

We care about your data, and we'd like to use cookies to give you a smooth browsing experience. Please agree and read more about our

[privacy\\_policy](#).

AGREE

## NUMBER THEORY

### Mathematicians Discover the Perfect Way to Multiply

By KEVIN HARTNETT

April 11, 2019

*By chopping up large numbers into smaller ones, researchers have rewritten a fundamental mathematical speed limit.*

65 |



Mengxin Li for Quanta Magazine

We care about your data, and we'd like to use cookies to give you a smooth browsing experience. Please agree and read more about our [privacy policy](#).

On March 18, two researchers described the fastest method ever discovered for multiplying two very large numbers. The paper marks the culmination of a long-running search to find the most efficient procedure for performing one of the most basic operations in math.

“Everybody thinks basically that the method you learn in school is the best one, but in fact it’s an active area of research,” said Joris van der Hoeven, a mathematician at the French National Center for Scientific Research and one of the co-authors.

The complexity of many computational problems, from calculating new digits of pi to finding large prime numbers, boils down to the speed of multiplication. Van der Hoeven describes their result as setting a kind of mathematical speed limit for how fast many other kinds of problems can be solved.

“In physics you have important constants like the speed of light which allow you to describe all kinds of phenomena,” van der Hoeven said. “If you want to know how fast computers can solve certain mathematical problems, then integer multiplication pops up as some kind of basic building brick with respect to which you can express those kinds of speeds.”

Most everyone learns to multiply the same way. We stack two numbers, multiply every digit in the bottom number by every digit in the top number, and do addition at the end. If you’re multiplying two two-digit numbers, you end up performing four smaller multiplications to produce a final product.

The grade school or “carrying” method requires about  $n^2$  steps, where  $n$  is the number of digits of each of the numbers you’re multiplying. So three-digit numbers require nine multiplications, while 100-digit numbers require 10,000 multiplications.

The carrying method works well for numbers with just a few digits, but it bogs down when we’re multiplying numbers with millions or billions of digits (which is what computers do to accurately calculate pi or as part of the worldwide search for large primes). To multiply two numbers with 1 billion digits requires 1 billion squared, or  $10^{18}$ , multiplications, which would take a modern computer roughly 30 years.

For millennia it was widely assumed that there was no faster way to multiply. Then in 1960, the 23-year-old Russian mathematician Anatoly Karatsuba took a seminar led by Andrey Kolmogorov, one of the great mathematicians of the 20th century. Kolmogorov asserted that there was no general procedure for doing multiplication that required fewer than  $n^2$  steps. Karatsuba thought there was — and after a week of searching, he found it.

Karatsuba’s method involves breaking up the digits of a number and recombining them in a novel way that allows you to substitute a small number of additions and subtractions for a large number of multiplications. The method saves time because addition takes only  $2n$  steps, as opposed to  $n^2$  steps.

We care about your data, and we'd like to use cookies to give you a smooth browsing experience. Please agree and read more about our [privacy\\_policy](#).

multiply two  $n$ -digit numbers. Then in 1960, the Russian mathematician Anatoly Karatsuba proposed a better way.

### Traditional Way to Multiply $25 \times 63$

Requires **four** single-digit multiplications and some additions.

<b>STEP A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>
$\begin{array}{r} 25 \\ \times 63 \\ \hline 1200 \end{array}$	$+$	$\begin{array}{r} 25 \\ \times 63 \\ \hline 15 \end{array}$	$+$	$\begin{array}{r} 25 \\ \times 63 \\ \hline 60 \end{array}$
$+$		$+$		$\begin{array}{r} 25 \\ \times 63 \\ \hline 300 \end{array}$
$= 1575$				

### Karatsuba Method for $25 \times 63$

Requires **three** single-digit multiplications plus some additions and subtractions.

<b>STEP A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>	<b>F</b>	<b>G</b>
Break numbers up.	Multiply the tens.	Multiply the ones.	Add the digits.	Multiply the sums.	Subtract B and C from E.	Assemble the numbers.
$25 \rightarrow \begin{array}{ c c } \hline 2 & 5 \\ \hline \end{array}$ $63 \rightarrow \begin{array}{ c c } \hline 6 & 3 \\ \hline \end{array}$	$\begin{array}{r} 2 \\ \times 6 \\ \hline 12 \end{array}$	$\begin{array}{r} 5 \\ \times 3 \\ \hline 15 \end{array}$	$2 + 5 = 7$ $6 + 3 = 9$	$\begin{array}{r} 7 \\ \times 9 \\ \hline 63 \end{array}$	$\begin{array}{r} 63 \\ - 15 \\ \hline - 12 \\ \hline 36 \end{array}$	$\begin{array}{r} 12 \\ 36 \\ + 15 \\ \hline 1575 \end{array}$

**MULTIPLIED SAVINGS:** As numbers increase in size, the Karatsuba method can be used repeatedly, breaking large numbers into small pieces to save an increasing number of single-digit multiplications.

Traditional way to multiply  $2,531 \times 1,467$  requires **16** single-digit multiplications.

$\begin{array}{r} 2531 \\ \times 1467 \\ \hline \end{array}$	$+$	$\begin{array}{r} 2531 \\ \times 1467 \\ \hline \end{array}$	$+$	$\begin{array}{r} 2531 \\ \times 1467 \\ \hline \end{array}$	$+$	$\begin{array}{r} 2531 \\ \times 1467 \\ \hline \end{array}$	$= 3712977$
--	-----	--	-----	--	-----	--	-------------

Karatsuba method to multiply  $2,531 \times 1,467$  requires **9** single-digit multiplications.

<b>STEP A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>	<b>F</b>	<b>G</b>																					
$\begin{array}{r} 25 \\ \times 14 \\ \hline 14 \end{array}$	$\begin{array}{r} 25 \\ \times 14 \\ \hline 350 \end{array}$	$\begin{array}{r} 31 \\ \times 67 \\ \hline 2077 \end{array}$	$25 + 31 = 56$ $14 + 67 = 81$	$\begin{array}{r} 56 \\ \times 81 \\ \hline 4536 \end{array}$	$\begin{array}{r} 4536 \\ - 2077 \\ \hline - 350 \\ \hline 2109 \end{array}$	$\begin{array}{r} 350 \\ 2109 \\ + 2077 \\ \hline 3712977 \end{array}$																					
<table border="0"> <tr> <td><b>A</b></td><td><b>B</b></td><td><b>C</b></td><td><b>D</b></td><td><b>E</b></td><td><b>F</b></td><td><b>G</b></td> </tr> <tr> <td>Run Karatsuba method on:</td> <td><math>\begin{array}{ c c } \hline 2 &amp; 5 \\ \hline \end{array}</math></td> <td><math>\begin{array}{ c c } \hline 3 &amp; 1 \\ \hline \end{array}</math></td> <td><math>\begin{array}{ c c } \hline 3 &amp; 1 \\ \hline \end{array}</math></td> <td><math>\begin{array}{ c c } \hline 5 &amp; 6 \\ \hline \end{array}</math></td> <td><math>\begin{array}{ c c } \hline 5 &amp; 6 \\ \hline \end{array}</math></td> <td><math>\begin{array}{ c c } \hline 5 &amp; 6 \\ \hline \end{array}</math></td> </tr> <tr> <td></td> <td><math>\begin{array}{ c c } \hline 1 &amp; 4 \\ \hline \end{array}</math></td> <td><math>\begin{array}{ c c } \hline 6 &amp; 7 \\ \hline \end{array}</math></td> <td></td> <td><math>\begin{array}{ c c } \hline 8 &amp; 1 \\ \hline \end{array}</math></td> <td></td> <td><math>\begin{array}{ c c } \hline 8 &amp; 1 \\ \hline \end{array}</math></td> </tr> </table>							<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>	<b>F</b>	<b>G</b>	Run Karatsuba method on:	$\begin{array}{ c c } \hline 2 & 5 \\ \hline \end{array}$	$\begin{array}{ c c } \hline 3 & 1 \\ \hline \end{array}$	$\begin{array}{ c c } \hline 3 & 1 \\ \hline \end{array}$	$\begin{array}{ c c } \hline 5 & 6 \\ \hline \end{array}$	$\begin{array}{ c c } \hline 5 & 6 \\ \hline \end{array}$	$\begin{array}{ c c } \hline 5 & 6 \\ \hline \end{array}$		$\begin{array}{ c c } \hline 1 & 4 \\ \hline \end{array}$	$\begin{array}{ c c } \hline 6 & 7 \\ \hline \end{array}$		$\begin{array}{ c c } \hline 8 & 1 \\ \hline \end{array}$		$\begin{array}{ c c } \hline 8 & 1 \\ \hline \end{array}$
<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>	<b>F</b>	<b>G</b>																					
Run Karatsuba method on:	$\begin{array}{ c c } \hline 2 & 5 \\ \hline \end{array}$	$\begin{array}{ c c } \hline 3 & 1 \\ \hline \end{array}$	$\begin{array}{ c c } \hline 3 & 1 \\ \hline \end{array}$	$\begin{array}{ c c } \hline 5 & 6 \\ \hline \end{array}$	$\begin{array}{ c c } \hline 5 & 6 \\ \hline \end{array}$	$\begin{array}{ c c } \hline 5 & 6 \\ \hline \end{array}$																					
	$\begin{array}{ c c } \hline 1 & 4 \\ \hline \end{array}$	$\begin{array}{ c c } \hline 6 & 7 \\ \hline \end{array}$		$\begin{array}{ c c } \hline 8 & 1 \\ \hline \end{array}$		$\begin{array}{ c c } \hline 8 & 1 \\ \hline \end{array}$																					

We care about your data, and we'd like to use cookies to give you a smooth browsing experience. Please agree and read more about our [privacy\\_policy](#).

algorithm.

When dealing with large numbers, you can repeat the Karatsuba procedure, splitting the original number into almost as many parts as it has digits. And with each splitting, you replace multiplications that require many steps to compute with additions and subtractions that require far fewer.

“You can turn some of the multiplications into additions, and the idea is additions will be faster for computers,” said [David Harvey](#), a mathematician at the University of New South Wales and a co-author on the new paper.

Karatsuba’s method made it possible to multiply numbers using only  $n^{1.58}$  single-digit multiplications. Then in 1971 [Arnold Schönhage](#) and Volker Strassen published a method capable of multiplying large numbers in  $n \times \log n \times \log(\log n)$  multiplicative steps, where  $\log n$  is the logarithm of  $n$ . For two 1-billion-digit numbers, Karatsuba’s method would require about 165 trillion additional steps.



Joris van der Hoeven, a mathematician at the French National Center for Scientific Research, helped create the fastest-ever method for multiplying very large integers.

---

École Polytechnique

Schönhage and Strassen’s method, which is how computers multiply huge numbers, had two other important long-term consequences. First, it introduced the use of a technique from the field of signal processing called a fast Fourier transform. The technique has been the basis for every fast multiplication algorithm since.

We care about your data, and we'd like to use cookies to give you a smooth browsing experience. Please agree and read more about our [privacy policy](#).

operation as fundamental as multiplication must have a limit more elegant than  $n \times \log n \times \log(\log n)$ .

“It was kind of a general consensus that multiplication is such an important basic operation that, just from an aesthetic point of view, such an important operation requires a nice complexity bound,” Fürer said. “From general experience the mathematics of basic things at the end always turns out to be elegant.”

Schönhage and Strassen’s ungainly  $n \times \log n \times \log(\log n)$  method held on for 36 years. In 2007 Fürer beat it and the floodgates opened. Over the past decade, mathematicians have found successively faster multiplication algorithms, each of which has inched closer to  $n \times \log n$ , without quite reaching it. Then last month, Harvey and van der Hoeven got there.

Their method is a refinement of the major work that came before them. It splits up digits, uses an improved version of the fast Fourier transform, and takes advantage of other advances made over the past forty years. “We use [the fast Fourier transform] in a much more violent way, use it several times instead of a single time, and replace even more multiplications with additions and subtractions,” van der Hoeven said.

Harvey and van der Hoeven’s algorithm proves that multiplication can be done in  $n \times \log n$  steps. However, it doesn’t prove that there’s no faster way to do it. Establishing that this is the best possible approach is much more difficult. At the end of February, a team of computer scientists at Aarhus University [posted a paper arguing](#) that if another unproven conjecture is also true, this is indeed the fastest way multiplication can be done.

And while the new algorithm is important theoretically, in practice it won’t change much, since it’s only marginally better than the algorithms already being used. “The best we can hope for is we’re three times faster,” van der Hoeven said. “It won’t be spectacular.”

In addition, the design of computer hardware has changed. Two decades ago, computers performed addition much faster than multiplication. The speed gap between multiplication and addition has narrowed considerably over the past 20 years to the point where multiplication can be even faster than addition in some chip architectures. With some hardware, “you could actually do addition faster by telling the computer to do a multiplication problem, which is just insane,” Harvey said.

Hardware changes with the times, but best-in-class algorithms are eternal. Regardless of what computers look like in the future, Harvey and van der Hoeven’s algorithm will still be the most efficient way to multiply.

*This article was reprinted on [Wired.com](#) and in Spanish at [Investigacionyciencia.es](#).*