A Fast Probabilistic Algorithm to represent a Number as Sum of Four Squares

Klaus Huber, FI 17d
Deutsche Bundespost Telekom
Research Institute
P.O.Box 10 00 03
6100 Darmstadt
Germany

Abstract

In this contribution an efficient algorithm is given to represent an integer N as sum of four squares. The expected running time of the algorithm is $O(\log^5 N)$.

Index Terms — Number Theory, Sum of four squares, Quaternions.

1 Introduction

A famous century-old theorem in number theory states that every positive integer can be represented as sum of four squares (see e.g.[1], p.302). In this correspondence we present a fast probabilistic algorithm to find such a representation. The base of the algorithm are primes $p \equiv 1 \bmod 4$ and the Euclidean algorithm for quaternions.

2 Representing a Number as Sum of four Squares

To represent an integer N as sum of four squares, we can, without loss of generality, limit ourselves to those numbers which are either odd or twice an odd number. Representations for all other positive integers can be obtained in an obvious way from these numbers.

2.1 The case $N = 2 \cdot n$ with n odd

We first give an algorithm for the case $N = 2 \cdot n$ with n odd and $n \notin \{1, 3, 5, 7, 13, 19, 31\}$. The algorithm (A I) consists of two steps:

A I

1.) Find a representation $N = 2 \cdot n = p_1 + p_2$, with two primes $p_{1/2} \equiv 1 \bmod 4$.

2.) Find $p_1 = a^2 + b^2$ and $p_2 = c^2 + d^2 \Rightarrow N = a^2 + b^2 + c^2 + d^2$.

Both steps of algorithm A I can be carried out quite efficiently. To get a representation of $N = 2n = p_1 + p_2$ we can use the following algorithm (A II):

A II

1.) Select a startingvalue $s \in \{1, 2, \ldots, n\}$ at random.

2.) Find the next prime $p_1 \equiv 1 \bmod 4$, such that $p_2 = 2 \cdot n - p_1$ is also prime.

This search usually will not take very long. The longest running time of algorithm A II can be expected if the size of the starting value s is about n. By the prime number theorem about every $\log(n)$-th number around n is prime, hence it will take about $O(\log^2 n)$ numbers until we will come across two primes p_1 and p_2. (To recognize the primes we will of course use a fast probabilistic primality test algorithm.)

For the second step of algorithm A I we use a known algorithm. A detailed exposition of this algorithm can be found in [3]. To represent $p \equiv 1 \bmod 4$ as sum of two squares ($p = a^2 + b^2$), the algorithm is as follows:

A III

1.) Find x such that $x^2 \equiv -1 \bmod p$.
(If q_{nr} is a quadratic nonresidue of p then $x \equiv q_{nr}^{(p-1)/4} \bmod p$.)

2.) Apply the Euclidean algorithm to p and x;
the first two remainders less than $\sqrt{p}$ are a and b.

For details see e.g. Wagons paper. If the following conjecture is true, then algorithm A I works for $N = 2n$ for all odd $n \notin \{1, 3, 5, 7, 13, 19, 31\}$.

Conjecture 1 *Let n be an odd number with $n \notin \{1, 3, 5, 7, 13, 19, 31\}$. Then any number $N = 2 \cdot n$ can be represented as sum $N = p_1 + p_2$ with p_1, p_2 primes of the form $p_{1/2} \equiv 1 \bmod 4$.*

2.2 Representing an odd Number as Sum of four Squares

Representing an odd number n as sum of four squares is only slightly more difficult. To do this we first represent $2n$ as sum of four squares using algorithm A I. A representation of n as sum of four squares can then be found by use of the Euclidean algorithm with quaternions. For quaternions see ([1] pp.303-310, [2]) and the appendix below. If the theory of quaternions is developed according to Hurwitz, then any two integral quaternions have a greatest common (right-hand) divisor (cf. the notes in [1], pp.315-316) and we get:

A IV

1.) Select a startingvalue $s \in \{1, 2, \ldots, n\}$ at random.

2.) Determine $N = 2 \cdot n = a^2 + b^2 + c^2 + d^2$ using algorithm A I.

3.) Set $\alpha = a + i \cdot b + j \cdot c + k \cdot d$ and compute $\beta = \gcd(n, \alpha)$.

4.) If $N(\beta) = n$ then $\beta = b_0 + i \cdot b_1 + j \cdot b_2 + k \cdot b_3 \Rightarrow$ output $n = b_0^2 + b_1^2 + b_2^2 + b_3^2$. Else go to 1.).

Example: Using the above algorithms it took about a minute on a 486/50 MHz PC to find a representation of

$$\begin{aligned} 2^{128} + 1 &= a^2 + b^2 + c^2 + d^2 \\ \text{namely} \quad a &= 2078596997814651645 \\ b &= 14062795582136156754 \\ c &= 11754187844410685804 \\ d &= 196595968865712250 \ . \end{aligned}$$

3 Conclusion

A probabilistic algorithm has been given to determine a representation of an integer as sum of four squares. The running time of the algorithm is essentially given by the time to run $O(\log^2(N))$ primality tests. Counting a primality test with $O(\log^3(N))$, we get a time complexity of $O(\log^5(N))$. As all the other steps of algorithms A I-IV are faster, the overall complexity is expected to be $O(\log^5(N))$.

A Quaternions

To make this paper self-contained we give the most important properties of quaternions following [1]. Quaternions or hyper-complex numbers are extensions of real and complex numbers. A quaternion α has four coordinates:

$$\alpha = a_0 + a_1 \cdot i + a_2 \cdot j + a_3 \cdot k = (a_0, a_1, a_2, a_3) \quad \text{with } a_i \text{ real numbers.}$$

Addition and subtraction is done componentwise

$$\alpha + \beta = (a_0, a_1, a_2, a_3) + (b_0, b_1, b_2, b_3) = (a_0 + b_0, a_1 + b_1, a_2 + b_2, a_3 + b_3) \ .$$

Multiplication is associative and distributive but in general not commutative. It is defined by

$$i^2 = j^2 = k^2 = -1; \ \ i \cdot j = k = -j \cdot i; \ \ j \cdot k = i = -k \cdot j; \ \ k \cdot i = j = -i \cdot k \ .$$

Hence multiplication of two quaternions is done by:

$$\begin{aligned} \alpha \cdot \beta &= (a_0, a_1, a_2, a_3) \cdot (b_0, b_1, b_2, b_3) \\ &= (c_0, c_1, c_2, c_3) \\ \text{where} \quad c_0 &= a_0 \cdot b_0 - a_1 \cdot b_1 - a_2 \cdot b_2 - a_3 \cdot b_3 \\ c_1 &= a_0 \cdot b_1 + a_1 \cdot b_0 + a_2 \cdot b_3 - a_3 \cdot b_2 \\ c_2 &= a_0 \cdot b_2 - a_1 \cdot b_3 + a_2 \cdot b_0 + a_3 \cdot b_1 \\ c_3 &= a_0 \cdot b_3 + a_1 \cdot b_2 - a_2 \cdot b_1 + a_3 \cdot b_0 \end{aligned}$$

The conjugate α^* of a quaternion $\alpha = a_0 + a_1 \cdot i + a_2 \cdot j + a_3 \cdot k$ is defined by

$$\alpha^* = a_0 - a_1 \cdot i - a_2 \cdot j - a_3 \cdot k \ .$$

We then get the norm $N(\alpha)$ of a quaternion as

$$N(\alpha) = \alpha \cdot \alpha^* = a_0^2 + a_1^2 + a_2^2 + a_3^2 \ .$$

A quaternion α is called integral quaternion if a_0, a_1, a_2, a_3 are either (i) all integers or (ii) all halves of integers. The norm of an integral quaternion is an integer. It also follows that

$$(\alpha \cdot \beta)^* = \beta^* \cdot \alpha^* \ ,$$

and from this

$$N(\alpha \cdot \beta) = N(\alpha) \cdot N(\beta) \ .$$

Now we can define α^{-1} by:

$$\alpha^{-1} = \alpha^* / N(\alpha) \ .$$

The quaterions for which $N(\alpha) = N(\alpha^{-1})$ are integers are called unities, and its norm is 1. There are exactly 24 unities, namely $\pm 1, \pm i, \pm j, \pm k, \frac{1}{2}(\pm 1, \pm i \pm j \pm k)$. Any integral quaternion can be represented as

$$k_0 \cdot \rho + k_1 \cdot i + k_2 \cdot j + k_3 \cdot k \quad \text{with } k_i \text{ integers,}$$

where

$$\rho = \frac{1}{2}(1 + i + j + k) \ .$$

The product of any two integral quaternions is integral. If ϵ is a unity, then $\epsilon \cdot \alpha$ and $\alpha \cdot \epsilon$ are said to be associates of α. If $\gamma = \alpha \cdot \beta$, then α is called left-hand divisor of γ and β right-hand divisor. The following theorems are also of interest for this correspondence (Theorems 371 and 373 in Hardy and Wrights book):

Theorem 1 *If α is an integral quaternion, then one at least of its assosciates has integral coordinates; and if $N(\alpha)$ is odd, then at least one of its associates has non-integral coordinates.*

Theorem 2 *If α and β are integral quaternions, and $\beta \neq 0$ then there are integral quaternions λ and γ such that*

$$\alpha = \lambda \cdot \beta + \gamma, \qquad N(\gamma) < N(\beta).$$

The last theorem is a central theorem as it defines the Euclidean algorithm for quaternions. Let [.] denote rounding to the closest integer, then we can define rounding of quaternions as:

$$[(x_0, x_1, x_2, x_3)] = (\delta, [x_1 - \delta], [x_2 - \delta], [x_3 - \delta]) \quad \text{with } \delta = [2 \cdot x_0]/2.$$

Hence in theorem 2 above we can set $\gamma = \alpha \bmod \beta = \alpha - [\alpha \cdot \pi^* / N(\pi)] \cdot \pi$.

Acknowledgement: I would like to thank A.Sparschuh for helpful comments on this correspondence.

References

[1] G.H.Hardy, E.M.Wright, "An introduction to the theory of numbers", fifth edition, Oxford 1979.

[2] A.Hurwitz, "Vorlesungen über die Zahlentheorie der Quaternionen", Verlag Julius Springer, Berlin 1919.

[3] S.Wagon, "The Euclidean Algorithm Strikes again", *American Mathematical Monthly*, Vol.97, No.2, 1990, pp.125-129.