$$\mathbf{1\ in = 2.54\ cm}$$

$$D = 12.363'' \approx 31{,}4cm$$

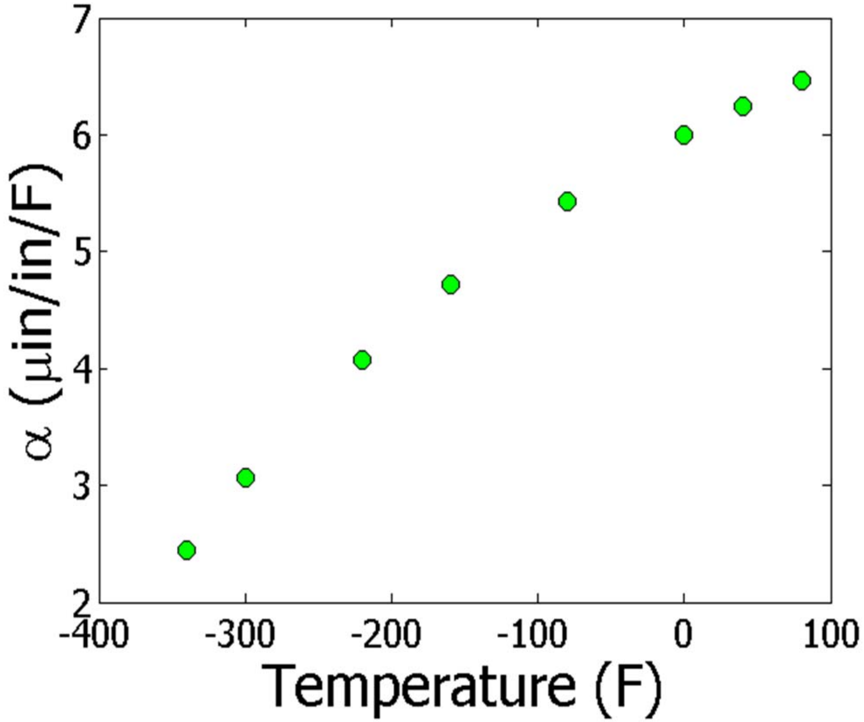
$$\boldsymbol{T}_F = \left(\frac{\mathbf{9}}{\mathbf{5}}T_C + 32\right)^0 F$$

$$T = 80^o F \approx 26{,}7^0 C$$

$$\Delta D = 0.01504\ \ in = 0.03820\ \ cm$$

Is the formula used correct?

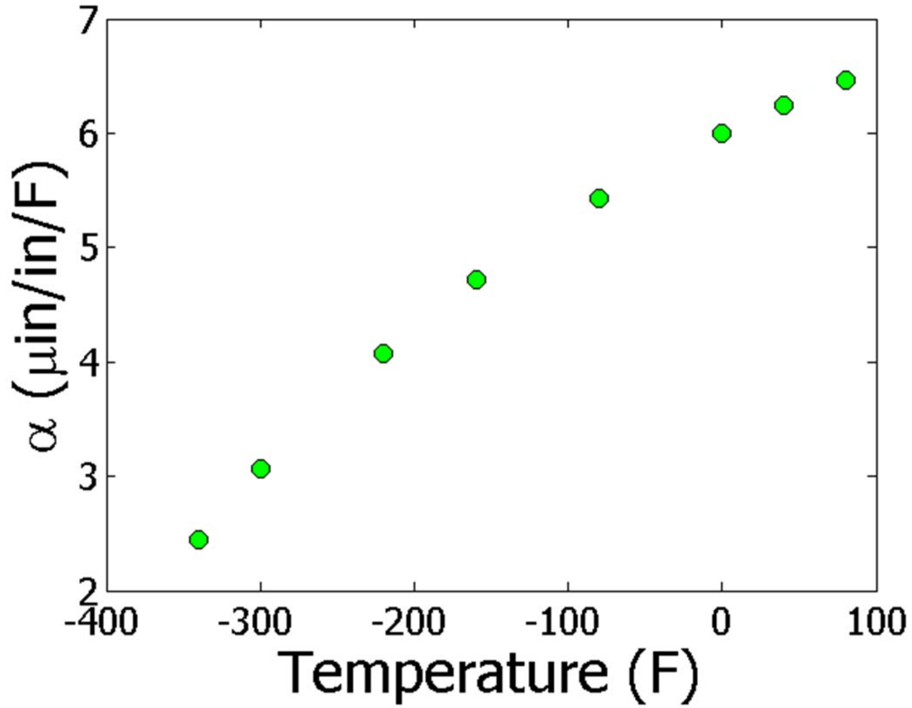
$$\Delta D = D \times \alpha \times \Delta T$$



T(°F)	α (μin/in/F)
-340	2.45
-300	3.07
-220	4.08
-160	4.72
-80	5.43
0	6.00
40	6.24
80	6.47

Problem solving

The correct model should account for varying thermal expansion coefficient α

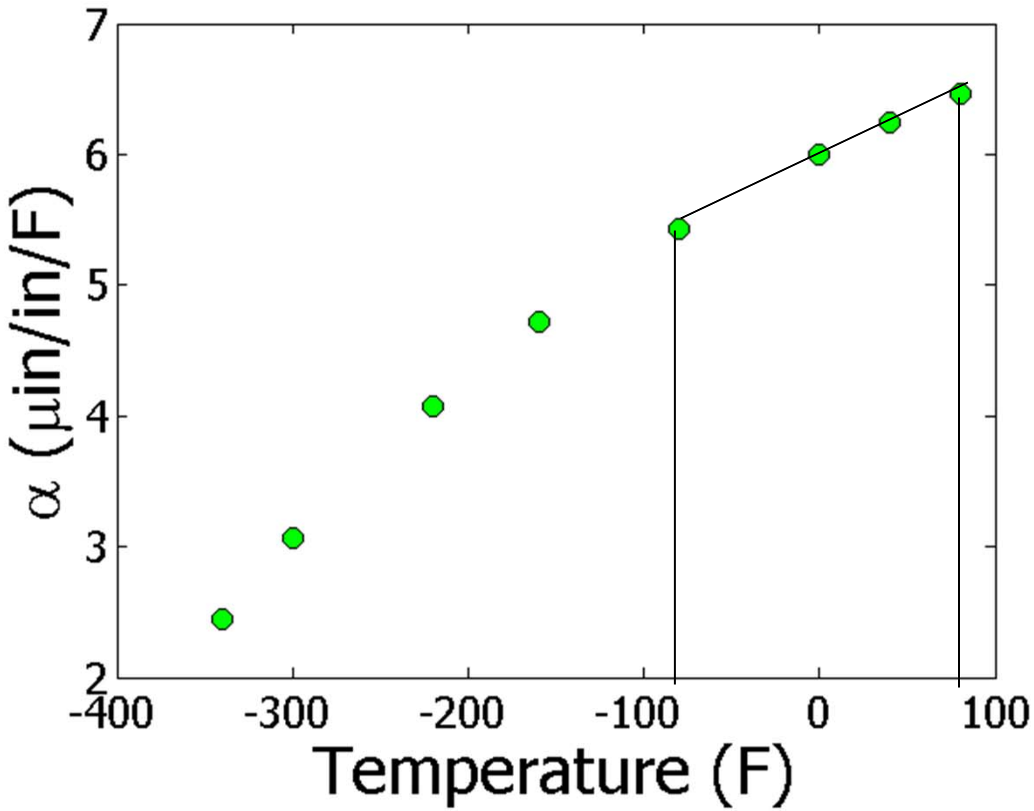


$$\Delta D = D\int_{T_a}^{T_c} \alpha(T)\,dT$$

Can you roughly estimate the contraction?

$$\Delta D = D\int_{T_a}^{T_c} \alpha(T)\,dT$$

T_a=80F; T_c=-108F; D=12.363"

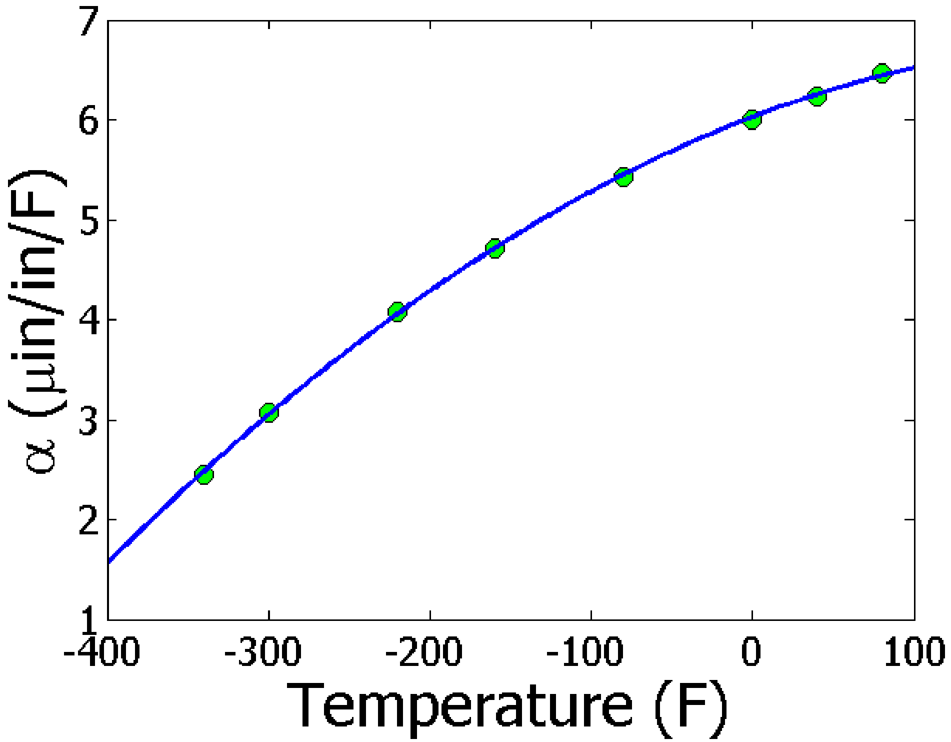


Estimating Contraction Accurately

Change in diameter (ΔD) by cooling it in dry ice/alcohol is given by

$$\Delta D = D\int_{T_a}^{T_c} \alpha(T)\,dT$$

$T_a = 80^{o}$F
$T_c = -108^{o}$F
$D = 12.363"$



$$\alpha = -1.2278\times10^{-5}T^2 + 6.1946\times10^{-3}T + 6.0150$$

$$\Delta D = -0.0137\,"$$

to small!!!

So what is the solution to the problem?

One solution is to immerse the trunnion in liquid nitrogen which has a boiling point of -321F as opposed to the dry-ice/alcohol temperature of -108F.

$$\Delta D = -0.0244\,''$$

Revisiting steps to solve a problem

1) Problem statement: trunnion stuck in the hub

2) Modeling: a new model

$$\Delta D = D\int_{T_a}^{T_c} \alpha(T)\,dT$$

3) Solution: a) trapezoidal rule or b) regression and integration.

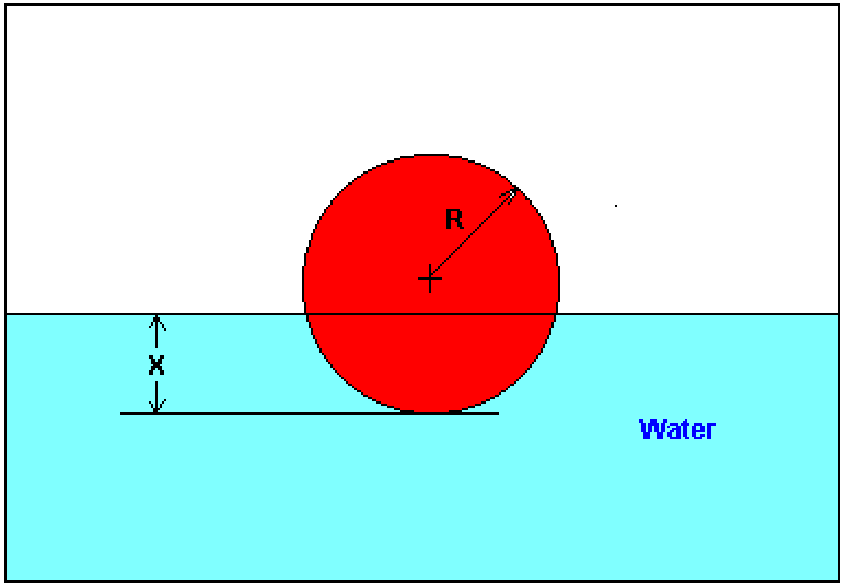
4) Implementation: cool the trunnion in liquid nitrogen.

Mathematical Procedures

- Nonlinear Equations
- Differentiation
- Simultaneous Linear Equations
- Curve Fitting
  - Interpolation
  - Regression
- Integration
- Ordinary Differential Equations
- Other Advanced Mathematical Procedures:
  - Partial Differential Equations
  - Optimization
  - Fast Fourier Transforms

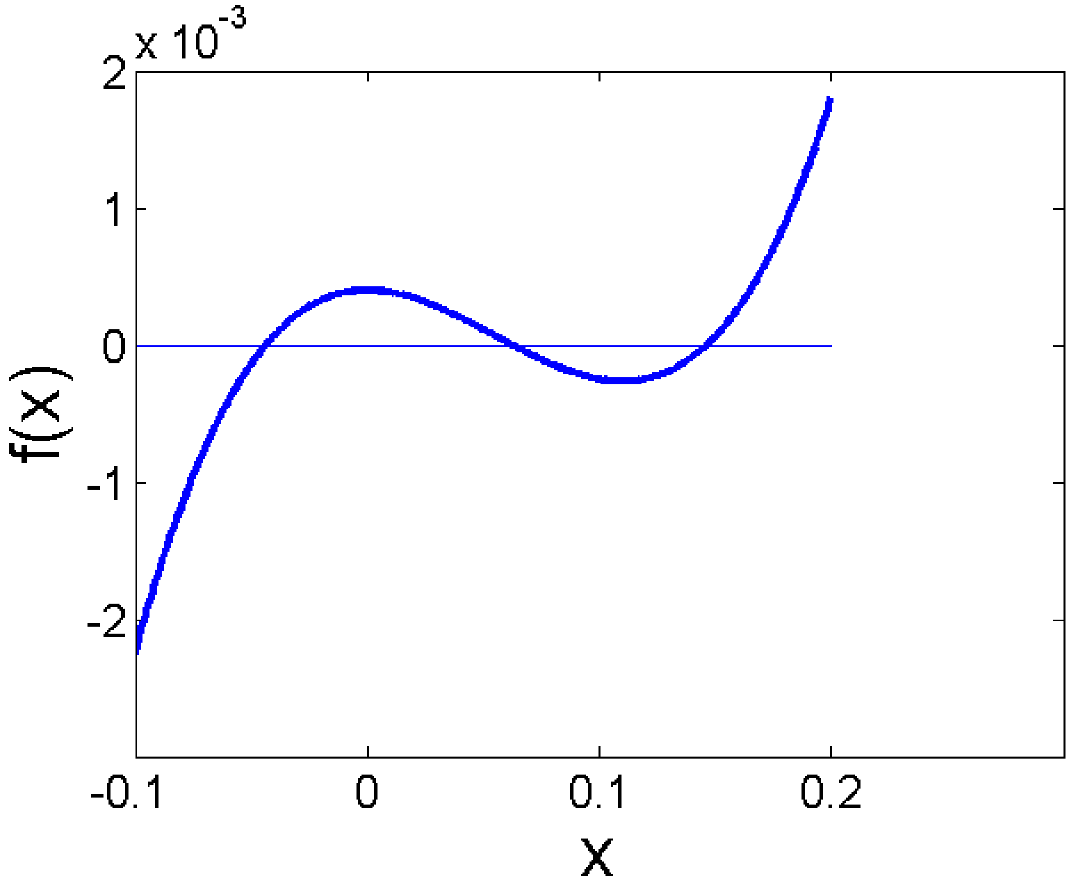
Nonlinear Equations

How much of the floating ball is under water?

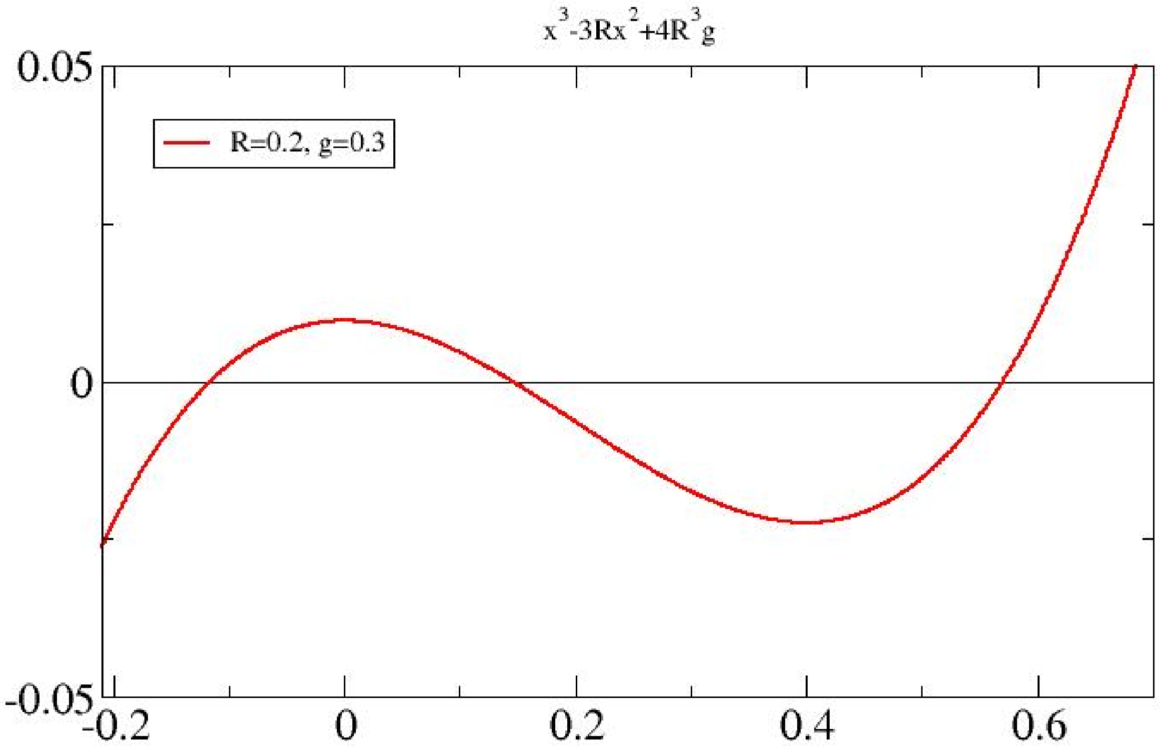


2R=0.11m

$$x^3 - 0.165x^2 + 3.993 \times 10^{-4} = 0$$

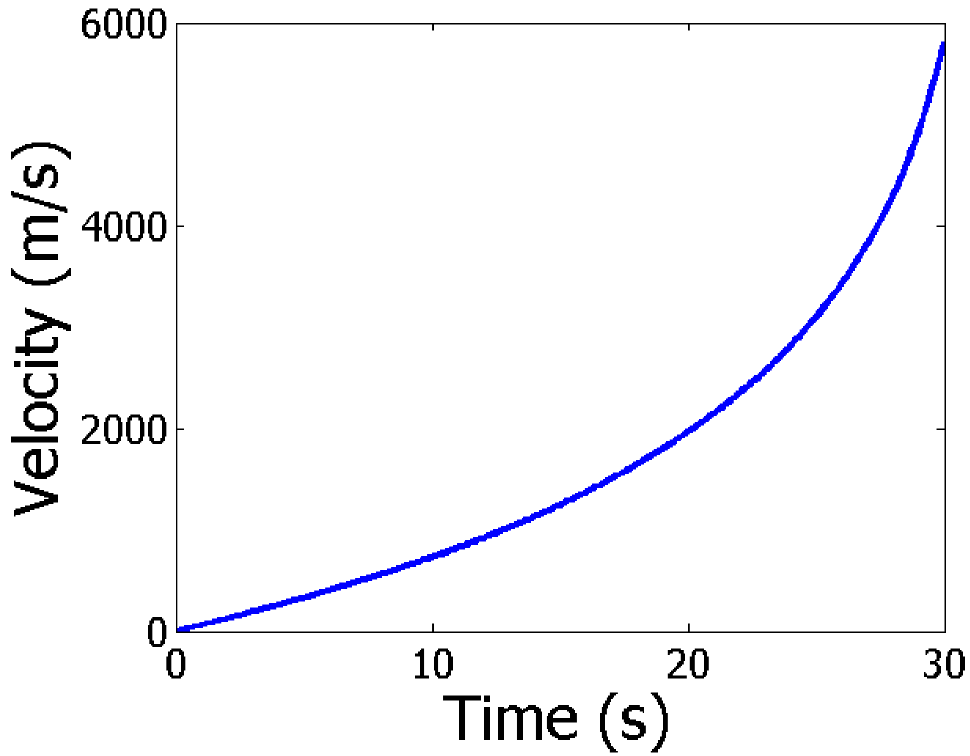


$$f(x) = x^3 - 0.165x^2 + 3.993 \times 10^{-4} = 0$$


x^3-3Rx^2+4R^3g
R=0.2, g=0.3
0.05
0
-0.05
-0.2
0
0.2
0.4
0.6

Differentiation

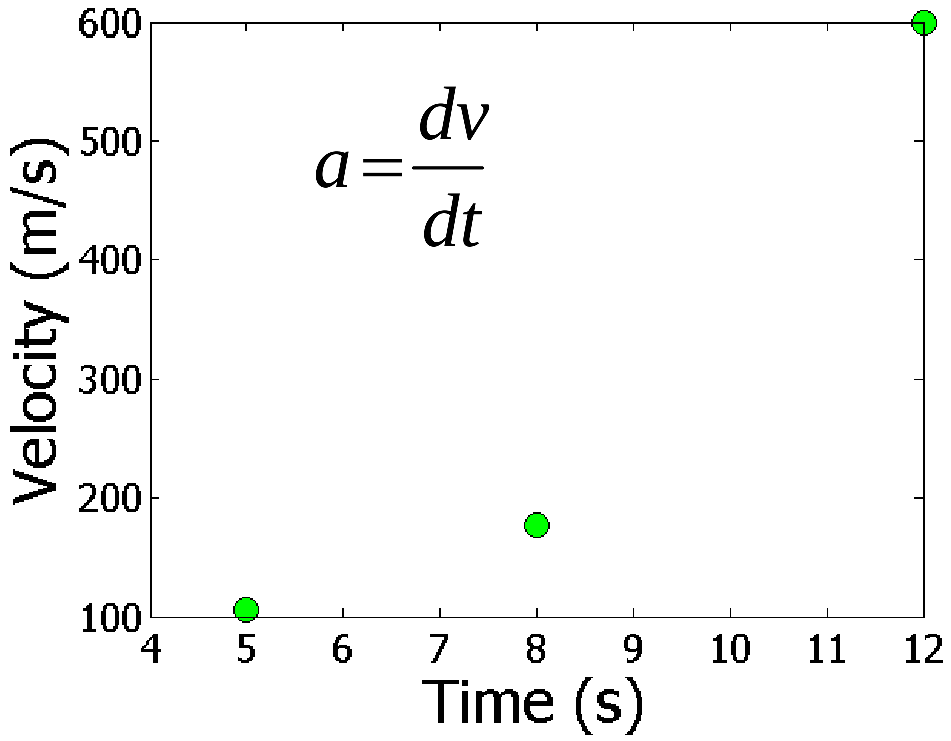
What is the acceleration at t=7 seconds?



$$v(t) = 2200 \ln\left(\frac{16\times10^4}{16\times10^4 - 5000t}\right) - 9.8t$$

$$a = \frac{dv}{dt}$$

Time (s)	5	8	12
V(m/s)	106	177	600



Simultaneous Linear Equations

Find the velocity profile, given:

Time (s)	5	8	12
V (m/s)	106	177	600

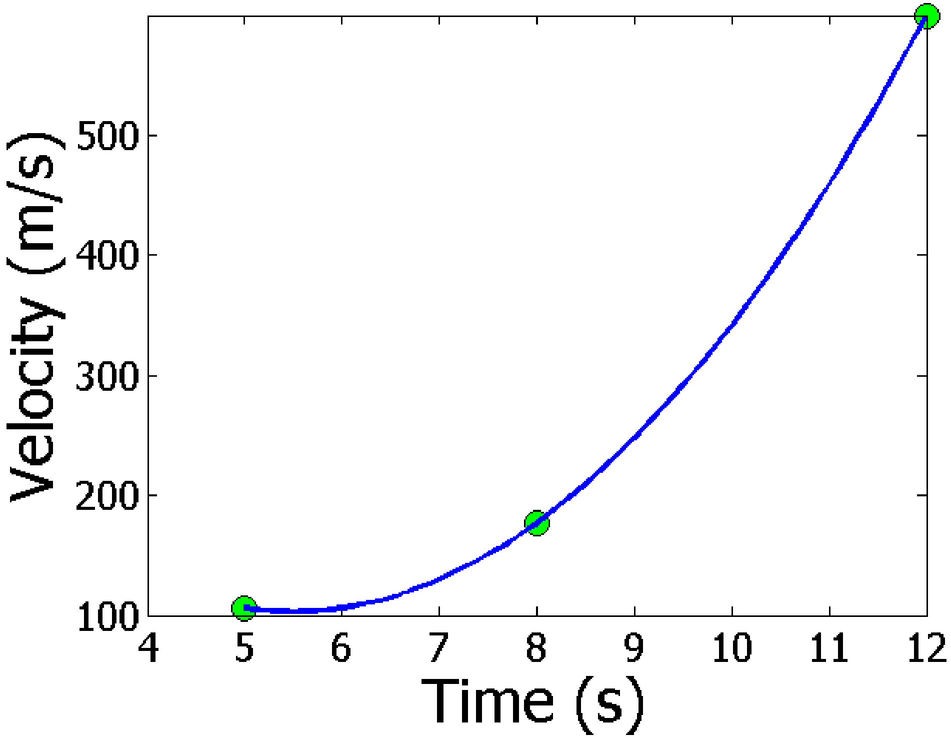
$$v(t) = at^2 + bt + c, \qquad 5 \le t \le 12$$

Three simultaneous linear equations:

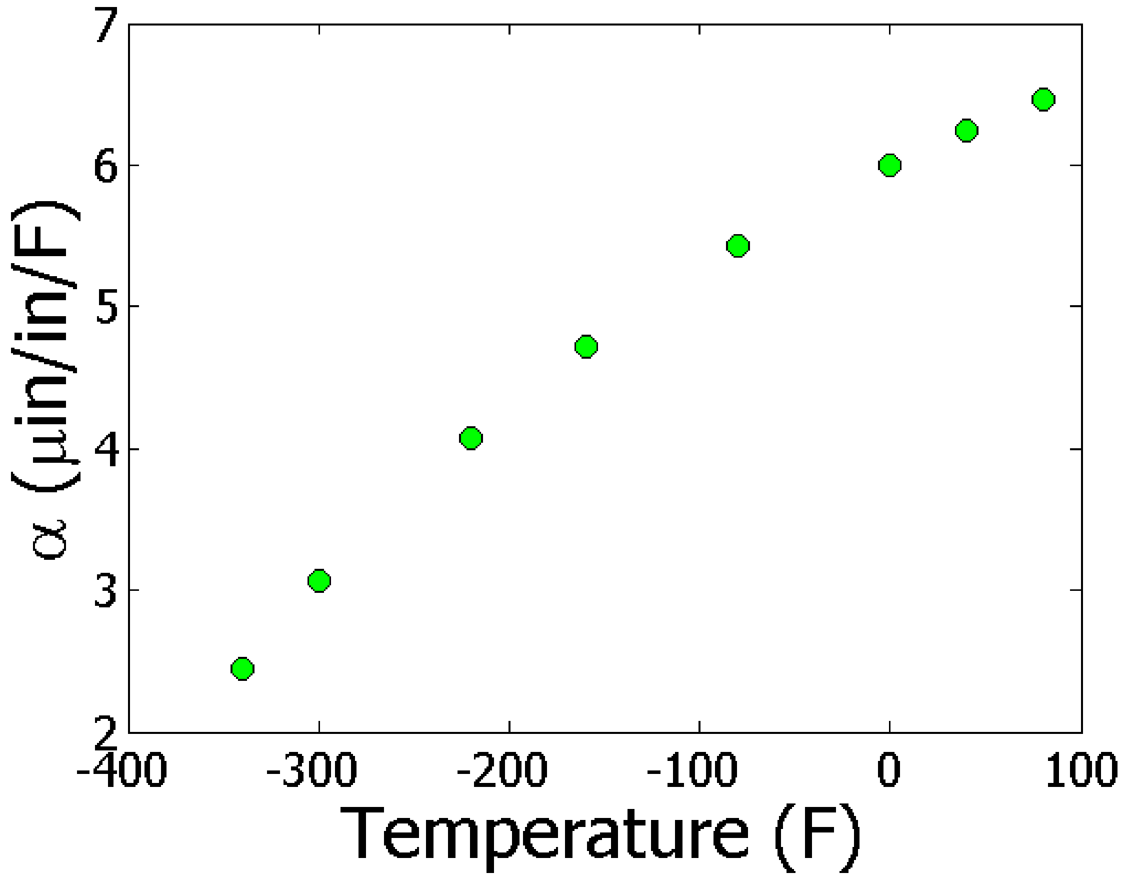
$$\begin{cases} 25a + 5b + c = 106 \\ 64a + 8b + c = 177 \\ 144a + 12b + c = 600 \end{cases}$$

What is the velocity of the rocket at t=7 s?

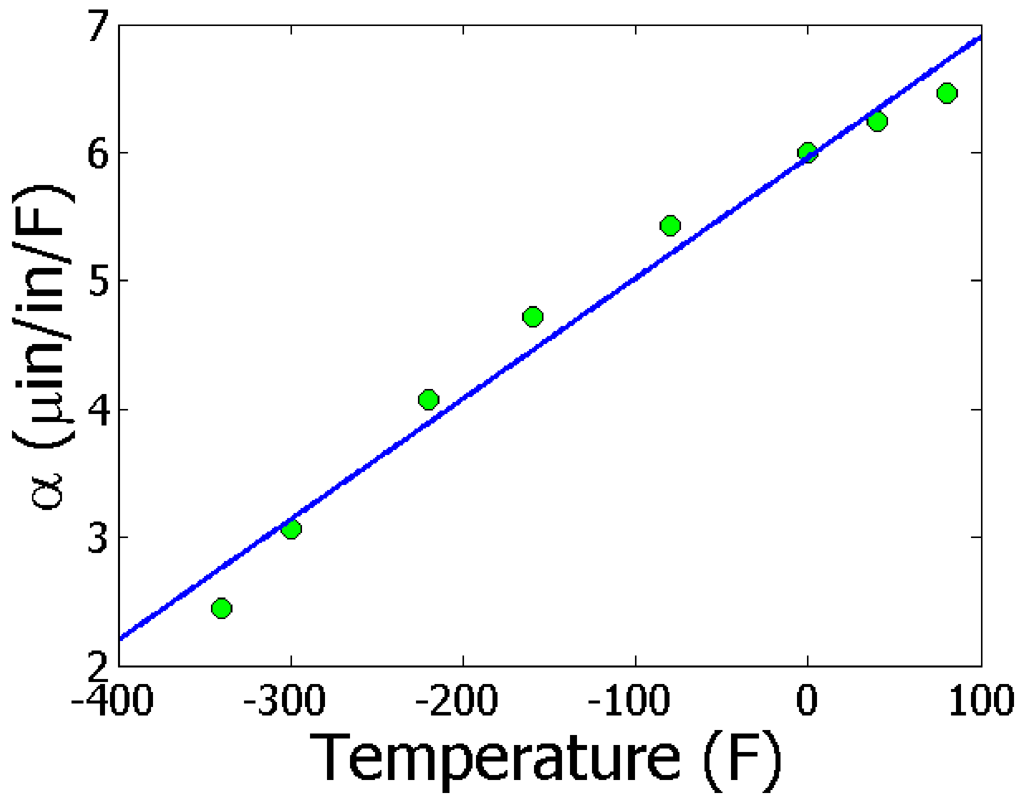
Time (s)	5	8	12
V (m/s)	106	177	600



Thermal expansion coefficient data for cast steel

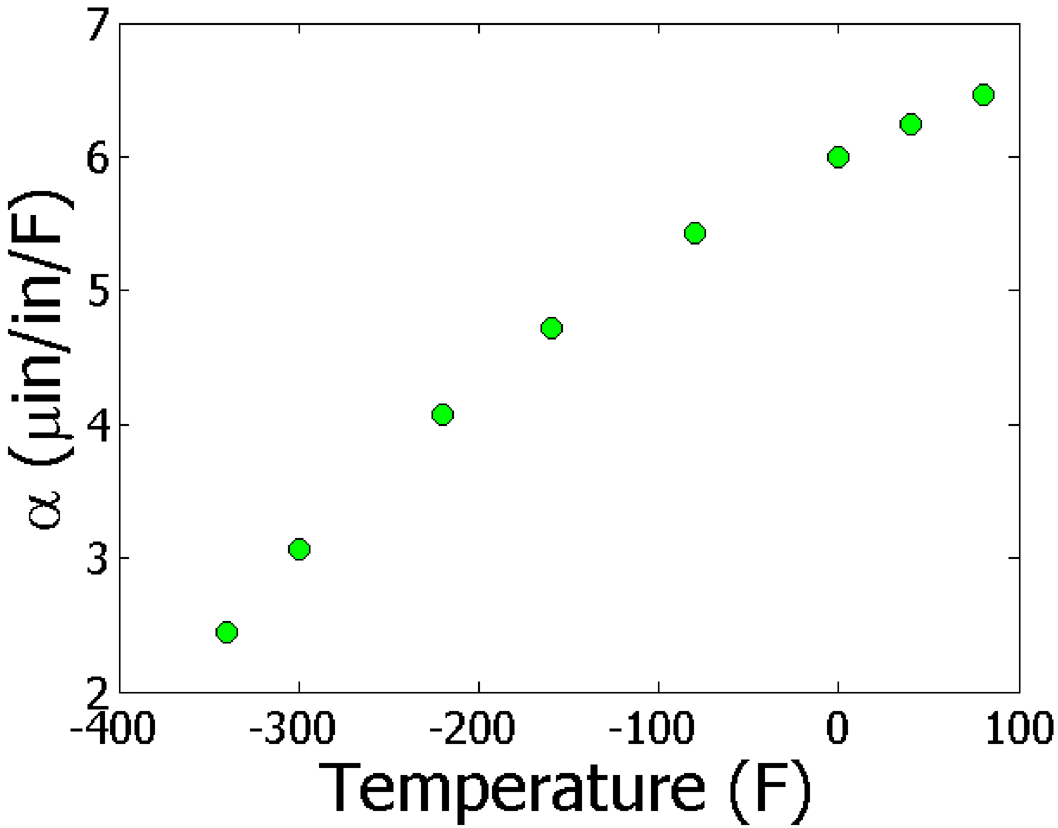


Regression

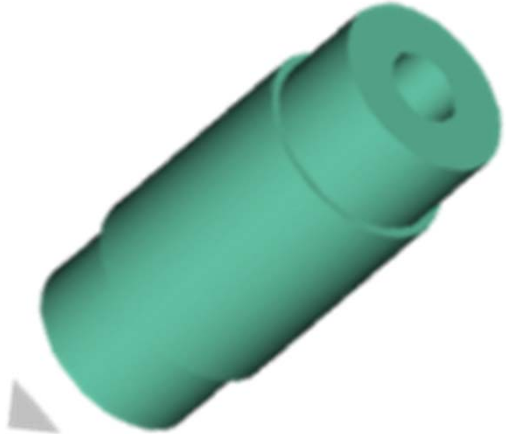


Integration

$$\Delta D = D \int_{T_{room}}^{T_{fluid}} \alpha \, dT$$



How long does it take a trunnion to cool down?



$$mc\frac{d\theta}{dt} = -hA(\theta - \theta_a), \quad \theta(0) = \theta_{room}$$

What you need to know to create your own computing algorithms?

- the size of your computer's memory
- the execution speed of arithmetic and logic operations
- the acceptable range of numbers during the calculations
- the accuracy of basic arithmetic operations performed on real numbers

Numbers representation in a computer memory

The numbers are stored as

- fixed-point numbers
- floating-point numbers

The computer works in the binary system, and communicates with the outside world in the decimal system, therefore conversion procedures are needed.

This is a source of errors.

How a Decimal Number is Represented

$$257.76 = 2\times10^{2} + 5\times10^{1} + 7\times10^{0} + 7\times10^{-1} + 6\times10^{-2}$$

Base 2

$$(1011.0011)_2 = \begin{pmatrix} (1\times2^{3} + 0\times2^{2} + 1\times2^{1} + 1\times2^{0}) \\ +(0\times2^{-1} + 0\times2^{-2} + 1\times2^{-3} + 1\times2^{-4}) \end{pmatrix}_{10}$$

$$= 11.1875$$

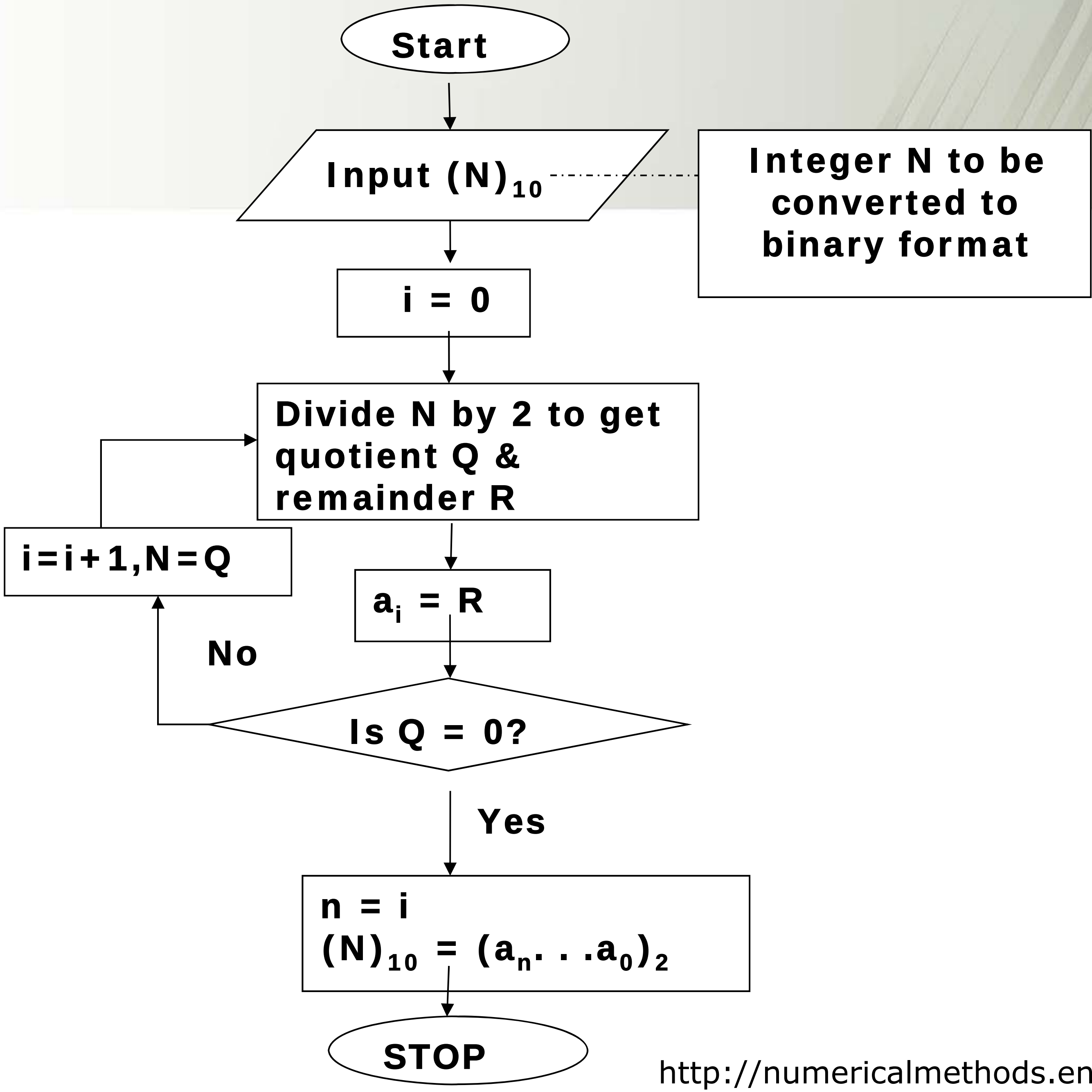
In the binary system we use two digits: 0 and 1, called **bits**

Convert base 10 integer to binary representation

	Quotient	Remainder
11/2	5	$1 = a_0$
5/2	2	$1 = a_1$
2/2	1	$0 = a_2$
1/2	0	$1 = a_3$

Hence

$$(11)_{10} = (a_3 a_2 a_1 a_0)_2$$
$$= (1011)_2$$

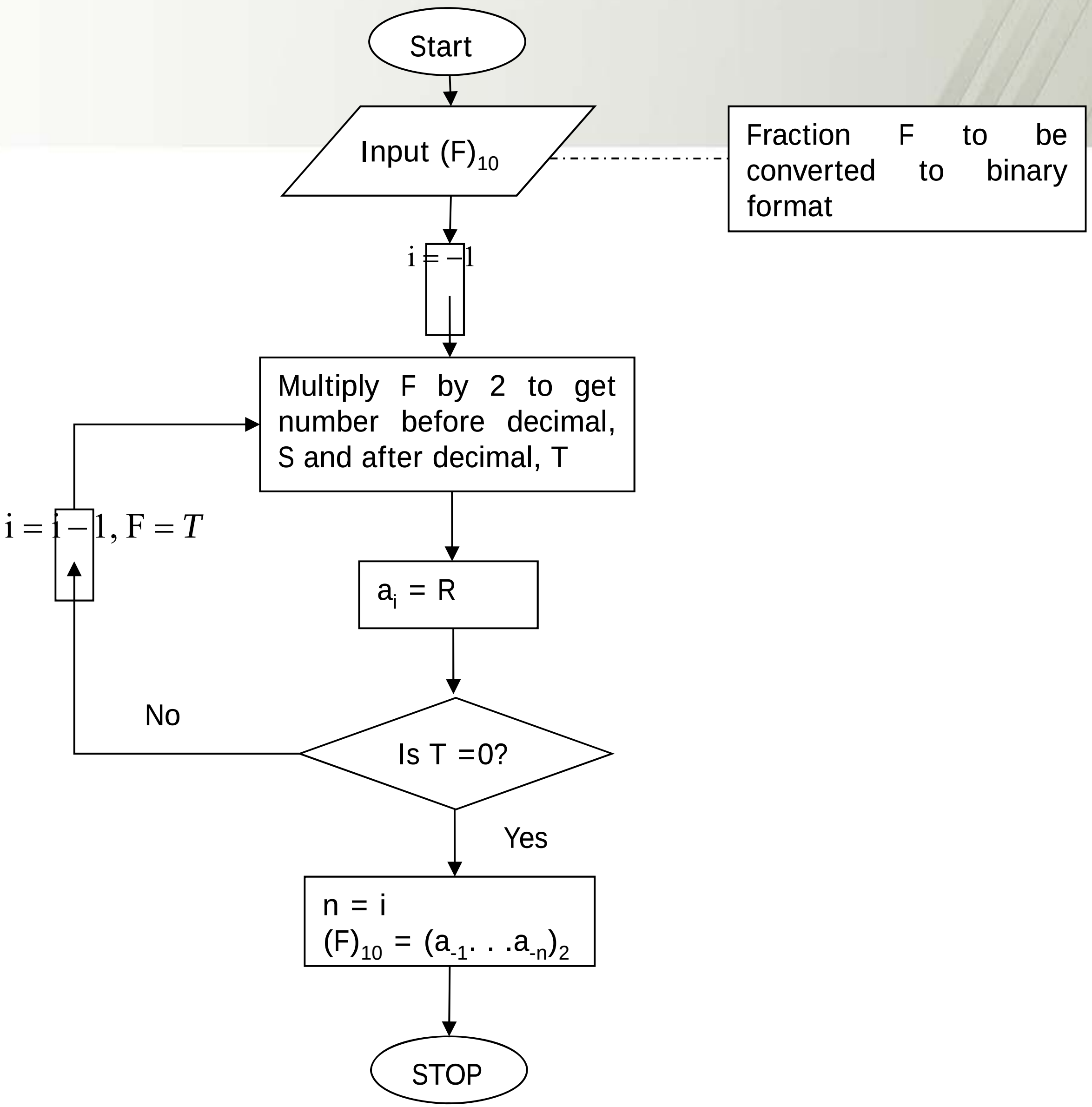


http://numericalmethods.eng.usf.edu

Converting a base-10 fraction to binary representation

	Number	Number after decimal	Number before decimal
0.1875×2	0.375	0.375	$0 = a_{-1}$
0.375×2	0.75	0.75	$0 = a_{-2}$
0.75×2	1.5	0.5	$1 = a_{-3}$
0.5×2	1.0	0.0	$1 = a_{-4}$

Hence $$(0.1875)_{10} = (a_{-1}a_{-2}a_{-3}a_{-4})_2$$
$$= (0.0011)_2$$



Start
Input $(F)_{10}$
Fraction F to be converted to binary format
$i = -1$
Multiply F by 2 to get number before decimal, S and after decimal, T
$a_i = R$
Is T =0?
No
$i = i - 1, F = T$
Yes
n = i
$(F)_{10} = (a_{-1}. . .a_{-n})_2$
STOP

Decimal Number to Binary

$$(11.1875)_{10} = (\quad ?.? \quad)_2$$

Since

$$(11)_{10} = (1011)_2$$

and

$$(0.1875)_{10} = (0.0011)_2$$

we have

$$(11.1875)_{10} = (1011.0011)_2$$

Different approach

$$(11.1875)_{10}$$

$$\begin{aligned}(11)_{10} &= 2^3 + 3 \\ &= 2^3 + 2^1 + 1 \\ &= 2^3 + 2^1 + 2^0 \\ &= 1\times 2^3 + 0\times 2^2 + 1\times 2^1 + 1\times 2^0 \\ &= (1011)_2\end{aligned}$$

Different approach

$$(0.1875)_{10} = 2^{-3} + 0.0625$$
$$= 2^{-3} + 2^{-4}$$
$$= 0\times 2^{-1} + 0\times 2^{-2} + 1\times 2^{-3} + 1\times 2^{-4}$$
$$= (.0011)_2$$

$$(11.1875)_{10} = (1011.0011)_2$$

The problem of accuracy

Example: Not all fractional decimal numbers can be represented exactly

	Number	Number after decimal	Number before decimal
0.3×2	0.6	0.6	$0=a_{-1}$
0.6×2	1.2	0.2	$1=a_{-2}$
0.2×2	0.4	0.4	$0=a_{-3}$
0.4×2	0.8	0.8	$0=a_{-4}$
0.8×2	1.6	0.6	$1=a_{-5}$

$$(0.3)_{10} \approx (a_{-1}a_{-2}a_{-3}a_{-4}a_{-5})_2 = (0.01001)_2 = 0.28125$$

The accuracy depends on the computer words length. Rounding off and chopping off lead to errors.

Floating Point Representation

$$x = M \times N^{w}$$

M - mantissa

W - exponent

N=2, 10

The floating point number is represented by two groups of bits:

I – mantissa M, fractional part

II - exponent W , an integer, W determines the range of the numbers represented in the computer

Floating Point Representation

Example:

If in binary representation M defines 5 bits and W defines 3 bits, the first bit represents the sign of a number ("-" is 1), then:

$$x = (1)1101 \quad (0)10$$

$$x = M \times N^{w} \qquad\qquad \text{M} \qquad \text{W}$$

$$x = -0{,}1101 \times 2^{+10}$$

$$x = -\left(\frac{1}{2} + \frac{1}{4} + \frac{0}{8} + \frac{1}{16}\right) \times 2^{+(1\cdot 2 + 0\cdot 1)}$$

in the decimal representation -3,25

Floating Point Representation

$$x = M \times N^{w}$$

In this notation, only certain positive number in the range from 0.0625 to 7.5, negative numbers from -0.0625 to -7.5 and the number 0 can be represented

There are some numbers that cannot be expressed

Number of x = 0.2 (in decimal) in binary notation has an infinite expansion:

$$x = 0{,}0011(0011)$$

The nearest number (for M = 5 and W = 3) is $x = 0{,}001100$

we have 0,1875

This is a source of input errors

Fixed-point representation

weights		2^n	...	2^2	2^1	2^0
bits	s	b_n	...	b_2	b_1	b_0
	sign	the absolute value of the number				

The fixed-point representation of the number allocated to n+2 bits (1 bit for the sign and n +1 bits for the absolute value of the number) has the following structure:

$$number = s \cdot \sum_{k=0}^{n} b_k 2^k$$

where:

s=1 or s=-1 (the sign of the number)

b_k takes the value 0 or 1 (the absolute value of the number)

Fixed-point representation

weights		2^n	...	2^2	2^1	2^0
bits	s	b_n	...	b_2	b_1	b_0
	sign	the absolute value of the number				

On n+2 bits, integers can be stored in the following range:

$$[-2^{n+1}+1;2^{n+1}-1]$$

Fixed numbers are a subset of the integers.

Overflow?

Fixed-point representation

weights		2^n	...	2^2	2^1	2^0
bits	s	b_n	...	b_2	b_1	b_0
	sign	the absolute value of the number				

High-level programming languages offer several types of fixed-point numbers:

Integer - 16 bits

LongInt – 32 bits

ShortInt – 8 bits