Lab - 3 Computational Physics I - Physics 381

Guilherme Contesini , 10140201
Gerswin Magat , 00325287

March 11, 2014

Abstract

A Case Study: Patriot Missile System
On February 25, 1991, a Patriot missile defense system operating at Dhahran, Saudi Arabia, during Operation Desert Storm failed to track and intercept an incoming Scud. This Scud subsequently hit an Army barracks, killing 28 Americans. This report responds to your request that we review the facts associated with this incident and determine if a computer software problem was involved.

...The Patriot battery at Dhahran failed to track and intercept the Scud missile because of a software problem in the system's weapons control computer. This problem led to an inaccurate tracking calculation that became worse the longer the system operated. At the time of the incident, the battery had been operating continuously for over 100 hours. By then, the inaccuracy was serious enough to cause the system to look in the wrong place for the incoming Scud. Two weeks before the incident, Army officials received Israeli data indicating some loss in accuracy after the system had been running for 8 consecutive hours. Consequently, Army officials modified the software to improve the system's accuracy. However, the modified software did not reach Dhahran until February 26, 1991–the day after the Scud incident.

- An excerpt from a report by the United States General Accounting Office, Information Management and Technology Division, in Washington, D.C., February 4, 1992

1 Introduction:

Computers take values by means of binary codes. This means that information is not held in values of base ten, but by values of base 2. A binary number consists only of a 0 or a 1. For example, the integer 5 is represented by the number '101' in binary. to translate a number of base 10 to a binary, we would simply divide by 2 and write down the remainder value in sequence including zeroes. to get a base-10 number from a binary, the opposite procedure is the correct method. Each number in sequence from right to left would correspond to 2^0, 2^1, 2^2, and so on. Taking the previous example, this would look like:

$5_{10} = 101_2 = [(1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0)]_{10}$
$5_{10} = 101_2 = [4 + 0 + 1]_{10}$
$5_{10} = 101_2 = 5_{10}$

Similar to the method of computing binary vs. base-10 values, computers also interpret values differently than humans. There are methods that computers do in order to contain the information in an "efficient" way. What we must keep in mind is that computers allocate a limited amount of memory for any value unless over-ridden by the user. These limitations by the computers can cause issues in terms of precision and accuracy.

The purpose of this lab is to examine the extent at which computers can take information correctly and what distortions could happen if we are not careful with information methods.

2 Part 2 : Loss of accuracy

2.1 i

$\epsilon =$		0.1	0.01	1.0×10^{-4}	1.0×10^{-8}	1.0×10^{-16}
$1-\sqrt{1-\epsilon}$	[4-byte]	$\underline{0.0513167}$	0.00501257	5.00083×10^{-5}	0.00000	0.00000
$1-\sqrt{1-\epsilon}$	[8-byte]	$\underline{0.0513167}$	$\underline{0.00501256}$	$\underline{5.00013\times10^{-5}}$	$\underline{5.00000\times10^{-9}}$	0.00000
$\frac{\epsilon}{1+\sqrt{1-\epsilon}}$	[4-byte]	$\underline{0.0513167}$	$\underline{0.00501256}$	$\underline{5.00013\times10^{-5}}$	$\underline{5.00000\times10^{-9}}$	$\underline{5.00000\times10^{-17}}$
$\frac{\epsilon}{2}$	[4-byte]	0.0500000	0.00500000	5.00000×10^{-5}	$\underline{5.00000\times10^{-9}}$	$\underline{5.00000\times10^{-17}}$
$\frac{\epsilon}{2}+\frac{\epsilon^2}{8}$	[4-byte]	0.0512500	0.00501250	$\underline{5.00013\times10^{-5}}$	$\underline{5.00000\times10^{-9}}$	$\underline{5.00000\times10^{-17}}$
$\frac{\epsilon}{2}+\frac{\epsilon^2}{8}+\frac{\epsilon^3}{16}$	[4-byte]	0.0513125	$\underline{0.00501256}$	$\underline{5.00013\times10^{-5}}$	$\underline{5.00000\times10^{-9}}$	$\underline{5.00000\times10^{-17}}$

Table 1: Extracted Data with Calculated Values

2.2 ii

$\epsilon =$		0.1	0.01	1.0×10^{-4}	1.0×10^{-8}	1.0×10^{-16}
$1-\sqrt{1-\epsilon}$	[4-byte]	5.13167×10^{-2}	5.01257×10^{-3}	5.000829×10^{-5}	0.000000	0.00000
$1-\sqrt{1-\epsilon}$	[8-byte]	5.13167×10^{-2}	5.01256×10^{-3}	5.000124×10^{-5}	4.99999×10^{-9}	0.00000
$\frac{\epsilon}{1+\sqrt{1-\epsilon}}$	[4-byte]	5.13167×10^{-2}	5.01256×10^{-3}	5.000125×10^{-5}	4.99999×10^{-9}	5.00000×10^{-17}
$\frac{\epsilon}{2}$	[4-byte]	5.00000×10^{-2}	4.99999×10^{-3}	4.999999×10^{-5}	4.99999×10^{-9}	5.00000×10^{-17}
$\frac{\epsilon}{2}+\frac{\epsilon^2}{8}$	[4-byte]	5.12499×10^{-2}	5.01250×10^{-3}	5.000125×10^{-5}	4.99999×10^{-9}	5.00000×10^{-17}
$\frac{\epsilon}{2}+\frac{\epsilon^2}{8}+\frac{\epsilon^3}{16}$	[4-byte]	5.13125×10^{-2}	5.01256×10^{-3}	5.000125×10^{-5}	4.99999×10^{-9}	5.00000×10^{-17}

Table 2: Data produced

2.3 iii

The variable uses more memory can store a larger number of digits resulting in improved accuracy for that number. Consequences of this difference is the greater precision in calculations that have operations between very next numbers.

3 Conclusion :

From the results of the lab, we can gather that accuracy and precision are both vital to receive usable results. The example in class that Dr. Ouyed mentioned was a catastrophic event where American soldiers lost their lives due to the carelessness of the scientists who overlooked the implications of small errors over millions and millions of loops. This shows that these calculations are very important and must be taken seriously, even the error propagations and calculations!

4 Appendix:

4.1 Fortran Codes:

```
program x1
implicit none

real :: x1_1 , eps_1
real*8 :: x1_2 , eps_2

eps_2 = 1e-16
open(12,file="1d-16.dat")
eps_1=eps_2
x1_1=1-(1-eps_1)**(0.5)
x1_2=1-(1-eps_2)**(0.5)

write(12,*) eps_1
write(12,*)'[4-byte]',
x1_1
write(12,*)'[8-byte]',
x1_2
write(12,*)'[4-byte]',
eps_1/(1+(1-eps_1)**(0.5))
write(12,*)'[4-byte]',
(eps_1/2)
write(12,*)'[4-byte]',
(eps_1/2)+((eps_1**2)/8)
write(12,*)'[4-byte]',
(eps_1/2)+((eps_1**2)/8)+((eps_1**3)/16)
write(12,*)
write(12,*)

end program
```