Lecture 2

Preliminaries: Error, Floating Point

Owen L. Lewis

Department of Mathematics and Statistics
University of New Mexico

August 22, 2024

What we'll do:

- Error
- Convergence and Bit-Oh notation
- Floating point representation (beginning)

Errors

- How do I classify my method?
- Common case: we want to calculate f.
  - Begin by calculating $p_0 \approx f$.
  - Use p_0 to calculate $p_1 \approx f$ (better!).
  - Generate sequence $p_0, p_1, p_2 \ldots$.
  - Stop at some point which is "good enough".
- Goal 1: determine how the error $|f - p_n|$ behaves relative to n (and f).
- Goal 2: determine how the cost of computing p_n behaves relative to n (and f).

Errors

- Suppose we are prevented from using division, but we want to evaluate the function $f(x) = \frac{1}{1-x}$.
- This function has a very well known Taylor Series (more on these next week)

$$p_n(x) = \sum_{k=0}^{n} x^k = 1 + x + x^2 + \cdots + x^n \approx f(x).$$

- so

$$E_n = |f(x) - p_n(x)|$$

- Is $E_n \approx 1/n^r$?
- Is $E_n \approx 1/\sqrt{n}$?
- Is $E_n \approx 1/e^n$?

Aside: Absolute vs. Relative Error

- Here we used the absolute error:

$$\begin{aligned}\text{Error} &= |p_n - f| \\ &= |\text{true value} - \text{approximate value}|\end{aligned}$$

- This doesn't always tell the whole story. For example, if the values are large, like billions, then an Error of 100 is small. If the values are smaller, say around 10, then an Error of 100 is large. We need the relative error:

$$\begin{aligned}\text{Rel} &= \left|\frac{p_n - f}{f}\right| \\ &= \left|\frac{\text{true value} - \text{approximate value}}{\text{true value}}\right|\end{aligned}$$

- Both are important in different contexts. Always keep both in mind.

Big-O

How to measure the impact of n on error or algorithmic cost?

$\mathcal{O}(\cdot)$

Let $g(n)$ be a function of n. Then define

$$\mathcal{O}(g(n)) = \{f(n) \,|\, \exists c, n_0 > 0 \,:\, 0 \leqslant f(n) \leqslant cg(n),\, \forall n \geqslant n_0\}$$

- assume non-negative functions (otherwise add $|\cdot|$) to the definitions
- $f(n) \in \mathcal{O}(g(n))$ represents an asymptotic upper bound on $f(n)$ up to a constant
- example: $f(n) = 3\sqrt{n} + 2\log n + 8n + 85n^2 \in \mathcal{O}(n^2)$

Notes

To the Board

Big-O

How to measure the impact of n on error or algorithmic cost?

$\mathcal{O}(\cdot)$

Let $g(n)$ be a function of n. Then define

$$\mathcal{O}(g(n)) = \{f(n) \,|\, \exists c, n_0 > 0 \,:\, 0 \leqslant f(n) \leqslant cg(n),\, \forall n \geqslant n_0\}$$

- assume non-negative functions (otherwise add $|\cdot|$) to the definitions
- $f(n) \in \mathcal{O}(g(n))$ represents an asymptotic upper bound on $f(n)$ up to a constant
- example: $f(n) = 3\sqrt{n} + 2\log n + 8n + 85n^2 \in \mathcal{O}(n^2)$

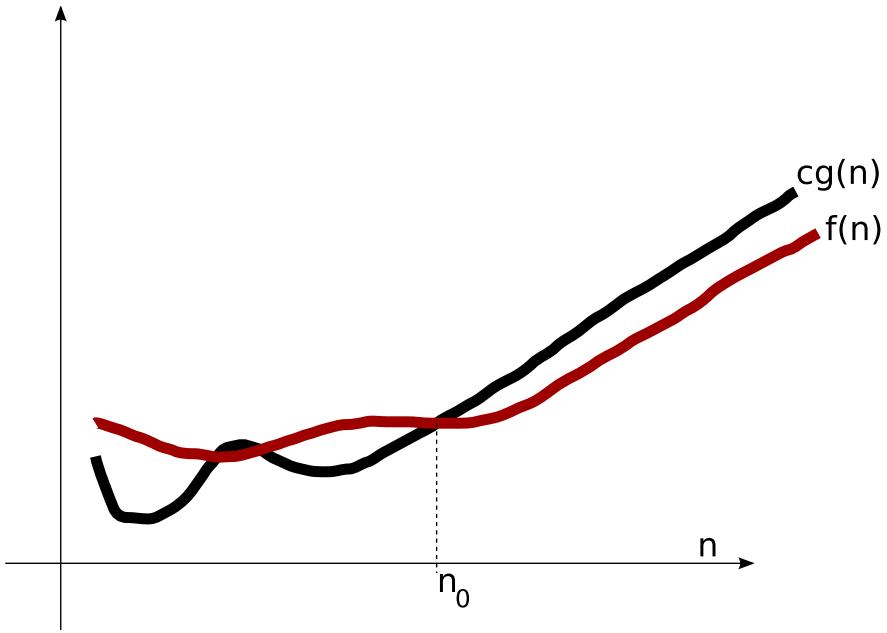
Big-O (Omicron)

asymptotic upper bound

$\mathcal{O}(\cdot)$

Let $g(n)$ be a function of n. Then define

$$\mathcal{O}(g(n)) = \{f(n) \mid \exists c, n_0 > 0 : 0 \leqslant f(n) \leqslant cg(n), \forall n \geqslant n_0\}$$



Big-O (Omicron)

asymptotic upper bound

$\mathcal{O}(\cdot)$

Let $g(n)$ be a function of n. Then define

$$\mathcal{O}(g(n)) = \{f(n) \,|\, \exists c, n_0 > 0 \,:\, 0 \leqslant f(n) \leqslant cg(n),\, \forall n \geqslant n_0\}$$

Note: If $f(n) \in \mathcal{O}(n^2)$, then technically $f(n) \in \mathcal{O}(n^3)$ and $f(n) \in \mathcal{O}(n^4)$. However, as a general rule we want the most restrictive case. This tells us "how fast" f is growing (shrinking).

We use $f(n) = \mathcal{O}(\cdot)$, $f(n) \sim \mathcal{O}(\cdot)$ and $f(n) \in \mathcal{O}(\cdot)$ interchangably.

Big-O is useful for quantifying how fast things grow (like flop counts), as well as how fast things shrink (like error).

Big-O

How to measure how fast something gets small?

$\mathcal{O}(\cdot)$

$$\mathcal{O}(g(n)) = \{f(n) \mid \exists c, n_0 > 0 : 0 \leqslant f(n) \leqslant cg(n), \forall n \geqslant n_0\}$$

What if $g(n)$ is shrinking, not growing?

Example: $\frac{\sin(n)+1}{n} = \mathcal{O}(1/n)$.

Notes

To the Board

Algebraic Convergence (J. P. Boyd)

Definition

The Algebraic Index of Convergence α is the largest number for which

$$\lim_{n\to\infty} |a_n| n^{\alpha} < \infty$$

where a_n are the coefficients in the sequence. Alternatively, α is the algebraic index if

$$a_n \sim \mathcal{O}(1/n^{\alpha})$$

Example

$$f(n) = 37n^{-3} + 5n^{-1/2}$$

Exponential Convergence (J. P. Boyd)

Definition

If the algebraic index α is unbounded (i.e. a_n decrease faster than $1/n^\alpha$ for any finite α), then the sequence converges exponentially (a.k.a. spectrally).

Alternatively, the sequence converges exponentially if for constants q and β

$$a_n \sim \mathcal{O}(e^{-qn^\beta})$$

where β is the exponential index of convergence and

$$\beta = \lim_{n\to\infty} \frac{\log|\log(|a_n|)|}{\log(qn)}$$

Rates of Exponential Convergence (J. P. Boyd)

Definition

A sequence a_n has supergeometric, geometric, or subgeometric if

$$\lim_{n\to\infty} \log(|a_n|)/n = \begin{cases} \infty, & \text{supergeometric} \\ \textit{constant}, & \text{geometric} \\ 0, & \text{subgeometric} \end{cases}$$

or, alternatively,

$$a_n \sim \mathcal{O}(e^{-(n/j)\log(n)}) : \text{ supergeometric}$$
$$a_n \sim \mathcal{O}(e^{-qn}) : \text{ geometric}$$
$$a_n \sim \mathcal{O}(e^{-qn^\beta}), \quad \beta < 1 : \text{ subgeometric}$$

Asymptotic Rate of Geometric Convergence (J. P. Boyd)

definition

If a sequence a_n has geometric convergence ($\beta = 1$) so that

$$a_n \sim \mathcal{O}(e^{-nq})$$

then the asymptotic rate of geometric convergence is q. Alternatively,

$$q = \lim_{n\to\infty} \{-\log |a_n|/n\}$$

Types/Rates of Convergence

So what?

More importantly, how do we even begin to determine which type of convergence we're looking at?

Graph It!

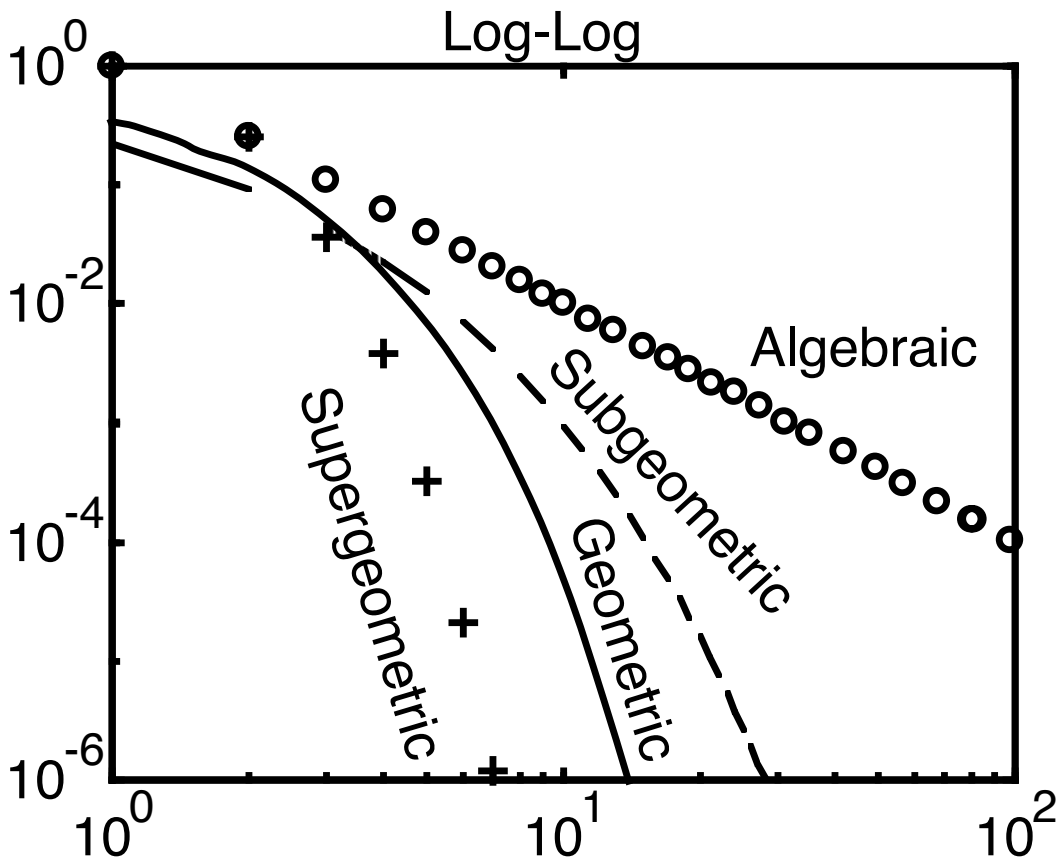


Figure 2.6: Same as previous figure except that the graph is log-log: the degree of the spectral coefficient n is now plotted on a logarithmic scale, too.

Graph It!

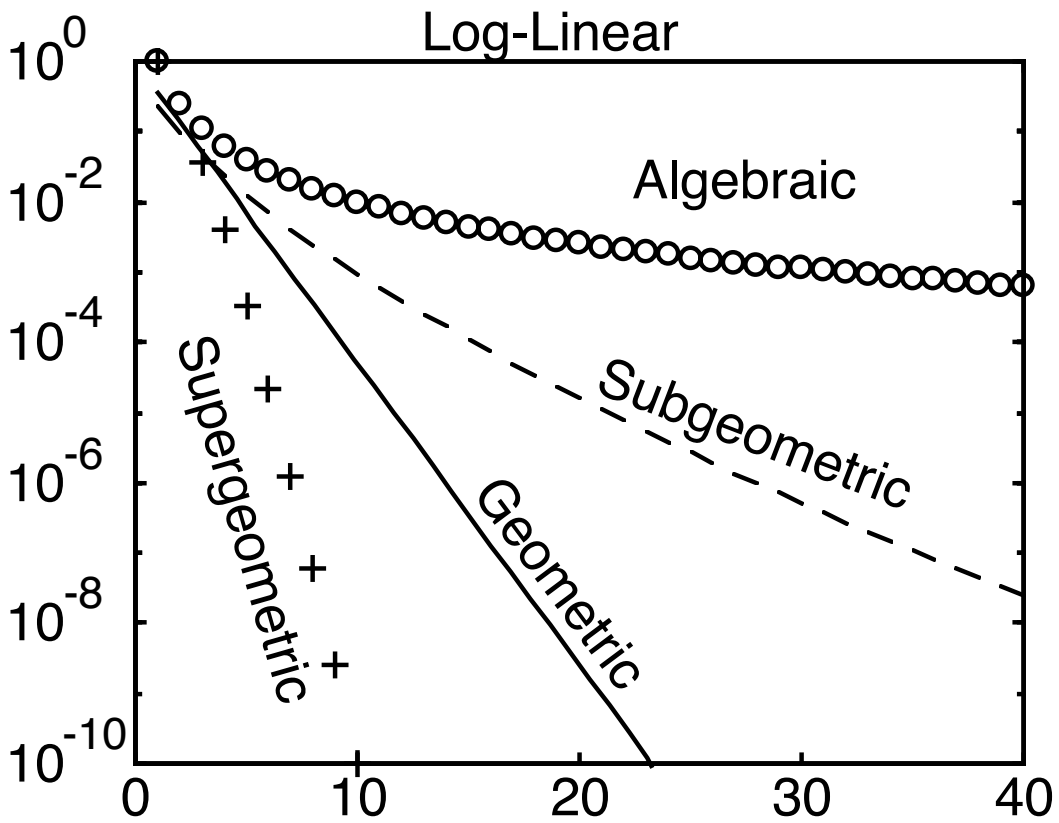


Figure 2.5: $\log |\ a_n\ |$ versus n for four rates of convergence. Circles: algebraic convergence, such as $a_n \sim 1/n^2$. Dashed: subgeometric convergence, such as $a_n \sim \exp(-1.5\ n^{2/3})$. Solid: geometric convergence, such as $\exp(-\mu\ n)$ for any positive μ. Pluses: supergeometric, such as $a_n \sim \exp(-n\ \log(n)\ )$ or faster decay.

Graph It!

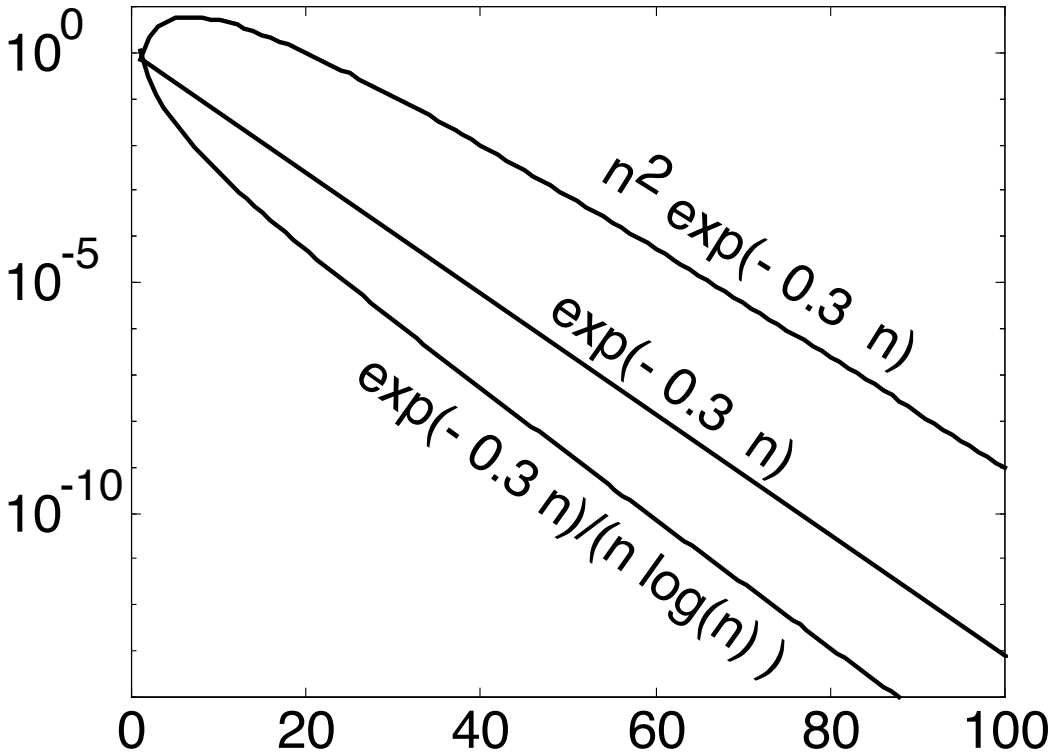


Figure 2.8: Spectral coefficients for three geometrically converging series. Although the three sets of coefficients differ through algebraic coefficients — the top curve is larger by n^2 than the middle curve, which in turn is larger by a factor of $n\log(n)$ than the bottom curve — the *exponential* dependence on n is the same for all. Consequently, all three sets of coefficients asymptote to parallel lines on this log-linear plot.

Convergence Types/Rates

This will be on homeworks and exams.
We'll learn some more techniques as the semester progresses.

Towards Floating Point Numbers

- We're familiar with base 10 representation of numbers:

$$1234 = 4 \times 10^0 + 3 \times 10^1 + 2 \times 10^2 + 1 \times 10^3$$

and

$$.1234 = 1 \times 10^{-1} + 2 \times 10^{-2} + 3 \times 10^{-3} + 4 \times 10^{-4}$$

- we write 1234.1234 as an integer part and a fractional part:

$$a_3 a_2 a_1 a_0 . b_1 b_2 b_3 b_4$$

- For some (even simple) numbers, there may be an *infinite* number of digits to the right:

$$\pi = 3.14159\ldots$$
$$1/9 = 0.11111\ldots$$
$$\sqrt{2} = 1.41421\ldots$$

Other bases

- So far, we have just base 10. What about base β?
- binary ($\beta = 2$), octal ($\beta = 8$), hexadecimal ($\beta = 16$), etc
- In the β-system we have

$$(a_n \dots a_2 a_1 a_0 . b_1 b_2 b_3 b_4 \dots)_\beta = \sum_{k=0}^{n} a_k \beta^k + \sum_{k=0}^{\infty} b_k \beta^{-k}$$

Notes

To the Board

Conversion from base-2 to base-10

Convert a base-2 number to base-10:

$$
\begin{aligned}
&(1001.101)_2 \\
&= 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} \\
&= 8 + 0 + 0 + 1 + 0.5 + 0 + 0.125 \\
&= 9.62
\end{aligned}
$$

For other numbers, such as $\frac{1}{5} = 0.2$, an infinite length is needed.

$$0.2 \quad \rightarrow \quad .0011\,0011\,0011\,\ldots$$

So 0.2 is stored just fine in base-10, but needs infinite number of digits in base-2

!!!

This is *roundoff* error in its basic form...

Intel

Business Day
The New York Times
D1
THURSDAY, NOVEMBER 24, 199

BUSINESS Digest

THURSDAY, NOVEMBER 24, 1994

DOW 30 Industrials	DOLLAR vs. Japanese Yen	OIL Nymex Spot	BONDS 30-Year Treasuries
3,674.63 −3.36	98.47 Yen +0.24 Yen	$18.15 +$0.33	7.94% −0.08

Companies

A flaw in the Pentium, the top chip made by Intel, can cause inaccurate calculations in certain rare cases. Intel said the chip would not have to be recalled. But many scientists and engineers who depend on precise calculations are concerned. [Page D1.]

Gibson Greetings and Bankers Trust settled a lawsuit over derivatives. Gibson agreed to pay Bankers Trust nearly $6.2 million, or about 30 percent of the $20.7 million that Bankers claimed it was owed in the trades between the two. [D1.]

Sorry, Charlie: H. J. Heinz and Leo Burnett are parting ways after 27 years. The Burnett agency created familiar advertising characters like Morris the finicky cat and Charlie the tuna. [D1.]

Pfizer will buy SmithKline Beecham's animal health business for $1.45 billion in cash. The deal would make Pfizer the world's largest maker of animal drugs, topping Merck. [D4.]

Kmart is seeking ways to save as much as $800 million in two years in addition to store closings and job cuts. [D4.]

A Federal judge froze a New York brokerage account of a British businessman accused of illegal insider trading. [D4.]

The F.T.C. is seeking data on alliances with Caremark International, in a broadening of an investigation of relations between drug makers and managed health-care drug distributors. [D4.]

Metallgesellschaft of Germany plans to write down the paper value of its shares by half in an effort to rescue itself. [D4.]

I.B.M., AT&T, Apple Computer and Siemens plan to create a universal communication language for computers. [D5.]

Markets

After a roller-coaster session, stocks ended slightly lower. The Standard & Poor's 500 index fell just 0.16 point, easing investors' concerns that stocks were in for a long, hard descent. [D1.]

A stock slide is not likely to change Federal Reserve policy. Indeed, stocks' strength until now surprised Fed officials. [D8.]

Stock markets fell sharply across Europe as investors took their cue from Wall Street and sought refuge in bonds. [D6.]

Flaw Undermines Accuracy of Pentium Chips

By JOHN MARKOFF

Special to The New York Times

SAN FRANCISCO, Nov. 23 — An elusive circuitry error is causing a chip used in millions of computers to generate inaccurate results in certain rare cases, heightening anxiety among many scientists and engineers who rely on their machines for precise calculations.

The flaw, an error in division, has been found in the Pentium, the current top microprocessor of the Intel Corporation, the world's largest chip maker. The chip, in several different configurations, is used in many computers sold for home and business use, including those made by I.B.M., Compaq, Dell, Gateway 2000 and others.

The flaw appears in all Pentium chips now on the market, in certain types of division problems involving more than five significant digits, a mathematical term that can include numbers before and after a decimal point.

Intel declined to say how many Pentium chips it made or sold, but Dataquest, a market research company in San Jose, Calif., estimated that in 1994 Intel would sell 5.5 million to 6 million Pentiums, roughly 10 percent of the number of personal computers sold worldwide.

Intel said yesterday that it did not believe the chip needed to be recalled, asserting that the typical user would have but one chance in more than nine billion of encountering an inaccurate result as a consequence of the error, and thus there was no noticeable consequence to users of business or home computers. Indeed, the company said it was continuing to send computer makers Pentium chips built before the problem was detected.

William Kahan of the University of California at Berkeley, one of the nation's experts on computer mathematics, expressed skepticism about Intel's contentions that the error would only occur in extremely rare instances.

"These kinds of statistics have to cause some wonderment," he said. "They are based on assertions about the probability of events whose probability we don't know."

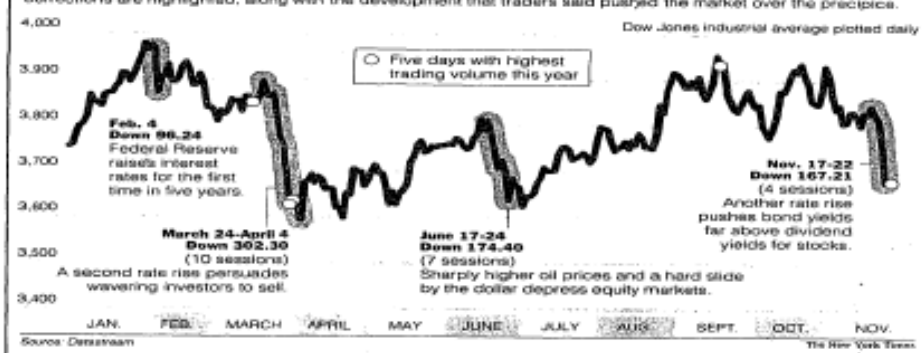
At the Jet Propulsion Laboratory in Pasadena, Calif., one satellite communications researcher who learned of the error this week, said six Pentium machines were used in his group and their use had been suspended for now.

"The Pentium appeared as a cou-

Continued on Page D5

Four Sharp Corrections in the Dow

This year the stock market has tended to gain gradually for weeks or bob nervously around a trading range with little direction, waiting for the moment when some bad news or unfavorable trend sets off a headlong flight. Then, once prices have been beaten down for a few days, the buyers reappear and the bleeding stops. Four of these corrections are highlighted, along with the development that traders said pushed the market over the precipice.



Gibson Suit On Trades Is Settled

Bankers Trust Gets 30% of Debt Claimed

By MICHAEL QUINT

Gibson Greetings Inc. and the Bankers Trust Company said yesterday that they had settled Gibson's lawsuit accusing the bank of improperly leading the company to engage in risky financial trades.

Under the out-of-court settlement, Gibson will pay Bankers Trust nearly $6.2 million, or about 30 percent of the $20.7 million that Bankers Trust contended it was owed under the trades.

While neither side boasted of victory, Douglas Kidd, a spokesman for Bankers Trust, acknowledged that

Intel

Flaw Undermines Accuracy of Pentium Chips
By JOHN MARKOFFSpecial to The New York Times
New York Times (1857-Current file); Nov 24, 1994; ProQuest Historical Newspapers The New York Times (1851 - 2002)
pg. D1

Flaw Undermines Accuracy of Pentium Chips

By JOHN MARKOFF

Special to The New York Times

SAN FRANCISCO, Nov. 23 — An elusive circuitry error is causing a chip used in millions of computers to generate inaccurate results in certain rare cases, heightening anxiety among many scientists and engineers who rely on their machines for precise calculations.

The flaw, an error in division, has been found in the Pentium, the current top microprocessor of the Intel Corporation, the world's largest chip maker. The chip, in several different configurations, is used in many computers sold for home and business use, including those made by I.B.M., Compaq, Dell, Gateway 2000 and others.

The flaw appears in all Pentium chips now on the market, in certain types of division problems involving more than five significant digits, a mathematical term that can include numbers before and after a decimal point.

Intel declined to say how many Pentium chips it made or sold, but Dataquest, a market research company in San Jose, Calif., estimated that in 1994 Intel would sell 5.5 million to 6 million Pentiums, roughly 10 percent of the number of personal computers sold worldwide.

Intel said yesterday that it did not believe the chip needed to be recalled, asserting that the typical user would have but one chance in more than nine billion of encountering an inaccurate result as a consequence of the error, and thus there was no noticeable consequence to users of business or home computers. Indeed, the company said it was continuing to send computer makers Pentium chips built before the problem was detected.

William Kahan of the University of California at Berkeley, one of the nation's experts on computer mathematics, expressed skepticism about Intel's contentions that the error would only occur in extremely rare instances.

"These kinds of statistics have to cause some wonderment," he said. "They are based on assertions about the probability of events whose probability we don't know."

At the Jet Propulsion Laboratory in Pasadena, Calif., one satellite communications researcher who learned of the error this week, said six Pentium machines were used in his group and their use had been suspended for now.

"The Pentium appeared as a cost-

Continued on Page D5

Intel

tium Chips

In some complex division problems, annoying errors.

corrected.

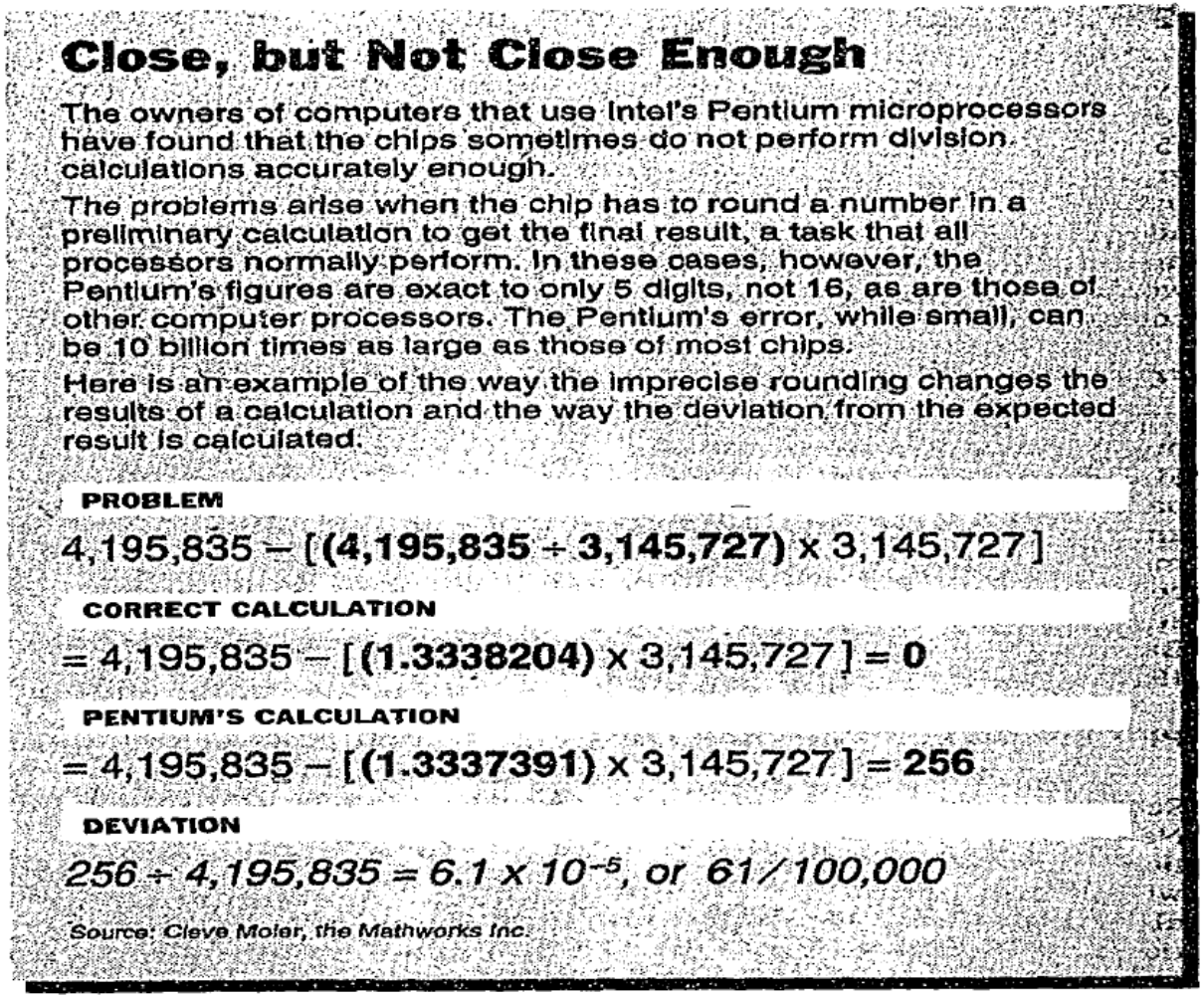
Some computer users said they believed that Intel had not acted quickly enough after discovering the error.

"Intel has known about this since the summer; why didn't they tell anyone?" said Andrew Schulman, the author of a series of technical books on PC's. "It's a hot issue, and I don't think they've handled this well.

The company said that after it discovered the problem this summer, it ran months of simulations of different applications, with the help of outside experts, to determine whether the problem was serious.

The Pentium error occurs in a portion of the chip known as the floating point unit, which is used for extremely precise computations. In rare cases, the error shows up in the result of a division operation.

Intel said the error occurred because of an omission in the translation of a formula into computer



Close, but Not Close Enough

The owners of computers that use Intel's Pentium microprocessors have found that the chips sometimes do not perform division calculations accurately enough.

The problems arise when the chip has to round a number in a preliminary calculation to get the final result, a task that all processors normally perform. In these cases, however, the Pentium's figures are exact to only 5 digits, not 16, as are those of other computer processors. The Pentium's error, while small, can be 10 billion times as large as those of most chips.

Here is an example of the way the imprecise rounding changes the results of a calculation and the way the deviation from the expected result is calculated.

PROBLEM

$4{,}195{,}835 - [(4{,}195{,}835 \div 3{,}145{,}727) \times 3{,}145{,}727]$

CORRECT CALCULATION

$= 4{,}195{,}835 - [(1.3338204) \times 3{,}145{,}727] = 0$

PENTIUM'S CALCULATION

$= 4{,}195{,}835 - [(1.3337391) \times 3{,}145{,}727] = 256$

DEVIATION

$256 \div 4{,}195{,}835 = 6.1 \times 10^{-5}$, or $61/100{,}000$

Source: Cleve Moler, the Mathworks Inc.

Intel Timeline

from emery.com

June 1994	Intel engineers discover the division error. Managers decide the error will not impact many people. Keep the issue internal.
June 1994	Dr Nicely at Lynchburg College notices computation problems
Oct 19, 1994	After months of testing, Nicely confirms that other errors are not the cause. The problem is in the Intel Processor.
Oct 24, 1994	Nicely contacts Intel. Intel duplicates error.
Oct 30, 1994	After no action from Intel, Nicely sends an email

Intel Timeline

from emery.com

```
FROM:  Dr. Thomas R. Nicely
       Professor of Mathematics
       Lynchburg College
       1501 Lakeside Drive
       Lynchburg, Virginia 24501-3199

       Phone:     804-522-8374
       Fax:       804-522-8499
       Internet:  nicely@acavax.lynchburg.edu

TO:    Whom it may concern

RE:    Bug in the Pentium FPU

DATE:  30 October 1994

It appears that there is a bug in the floating point unit
(numeric coprocessor) of many, and perhaps all, Pentium
processors.

In short, the Pentium FPU is returning erroneous values
for certain division operations. For example,
     0001/824633702441.0
is calculated incorrectly (all digits beyond the eighth
significant digit are in error). This can be verified in
compiled code, an ordinary spreadsheet such as Quattro Pro
or Excel, or even the Windows calculator (use the
scientific mode), by computing
     00(824633702441.0)*(1/824633702441.0),
which should equal 1 exactly (within some extremely small
rounding error; in general, coprocessor results should
contain 19 significant decimal digits).  However, the
Pentiums tested return
     0000.999999996274709702
.
.
.
```

Intel Timeline

from emery.com

Nov 1, 1994	Software company Phar Lap Software receives Nicely's email. Sends to colleagues at Microsoft, Borland, Watcom, etc. decide the error will not impact many people. Keep the issue internal.
Nov 2, 1994	Email with description goes global.
Nov 15, 1994	USC reverse-engineers the chip to expose the problem. Intel still denies a problem. Stock falls.
Nov 22, 1994	CNN *Moneyline* interviews Intel. Says the problem is minor.
Nov 23, 1994	The MathWorks develops a fix.
Nov 24, 1994	New York Times story. Intel still sending out flawed chips. Will replace chips only if it caused a problem in an important application.
Dec 12, 1994	IBM halts shipment of Pentium based PCs
Dec 16, 1994	Intel stock falls again.
Dec 19, 1994	More reports in the NYT: lawsuits, etc.
Dec 20, 1994	Intel admits. Sets aside $420 million to fix.

Numerical ”bugs”

Obvious

Software has bugs

Not-SO-Obvious

Numerical software has two unique bugs:

1. roundoff error
2. truncation error

Numerical Errors

Roundoff

Roundoff occurs when digits in a decimal point (0.3333...) are lost (0.3333) due to a limit on the memory available for storing one numerical value.

Truncation

Truncation error occurs when discrete values are used to approximate a mathematical expression.

Uncertainty: well- or ill-conditioned?

Errors in input data can cause *uncertain* results

- input data can be experimental or rounded. leads to a certain variation in the results
- *well-conditioned*: numerical results are insensitive to small variations in the input
- *ill-conditioned*: small variations lead to drastically different numerical calculations (a.k.a. poorly conditioned)

Our Job

As numerical analysts, we need to

1. solve a problem so that the calculation is not susceptible to large roundoff error
2. solve a problem so that the approximation has a *tolerable* truncation error

How?

- incorporate roundoff-truncation knowledge into
  - the mathematical model
  - the method
  - the algorithm
  - the software design
- awareness $\rightarrow$ correct interpretation of results

Floating Points

Normalized Floating-Point Representation

Real numbers are stored as

$$x = \pm(0.d_1 d_2 d_3 \ldots d_m)_\beta \times \beta^e$$

- $d_1 d_2 d_3 \ldots d_m$ is the mantissa, e is the exponent
- e is negative, positive or zero
- the general normalized form requires $d_1 \neq 0$

Floating Point

Example

In base 10

- 1000.12345 can be written as

$$(0.100012345)_{10} \times 10^4$$

- 0.000812345 can be written as

$$(0.812345)_{10} \times 10^{-3}$$

Floating Point

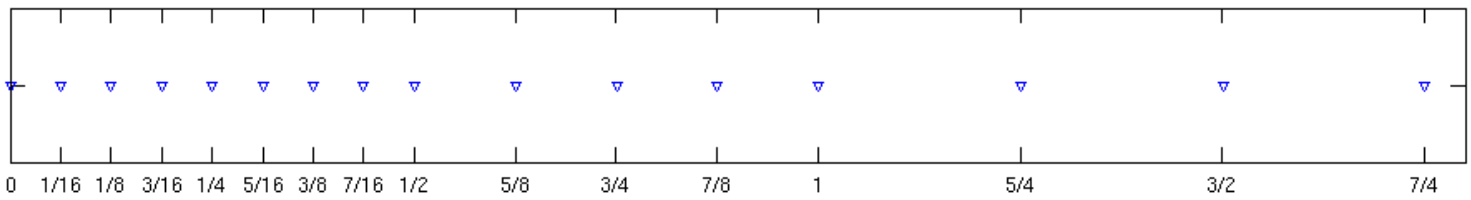
Suppose we have only 3 bits for a mantissa and a 2 bit exponent stored like

$.d_1$	d_2	d_3	e_1	e_2

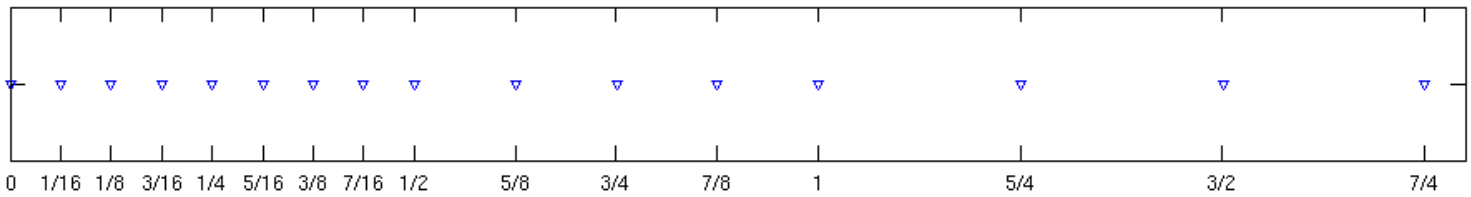
All possible combinations give:

$$\begin{aligned} 000_2 &= 0 \\ &\dots \\ 111_2 &= 7 \end{aligned} \qquad \times 2^{-1,0,1}$$

So we get $0, \frac{1}{16}, \frac{2}{16}, \dots, \frac{7}{16}, 0, \frac{1}{4}, \frac{2}{4}, \dots, \frac{7}{4}$, and $0, \frac{1}{8}, \frac{2}{8}, \dots, \frac{7}{8}$. On the real line:



Overflow, Underflow



- computations too close to zero may result in *underflow*
- computations too large may result in *overflow*
- overflow error is considered more severe
- underflow can just fall back to 0

Normalizing

If we use the normalized form in our 5-bit case, we lose $0.001_2 \times 2^{-1,0,1}$. So we cannot represent $\frac{1}{16}$, $\frac{1}{8}$, and $\frac{3}{16}$.