Going back to your **roots**

Mike Mudge presents a square-root algorithm suitable for newcomers to this column, and rational approximations to square roots of integers should crank your brains into gear.

This month's theme is based upon an article by P. Shiu in *Mathematical Spectrum*, vol 4, no. 1, 1971/72, pp26.30.

To approximate to the square root of N, i.e. $N^{1/2}$, where N is a given square-free integer, first seek an integer solution m_0,n_0 of the equation $n(n+1) = Nm_2$. Then observe that this equation is also satisfied by the sequence:

$$m_1 = 2m_0(2n_0 + 1),\ n_1 = 4n_0(n_0 + 1)$$
$$m_2 = 2m_1(2n_1 + 1),\ n_2 = 4n_1(n_1 + 1) \ldots\ldots$$
$$m_{k+1} = 2m_k(2n_k + 1);\ n_{k+1} = 4n_k(n_k + 1)$$

While $n^{1/2}$ is approximated to (from above) by:

$$r_k = (2n_k + 1)/(2m_k)$$

e.g. If N = 2 we may choose $m_0 = n_0 = 1$ when the above recurrence relations yield:

$$m_1 = 6,\ n_1 = 8;\ m_2 = 204,\ n_2 = 288;\ m_3 = 235416,\ n_3 = 332928;\ m_4 = 313506783024,\ n_4 = 443365544448;$$

These numbers yield an n_4 which differs from $2^{1/2}$ by less than 10^{-24}. We have an approximation to square root of two correct to 24 decimal places!

PROBLEM ROOTS. Implement the Shiu Algorithm to initially find an m_0,n_0 pair for a given N, followed by the sequence of fractions (r_k) which approximate to $N_{1/2}$.

PROBLEM ROOTS*. Attempt to generalise this process to cuberoots and beyond, comparing its computational efficiency with other, more commonly used algorithms.

An 'almost incomputable' function

The recently-published text by Arnold R. Krommer and Christoph W. Ueberhuber, "Numerical Integration on Advanced Computer Systems", Lecture Notes in Computer Science 848, Springer-Verlag 1994, has a 268-item bibliography and a commensurate body of text, an altogether outstanding publication. On page 186, readers are introduced to the function

$$f(x) = 3x^2 + (PI)^{-4}\log((PI - x)^2) + 1$$

which has a pole at x = PI, by which we mean that its value is unbounded below (infinitely large and negative) at x = PI.

Since clearly the function is positive over very large ranges of x, it must have two zeros (at least) one on either side of the pole. However, if it is sampled at ALL MACHINE NUMBERS differing by 2^{-54} (approximately 5.6×10^{-17}) and corresponding to Double Precision IEEE Arithmetic, the pole cannot be detected and indeed no negative values are generated.

PROBLEM FUNCTION. Devise a means of exhibiting either graphically or numerically the true behaviour of this function. Such revelations may come from a sophisticated programming technique, or by the use of some algebraic transformation?

PROBLEM FUNCTION*. Indicate some other functions which exhibit this type of behaviour. Do any of them have a practical application?

● Any investigations of the above problems may be sent to Mike Mudge, 22 Gors Fach, Pwll-Trap, St. Clears, Carmarthenshire SA33 4AQ, tel 01994 231121, to arrive by 1st February 1997. All material received will be judged using suitable subjective criteria and a prize will be awarded by Mike Mudge to the "best" entry arriving by the closing date. Each contribution should contain brief descriptions of the hardware and coding used, together with run times and a summary of the results obtained. (SAE for return entries, please.)

Report on Numbers Count -155- 'Pounding the beat', *PCW* March 1996

All aspects of this column generated interesting responses. The "Full Houses" or "Prime Decades" upto 100000 numbering 40 (less the two inadmissable 11,7,5,3 and 13,11,7,5) these consist of the 37 regular ones and the anomalous 2,3,5,7. Alan Cox obtained these with UBASIC and its NXTPRM(x) function (can any reader tell us how this function works?) in 48 seconds on a "slow 8086", while Hugh Spence used an AMD 585 running at 133MHz in Modula-2 ("the last Topspeed incarnation") to reproduce the results in 9.5 seconds.

Problem GS produced responses, including one from Tim Thorp who refers to Donald Knuth's *The Art of Programming* where the base three (being the integer nearest to e) is "in some sense" optimal for numerical operations.

This month's winner is David Price of 13 The Hall Close, Dunchurch, Rugby, Warwickshire CV22 6NP: his representation of numbers in various bases extended to complex bases and involved Fortran in double precision on a 486 PC. Altogether a commendable mixture of algebra/calculator arithmetic and programming.

•PCW Contributions Welcome

Mike Mudge welcomes correspondence from readers on any subject within the areas of number theory and computational mathematics, together with suggested subject areas and/or specific problems for future Numbers Count articles.