Re: rapidly converging rational sqrt

Source: http://sci.tech−archive.net/Archive/sci.math.research/2004−10/0252.html

---

From: David Loeffler (*loefflerdavid_at_hotmail.com*)
Date: 10/28/04

Date: Thu, 28 Oct 2004 20:30:08 +0000 (UTC)

TheBean wrote:

> Below is a description of an algorithm which, with
> each iteration, will double the number of significant
> digits in the computation of a rational square root
> approximation.

[snip]

> Here is an example run computing the sqrt of 1973 for 1 to 10
> iterations of the algorithm. Note that after iteration 6 we have
> more than 100 significant digits.
>
> 1 : 44.4184 |
552114124148567022233646060917619843207813905667651972754144711476673949363834982650044981364862
> 2 : 44.41846462 |
881467219505665982727838995343272680250852496023827788068578570854493706272852746588967910 48
> 3 : 44.418464629025618764 |
275243123255720402409859148429709034223054860565966164701660860057950498109567655 8
> 4 : 44.4184646290256187643810796574090605395 |
68332729413549611124685085733754437910815511310326015577559783 4
> 5 : 44.41846462902561876438107965740906053959497442704659903610246205761940066180 |
26805283703947239542091268
> 6 :
44.41846462902561876438107965740906053959497442704659903610246205761940066180436869171473600589 1
> 7 :
44.41846462902561876438107965740906053959497442704659903610246205761940066180436869171473600589 1
> 8 :
44.41846462902561876438107965740906053959497442704659903610246205761940066180436869171473600589 1
> 9 :
44.41846462902561876438107965740906053959497442704659903610246205761940066180436869171473600589 1
> 10 :
44.41846462902561876438107965740906053959497442704659903610246205761940066180436869171473600589 1

I don't wish to denigrate your algorithm unduly, but square root algorithms with this rate of convergence have been known for millennia. The iteration x_{n+1} = 1/2 (x_n + C/x_n), which is believed to have been known to the ancient Babylonians, yields the following output if we take C = 1973 and start

it at 44 (since your algorithm presupposes we know isqrt(C) this seems a fair comparison.) I have added vertical bars to indicate the correct portion of each decimal expansion; I have done the same for the quoted output from your algorithm above.

1 : 44.4 |
2045454545454545454545454545454545454545454545454545454545454545454545454545454545454545454545454545454545454545454545454545454
2 : 44.4184646 |
735970603967534128700667457382729830926300611642131212353775669201609339752087257843205655 9
3 : 44.418464629025618786 |
743552378903683842335292967543660539710480328526697275409294551866285646468618 10
4 : 44.4184646290256187643810796574090605 |
452241670431822943946992852924285398121711157344287940969052547 2
5 : 44.418464629025618764381079657409060539594974427046599036102462057619400661804368691714736005891 6
6 :
44.418464629025618764381079657409060539594974427046599036102462057619400661804368691714736005891

So like your algorithm this also produces 100 digits in 6 iterations, and it is easy to prove (since this is a special case of Newton's method for a general function) that convergence is quadratic in general.

In short, your algorithm is interesting but it doesn't outperform the standard algorithms for square roots.

Yours,

David Loeffler

(student, Trinity College,
University of Cambridge, UK)