Abstract.

Numerical results on Waring's problem for cubes are given. In particular strong evidence is presented indicating the truth of the conjecture $G(3)=4$, i.e. that every sufficiently large number can be written as the sum of at most 4 positive cubes.

Introduction.

Waring's problem is concerned with the representation of integers as sums of powers of integers. More specifically, for $k=2, 3, 4, \ldots$ there exists an integer $g(k)$ such that every positive number $n$ can be written as the sum of at most $g(k)$ $k$th powers. Similarly there exists a number $G(k)$ such that all sufficiently large numbers $n$, e.g. $n \geq N(k)$, can be written as the sum of not more than $G(k)$ $k$th powers. Good expository treatments of this subject are provided by Ellison [4] (with a comprehensive reference list), Graham [5], and Hardy-Wright [6], Ch. XXI.

It has long been known that $g(2)=G(2)=4$, and further it follows from a theorem by Davenport [2] that $G(4)=16$. There is also an algorithm for computation of $g(k)$, $k \geq 6$ [6]. Further, Dickson [3] proved that $g(3)=9$, Chen [1] that $g(5)=37$; the value of $g(4)$ is not known, but there are overwhelming arguments in favor of $g(4)=19$. It is believed that the general formula is $g(k) = [(3/2)k]^2 + 2k-2$. Hence the solution with respect to $g(k)$ is practically complete.

In contrast, no values of $G(k)$ are known, except when $k=2$ and $k=4$ as stated above. In this paper we will report on extensive numerical calculations on Waring's problem for cubes. The best theoretical result so far seems to be a theorem by Davenport [2] that the number of integers $\leq N$ requiring 5 cubes is $O(N^{1-1/30+\varepsilon})$ with $\varepsilon>0$. Hence almost all integers can be represented as sums of 4 cubes. Amazingly accurate and extensive numerical computations were performed by Alfred E. Western [7] already in 1926, long before the advent of automatic computers. At least two of his conjectures are supported by our investigation.

In this paper we shall present convincing numerical evidence that $G(3)=4$, and we also suggest definite results (some of which are previously known) for the number of members in the classes $C_6$, $C_7$, $C_8$ and $C_9$. Here $C_k$ is the set of numbers

Received January 23, 1981.
which can be represented as sums of \( k \) but not less than \( k \) cubes. Hence, \( 216 = 6^3 \in C_1 \) although \( 3^3 + 4^3 + 5^3 = 216 \). Our conjecture \( G(3) = 4 \) rests on determination of the density of numbers requiring 5 cubes up to \( 4 \cdot 10^{11} \) and use of an extrapolation technique suggested by Western \([7]\). For obvious reasons it has been impossible to investigate all numbers \( < 4 \cdot 10^{11} \) which would indeed demand computers at least 1000 times as powerful as the present ones. Instead we have taken samples, first of length 20,000, then (around \( 10^{10} - 2 \cdot 10^{11} \)) of length 100,000, and finally for \( 4 \cdot 10^{11} \) of length \( 10^6 \).

**Numerical algorithm for investigation of** \( G(3) \).

A number \( N \) which can be represented as the sum of not more than 4 cubes can be split into two parts \( a \) and \( b \), both belonging to a set of numbers consisting of cubes and sums of two cubes. With this observation we prepared a vector by adding all pairs of cubes (including zero). This was done by selecting all possible pairs with sum in a certain interval, and when a number appeared, a corresponding digit in a bit pattern was put equal to 1. In this way we could cope with the problems that the sequence formed did not necessarily come in ascending order, and further that some numbers could appear several times (note e.g. the well-known example \( 10^3 + 9^3 = 12^3 + 1^3 \)). Once the binary representation was clear it was an easy matter to convert it to the vector mentioned above, and we could avoid a difficult and time-consuming sorting. Successive parts of the vector were moved to disk storage; the final vector contained more than 26 million elements.

Using this vector we tried to form all numbers in a certain interval by adding two elements (\( a \) and \( b \)) starting with one small and one large element. The number of failures as well as the corresponding numbers, i.e. those which required 5 cubes, were recorded. The computations were performed on a UNIVAC 1100/80. The program, written in FORTRAN and assembly code, took about 8 hours for forming the vector while a run of a sample of \( 10^6 \) numbers (actually from \( 4 \cdot 10^{11} \) to \( 4.00001 \cdot 10^{11} \)) took about 7 hours. The total computing time was about 25 hours. Unfortunately, the capacity of the computer puts a limit at about \( 4 \cdot 10^{11} \) and we hope that better opportunities will be offered by future more powerful computers. Our impression is that a sample of 10-100 million numbers close to \( 10^{12} \) or possibly \( 2 \cdot 10^{12} \) would be desirable.

**Results.**

As mentioned above we have computed the density \( \varrho(N) \) of numbers which require 5 cubes, by taking samples of varying lengths for different values of \( N \leq 4 \cdot 10^{11} \). The numerical values are presented in Table 1, and further we have plotted \( N \varrho(N) \cdot 10^{-6} \) together with \( \ln \ln \varrho(N)^{-1} \) against \( \ln N \) in Fig. 1. The shape of curve 1 indicates that \( \varrho(N) \) drops off considerably faster than \( N^{-1} \), and this would imply that the number of elements in the class \( C_5 \) is finite. A numerical