Brun 常数和英特尔的 bug

郭　学　军
南京大学数学系

1994 年 6 月 13 日, Lynchburg College[1] 的 Thomas Nicely[2] 教授发现他的电脑计算结果似乎出了一些问题. 当时他正在编程计算下面这个级数

$$B_2 = \left(\frac{1}{3}+\frac{1}{5}\right)+\left(\frac{1}{5}+\frac{1}{7}\right)+\left(\frac{1}{11}+\frac{1}{13}\right)+\cdots \tag{1}$$

的近似值. 这个级数是对所有的孪生素数的倒数进行求和. 孪生素数是指相隔为 2 的两个素数, 例如一百以内共有 8 对孪生素数：

$$(3,5),\ (5,7),\ (11,13),\ (17,19),\ (29,31),\ (41,43),\ (59,61),\ (71,73).$$

如果我们用 $B_2(x)$ 代表所有小于 x 的孪生素数的倒数和, 那么

$$\begin{aligned} B_2(100) = &\left(\frac{1}{3}+\frac{1}{5}\right)+\left(\frac{1}{5}+\frac{1}{7}\right)+\left(\frac{1}{11}+\frac{1}{13}\right)+\left(\frac{1}{17}+\frac{1}{19}\right)\\ &+\left(\frac{1}{29}+\frac{1}{31}\right)+\left(\frac{1}{41}+\frac{1}{43}\right)+\left(\frac{1}{59}+\frac{1}{61}\right)+\left(\frac{1}{71}+\frac{1}{73}\right)\\ &= 1.330990365719\cdots \end{aligned}$$

Thomas Nicely 知道挪威数学家 Viggo Brun[3] 在 1919 年就证明了 (1) 式中的级数是收敛的. 公式 (1) 中的“B”就是 Brun 的首字母, 这个数被现在被数学界称为“Brun 常数”. 这个 Brun 常数有个特别之处, 就是定义它的级数

[1] 2018 年更名为University of Lynchburg
[2] Thomas Nicely, 1943-2019, 美国数学家
[3] Viggo Brun, 1885-1978, 挪威数学家

(1) 收敛地特别慢.

x	$B_2(x)$
10^2	$1.330990365719\cdots$
10^4	$1.616893557432\cdots$
10^6	$1.710776930804\cdots$
10^8	$1.758815621067\cdots$
10^{10}	$1.787478502719\cdots$
10^{12}	$1.806592419175\cdots$
10^{14}	$1.820244968130\cdots$
10^{15}	$1.825706013240\cdots$
10^{16}	$1.830484424658\cdots$

从上面的表里可以看出来, 随着 x 的增大, Brun 常数在缓慢地增长. 即使你计算了小于一百万亿的所有孪生素数, 你甚至都不敢说 Brun 常数是不是小于 2.

Thomas Nicely 在弗吉尼亚大学取得博士学位. 他本科时期学的是物理, 后来转向应用数学, 然后开始研究数论–一种看起来跟“应用”最不沾边的学问. 他对 Brun 常数的研究开始于 1993 年的三月. 到了 1994 年 4 月, Thomas Nicely 在他的计算集群中加入了英特尔公司刚推出的奔腾系统, 运算速度得到了极大的提升. 到 1994 年 6 月 13 日, 他的计算机在计算 Brun 常数时, 孪生素数已经到了 10^{13} 这个数量级. 在他之前, 1975 年, Richard P. Brent 计算 Brun 常数时, 使用的孪生素数是 10^{11} 这个数量级.

这时候, Thomas Nicely 发现了错误. 这个级数收敛极慢, 在他之前, 从没有人计算到他的精度, 那他是怎么发现错误的呢? 这里有一件很奇妙的事情. 数学家们借助一些尚未被证明的猜测和经验公式, 给出了 Brun 常数的估计:

$$B_2 = B_2(x) + \frac{4c_2}{\ln(x)} + O\left(\frac{1}{\sqrt{x}\ln(x)}\right), \tag{2}$$

这里的常数

$$\begin{aligned} c_2 &= \prod_{\substack{p \text{ 素数} \\ p\geq 3}} \left(1 - \frac{1}{(p-1)^2}\right) \\ &= 0.660161815846869573927812110014\cdots . \end{aligned}$$

换句话说, 我们可以不用 $B_2(x)$ 来逼近 B_2, 而用 $B_2(x) + \frac{4c_2}{\ln(x)}$ 来逼近 B_2. 这

次的逼近, 效果就好极了.

x	$B_2(x)+\frac{4c_2}{\ln(x)}$
10^2	$1.9043996332901\cdots$
10^4	$1.9035981912177\cdots$
10^6	$1.9019133533279\cdots$
10^8	$1.9021679379607\cdots$
10^{10}	$1.9021603562335\cdots$
10^{12}	$1.9021606304377\cdots$
10^{14}	$1.9021605777833\cdots$

作为对比, 如果你用 $B_2(x)$ 来逼近 B_2, 那么当你计算到了 $x=10^{530}$ 时, $B_2(x)$ 甚至还没有达到 1.9, 换句话说, 连 B_2 的小数点之后的第一位数字就没有算对.

所以在经验公式的加持下, 前人的计算结果其实已经达到了很不错的精度, Thomas Nicely 把自己的计算结果与前人的计算结果进行对比, 发现了自己的计算数据的错误. 他首先怀疑是编程的错误, 开始了漫长的找 bug 的过程, 他确实找到了一些. 到 9 月 10 日的时候, 他确信他的程序中没有 bug 了. 于是他重新开始计算, 作为对比, 他在 486 电脑和奔腾电脑上分别用不同的算法进行计算, 以便进行比对. 10 月 4 日, 他发现了第一个错误, 奔腾电脑在计算孪生素数对 $(824633702441, 824633702443)$ 时出现了差错, 把 $\frac{1}{824633702441}$ 计算错了.

10 月 17 日, 他在一位同事的崭新的奔腾电脑上做这个计算, 发现了同样的错误, 而老式的 486 电脑计算结果却是正确的. 慢慢地, 他发现这个错误应该是奔腾芯片造成的. 于是他通知了英特尔公司, 也通知了镁光科技, 但是并没有得到满意的答复. 10 月 30 日, Thomas Nicely 给多个组织发送了一封email,

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

Thomas Nicely 的 email 立刻引起了轩然大波, 大家纷纷报告自己的电脑在运行了 Thomas Nicely 提供的代码之后, 出现了同样的计算错误. 英特尔公司最终不得不宣布召回有缺陷的芯片, 并花费 4.75 亿美元为所有用户更换了新的芯片. 英特尔公司后来认为, Thomas Nicely 所发现的芯片错误会导致在大约 8.77×10^9 次随机除法运算中, 出现一次错误. 普通的用户, 可能永远也遇不到这样的错误. 然而对Thomas Nicely 这样的计算数论科研人员来说, 遇到这样的错误几乎是必然的.

Thomas Nicely 的故事被CNN 等新闻媒体,《科学》等学术杂志报道之后成了一位明星一般的人物. 然而Thomas Nicely 为人低调, 从未把自己发现奔

腾芯片的缺陷当成什么伟大的成就. 他继续教书, 做研究, 并于 2000 年退休. 退休以后, 他还经常返回校园辅导学生. 他的同事回忆说 Thomas Nicely 是一位了不起的老师, 他最喜欢的地方就是教室, 他希望自己作为一位好老师被人们记住. Thomas Nicely 是一位体育迷, 喜欢运动, 还与他人合作开发了一款很受欢迎的橄榄球桌面游戏“Paydirt”.

Thomas Nicely 于 2019 年 9 月 11 日去世.