Some disasters caused by numerical errors

Patriot Missile Failure

On February 25, 1991, during the Gulf War, an American Patriot Missile battery in Dharan, Saudi Arabia, failed to intercept an incoming Iraqi Scud missile. The Scud struck an American Army barracks and killed 28 soldiers. A report of the General Accounting office, GAO/IMTEC-92-26, entitled *Patriot Missile Defense: Software Problem Led to System Failure at Dhahran, Saudi Arabia*
reported on the cause of the failure. It turns out that the cause was an inaccurate calculation of the time since boot due to computer arithmetic errors. Specifically, the time in tenths of second as measured by the system's internal clock was multiplied by 1/10 to produce the time in seconds. This calculation was performed using a 24 bit fixed point register. In particular, the value 1/10, which has a non-terminating binary expansion, was chopped at 24 bits after the radix point. The small chopping error, when multiplied by the large number giving the time in tenths of a second, lead to a significant error. Indeed, the Patriot battery had been up around 100 hours, and an easy calculation shows that the resulting time error due to the magnified chopping error was about 0.34 seconds. (The number 1/10 equals $1/2^4+1/2^5+1/2^8+1/2^9+1/2^{12}+1/2^{13}+\ldots$. In other words, the binary expansion of 1/10 is 0.0001100110011001100110011001100.... Now the 24 bit register in the Patriot stored instead 0.000110011001100110011001100 introducing an error of 0.0000000000000000000011001100... binary, or about 0.000000095 decimal. Multiplying by the number of tenths of a second in 100 hours gives $0.000000095\times100\times60\times60\times10=0.34$.) A Scud travels at about 1,676 meters per second, and so travels more than half a kilometer in this time. This was far enough that the incoming Scud was outside the "range gate" that the Patriot

tracked. Ironically, the fact that the bad time calculation had been improved in some parts of the code, but not all, contributed to the problem, since it meant that the inaccuracies did not cancel.

The following paragraph is excerpted from the GAO report.

The range gate's prediction of where the Scud will next appear is a function of the Scud's known velocity and the time of the last radar detection. Velocity is a real number that can be expressed as a whole number and a decimal (e.g., 3750.2563...miles per hour). Time is kept continuously by the system's internal clock in tenths of seconds but is expressed as an integer or whole number (e.g., 32, 33, 34...). The longer the system has been running, the larger the number representing time. To predict where the Scud will next appear, both time and velocity must be expressed as real numbers. Because of the way the Patriot computer performs its calculations and the fact that its registers are only 24 bits long, the conversion of time from an integer to a real number cannot be any more precise than 24 bits. This conversion results in a loss of precision causing a less accurate time calculation. The effect of this inaccuracy on the range gate's calculation is directly proportional to the target's velocity and the length of the the system has been running. Consequently, performing the conversion after the Patriot has been running continuously for extended periods causes the range gate to shift away from the center of the target, making it less likely that the target, in this case a Scud, will be successfully intercepted.

This description is adapted from The Patriot Missile Failure by Douglas N. Arnold.

 Nederlands

Explosion of the Ariane 5

On June 4, 1996 an unmanned Ariane 5 rocket launched by the European Space Agency exploded just forty seconds after lift-off (918K QuickTime movie). The rocket was on its first voyage, after a decade of development costing $7 billion. The destroyed rocket and its cargo were valued at $500 million. A board of inquiry investigated the causes of the explosion and in two weeks issued a report. It turned out that the cause of the failure was a software error in the inertial reference system. Specifically a 64 bit floating point number relating to the horizontal velocity of the rocket with respect to the platform was converted to a 16 bit signed integer. The number was larger than 32,768, the largest integer storeable in a 16 bit signed integer, and thus the conversion failed.

The report of the Inquiry Board is available. The following paragraphs are extracted from that report.

On 4 June 1996, the maiden flight of the Ariane 5 launcher ended in a failure. Only about 40 seconds after initiation of the flight sequence, at an altitude of about 3700 m, the launcher veered off its flight path, broke up and exploded.

The failure of the Ariane 501 was caused by the complete loss of guidance and attitude information 37 seconds after start of the main engine ignition sequence (30 seconds after lift-off). This loss of information was due to specification and design errors in the software of the inertial reference system.

The internal SRI software exception was caused during execution of a data conversion from 64-bit floating point to 16-bit signed integer value. The floating point number which was converted had a value greater than what could be represented by a 16-bit signed integer.*

*SRI stands for Système de Référence Inertielle or Inertial Reference System.

This description is adapted from The Explosion of the Ariane 5 by Douglas N. Arnold.

An interesting article
on the accident and its implications by James Gleick appeared in The New York Times Magazine of 1 December 1996.

EURO page: Conversion Arithmetics

The introduction of the euro and the rounding of currency amounts

For each old local currency a conversion factor to EURO will be fixed. Each factor will contain six significant digits. The number of decimal digits depends on the number of digits before the decimal point. For example: the conversion factor for BEF will have two digits before the decimal points, and therefore four behind. For NLG this will be 1 digit before and 5 digits after.

There will be no conversion factor for converting EURO to old local currency.

The conversion will go as follows:

- if EURO must be converted to local currency, then MULTIPLY the amount in EURO using the factor for that currency. Then round the result according to the number of decimal digits appropriate for the currency. For exemple: BEF: no decimal digits; NLG, DEM, FRF: 2 decimal digits.
- If an old local currency must be converted to EURO, then DIVIDE the amount in local currency using the factor for that currency. Then round the result to 2 decimal digits.
- If a local currency must be converted to another local currency, then perform a two-step conversion: to and from EURO. The current directive stipulates that the division must be done first and the multiplication must be done afterwards. The intermediary result must contain *at least* (how imprecise!) three decimal digits. Perhaps this will be further refined.

Arithmetic errors

There are three arithmetic errors: conversion errors, reconversion errors and totalising errors. These errors cannot be avoided, but the conseqences can be brought down to a minimum.

Conversion errors

When an amount is being converted, then there can be a rounding error which can be at most half ot the value of the smallest unit or subdivision of unit of the target currency. A conversion of BEF to EUR can yield a rounding error of at most 0,005 EUR, since 0,01 EUR will be less then 1 BEF. This error is relatively important for small amounts and

negligible for large amounts. Because of the statistical *normal distribution* of the errors, generally spoken the positive and negative differences will level. However, if the same amount, say 1 BEF must be consistently converted, then the error is a considerable one...

The next tables show how high the error margin can become: the smaller the amount to be converted, the higher the relative error will be:

From BEF to EUR, factor: 1 EUR = 39,5225 BEF

BEF	quasi-exact amount in EUR	EUR after rounding	Difference	Percentage of difference
1	0,025302043	0,03	0,004698	16%
2	0,050604086	0,05	-0,0006	-1%
3	0,075906129	0,08	0,004094	5%
4	0,101208173	0,10	-0.00121	-1%
5	0,126510216	0,13	0,00349	3%

From EUR to BEF, factor: 1 EUR = 39,5225 BEF

EUR	quasi-exact amount in BEF	BEF after rounding	Difference	Percentage of difference
0,01	0,395225	0	-0,39523	-100%
0,02	0,79045	1	0,20955	27%
0,03	1,185675	1	-0,18568	-16%
0,04	1,5809	2	0,4191	27%
0,05	1,976125	2	0,023875	1%
0,06	2,37135	2	-0,37135	-16%
0,07	2,766575	3	0,233425	8%

This problem can be avoided by adding in the original currency and convert only after the total is made, if it is at all possible in the specific situation. For the owner of a vendor machine, that sells huge quantities of low-value products, this phenomenon is of the utmost importance: it can eat a considerable part of the trade margin or add to it.

Determining the maximum possible rounding error

Use the following formulas:

- National Currency Unit to EUR and back: 0.005 EUR x conversion factor NCU/EUR
  Example: 1 EUR = ITL 1906.48: 10 ITL
- EUR to National Currency Unit and back: half the smallest (sub)unit / conversion factor NCU/EUR
  Example:: 1 EUR = BEF 39.7191: 0,01 EUR

Rounding: who makes the profit?

Source: Het vergiftigd geschenk van de afrondingen, article in the EURO-annexe of the Financieel-Economische Tijd edition 23.11.1997, supplemented with own comments

When we looking at the rounding algorithm, one might be inclined to think that the result will be more rounding up then there will be rouding down. This is because the digit that is used to determine the rounding can assume 5 values for which rounding-up will be applied (5, 6, 7, 8 and 9) whileas it can assume only 4 values for which rounding-down will be allied (1, 2, 3 and 4). This means that in 5 cases out of 10, the payor will pay 'more' than he's due and therefore lose a bit of money, whileas only in 4 cases out of 10 he will pay less and therefore gain a bit. Only in one case out of 10 (0) there is neither gain or loss. The French National Council for the Consumer has, according to the article, calculated that due to this rounding process, all giant supermarkets in France together will gain about

20 million BEF (0,5 million EUR) extra.

Is this reasoning correct? In my view it is not.

If the conversion is performed correctly, then it should not take into account the digits following the digit that is used to determine the rounding operation. Since EUR is rounded to the second decimal digit, this means that the third decimal digit is the one that determines the rounding operation. The fourth and all subsequent digits should not be taken into account.

Let's assume a factor 1 EUR = 6.60067 FRF. When we convert 130.00 FRF to EUR, the intermediate result before rounding is about 19,69497035907 EUR. We see that the third digit after the decimal sign is 4; the fourth is 9. This fourth digit after the decimal position, and all subsequent digits are irrelevant for the rounding process. Since the third digit is 4, we round down. The result therefore is 19.69 EUR.

It is not correct to say that there is no rounding if the third decimal digit is a 0. When we use the same conversion factor to convert 115 FRF to EUR, then the intermediate result before rounding is 17,27092167637 EUR, or rounded it is 17,27 EUR. In this case, the payor pays less, and therefore gains, the payee receives less, and therefore looses.

This means that in cases where the third and fourth digit behind the decimal point has a value between 00 en 49 we round down, and in cases where the third and fourth digit is between 50 en 99 we round up. So it's more like a 50-50 chance.

Note that in order to determine the third digit after the decimal sign, rounding is NOT allowed. If the result of the division or multiplication is 1,234567, then it is not allowed to round to the third digit (1,235), and then round once more to become the final result (1,24)! When performing the division or multiplication, the intermediate result before rounding should have 3 decimal digits; subsequent digits must be truncated, not rounded.

Reconversion errors

Source: Belgian Bankers Association

If we perform a conversion of an amount denominated in the currency with the lowest intrinsic value of one (sub)unit to the other currency, and then we convert the result back to the original currency, then a slightly different amount may result.
If we perform a conversion of an amount denominated in the currency with the highest intrinsic value value of one (sub)unit to the other currency, and then we convert the result back to the original currency, then the same amount will result.

An example: conversion between BEF and EURO: the intrinsic value of the smallest (sub)unit of BEF, being 1 BEF is greater than the intrinsic value of the smallest (sub)unit of EUR, being 0,01 EUR. This means that:

- conversion from BEF to EUR and reconversion back to EUR will always yield the original amount in BEF;
- conversion from EUR to BEF and reconversoin back to EUR may yield a slightly different result.

The following example is taken from the brochure *De Euro, de munt van morgen* of the Generale Bank

From EUR to BEF and back using conversion factor 1 EUR = 38,45 BEF
101 EUR x 38.45 BEF/EUR = 3883,45 BEF; rounded 3883 BEF
3883 BEF / 38.45 BEF/EURO = 100.99 EUR. 0,01 EUR is missing

For other currencies this problem can be just the other way around: the intrinsic value of a Guilder Cent (NLG 0,01) is less then the intrinsic value of a Eurocent (EUR 0,01).

Totalising errors

When amount are to be added to or subtracted from a total, then totaliserings- of sommatiefouten can ontstaan. Dur to rounding errors the total amount can vary, depending on the precedence of the sommatie and the conversion.

As an example I use an invoice with a net amount of 1000 BEF and a VAT rate of 21%; 1 EURO is 39.9125 BEF:

	BEF	EURO
Net amount	1000	25,05
VAT 21%	210	5,26
Total	1210	30,32 of 30,31

If we add the converted detailed amounts, we get a total of 30,31 EURO; when we convert the total amount in BEF to an amount in EUR, we get 30,32 EURO! The difference of 0,01 EURO is called a *Totalising Error*.

The Vancouver Stock Exchange

In 1982 the Vancouver Stock Exchange instituted a new index initialized to a value of 1000.000. The index was updated after each transaction. Twenty two months later it had fallen to 520. The cause was that the updated value was truncated rather than rounded. The rounded calculation gave a value of 1098.892.

Sources

1. The Wall Street Journal November 8, 1983, p.37.
2. The Toronto Star, November 19, 1983.
3. B.D. McCullough and H.D. Vinod
   Journal of Economic Literature
   Vol XXXVII (June 1999), pp. 633-665.

Rounding error changes Parliament makeup

Debora Weber-Wulff, 7 Apr 1992

We experienced a shattering computer error during a German election this past Sunday (5 April). The elections to the parliament for the state of Schleswig-Holstein were affected. German elections are quite complicated to calculate. First, there is the 5% clause: no party with less than 5% of the vote may be seated in parliament. All the votes for this party are lost. Seats are distributed by direct vote and by list. All persons winning a precinct vote (i.e. having more votes than any other candidate in the precinct) are seated. Then a complicated system (often D'Hondt, now they have newer systems) is invoked that seats persons from the party lists according to the proportion of the votes for each party. Often quite a number of extra seats (and office space and salaries) are necessary so that the seat distribution reflects the vote percentages each party got.

On Sunday the votes were being counted, and it looked like the Green party was hanging on by their teeth to a vote percentage of exactly 5%. This meant that the Social Democrats (SPD) could not have anyone from their list seated, which was most unfortunate, as the candidate for minister president was number one on the list, and the SPD won all precincts: no extra seats needed.

After midnight (and after the election results were published) someone discovered that the Greens actually only had 4,97% of the vote. The program that prints out the percentages only uses one place after the decimal, and had *rounded the count up* to 5%! This software had been used for *years*, and no one had thought to turn off the

rounding at this very critical (and IMHO very undemocratic) region!

So 4,97% of the votes were thrown away, the seats were recalculated, the SPD got to seat one person from the list, and now have a one seat majority in the parliament. And the newspapers are clucking about the "computers" making such a mistake.

Debora Weber-Wulff, Institut fuer Informatik, Nestorstr. 8-9, D-W-1000 Berlin 31 dww@inf.fu-berlin.de +49 30 89691 124

Sources

1. Rounding error changes Parliament makeup
   Debora Weber-Wulff
   The Risks Digest, Volume 13, Issue 37, 1992

The sinking of the Sleipner A offshore platform

Nederlands

Excerpted from SINTEF, Civil and Environmental Engineering:

The Sleipner A platform produces oil and gas in the North Sea and is supported on the seabed at a water depth of 82 m. It is a Condeep type platform with a concrete gravity base structure consisting of 24 cells and with a total base area of 16 000 m^2. Four cells are elongated to shafts supporting the platform deck. The first concrete base structure for Sleipner A sprang a leak and sank under a controlled ballasting operation during preparation for deck mating in Gandsfjorden outside Stavanger, Norway on 23 August 1991.

Immediately after the accident, the owner of the platform, Statoil, a Norwegian oil company appointed an investigation group, and SINTEF was contracted to be the technical advisor for this group.

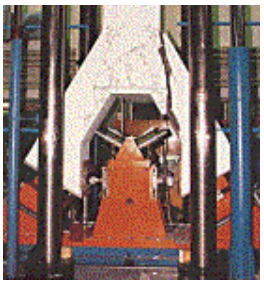
The investigation into the accident is described in 16 reports...

The conclusion of the investigation was that the loss was caused by a failure in a cell wall, resulting in a serious crack and a leakage that the pumps were not able to cope with. The wall failed as a result of a combination of a serious error in the finite element analysis and insufficient anchorage of the reinforcement in a critical zone.

A better idea of what was involved can be obtained from this photo and sketch of the platform. The top deck weighs

57,000 tons, and provides accommodation for about 200 people and support for drilling equipment weighing about 40,000 tons. When the first model sank in August 1991, the crash caused a seismic event registering 3.0 on the Richter scale, and left nothing but a pile of debris at 220m of depth. The failure involved a total economic loss of about $700 million.

The 24 cells and 4 shafts referred to above are shown to the left while at the sea surface. The cells are 12m in



diameter. The cell wall failure was traced to a tricell, a triangular concrete frame placed where the cells meet. At right one is pictured undergoing failure testing.

The post accident investigation traced the error to inaccurate finite element approximation of the linear elastic model of the tricell (using the popular finite element program NASTRAN). The shear stresses were underestimated by 47%, leading to insufficient design. In particular, certain concrete walls were not thick enough. More careful finite element analysis, made after the accident, predicted that failure would occur with this design at a depth of 62m, which matches well with the actual occurrence at 65m.

Further information can be found in the series of reports available for purchase from SINTEF and in an article from *Concrete International*, August 1997.

This description is adapted from The sinking of the Sleipner A offshore platform by Douglas N. Arnold.

Contact information:

Kees Vuik

Back to home page or educational page of Kees Vuik

Last modified on 19-02-2001 by Kees Vuik