Iterations Improving Accuracy

Using FACTORIZATION mode, values are maintained as ratios of prime factors. By selecting algorithms that are based on series of integer ratios it is possible to compute irrational values out to arbitrary numbers of significant digits providing that the numerator and denominator are allowed to grow to arbitrary numbers of digits. Having implemented this methodology the remaining issue is how fast the series converges since this impacts how many terms are required to give an accurate number of how many digits. The sample below uses the Ramanujan series to calculate 1/pi. This series converges very rapidly. Computation of 10 terms of the series gives a calculation of 85 digits of pi. The representation of the ratio has numerator and denominator values that are significantly longer than 85 digits

Script Source

```
SCRIPTPRINT ComputePiRamanujan.txt
Reading... C:\Users\Michael\workspace\CalcLib\scripts\ComputePiRamanujan.txt


x = 2

READ sqrtCompute.txt

// requires 7 iterations of Newton's method
// sqrt(2) = 1.4142_13562_37309_50488_01688_72420_96980_78569_67187_53769_48073_17667_97379_90732_47846_21070_38850_38753_43276 // 95 digits
//        = 1.4142_13562_37309_50488_01688_72420_96980_78569_67187_53769_48073_17667_97379_90732_47846_21070_38850_38753_43276 // per OEIS A002193
//        = 4946041176255201878775086487573351061418968498177 / 3497379255757941172020851852070562919437964212608
//        = ( 7681 * 1492993 * 4313027139808909476126333579645696967 69 ) / ( 2^7 * 3 * 17 * 257 * 577 * 1409 * 11777 * 665857 * 2393857 * 2448769 * 55780318173953 )

radical2   = sqrtx


// 1 / pi = ( 2 * sqrt(2) / 9801 ) * SIGMA [0 <= k <= INFINITY] ( (4*k)! * (1103 + 26390*k) / ((k!)^4 * 396 ^ (4*k)) )


!! series (n) = SIGMA [0 <= k <= n] ( (4*k)! * (1103 + 26390*k) / ((k!)^4 * 396 ^ (4*k)) )

s = series(10)

// s = 1103.0000268319745734638134088821330563369364054638896 7
//   = 387243007645026806445679460905436118145941214813373553635390935938527107809849309415162210716479714682763731082041669 77
                  2060777126802719277591020187485184860708456341122994882635648594364069184119994702 75
//    / 3510815940388163209292975628161374616891448276678347860097892396840833464993082052835202165786327128761649621423675210573 98
                  5612671493685721632947900318202137813691668941941229874985670718736078943027 2
//   = ( 5^2 * 7^3 * 13 * 23 * 29 * 31 * 37 * 41 * 43 * 47 * 53 * 59 * 61 * 67 * 71 * 73 * 79 * 83 * 191 * 1451 *
                  455246892847718757405046391883875553050178816429535657411420891346193556800559067130956704 3
                              4403864087806034470850422555296422129968289083320181488913830968277491841459 )
//    / ( 2^154 * 3^153 * 11^77 )
PRETTYPRINT s 50


c = 2 * radical2 / 9801

// c = 0.000288585565222547709172878017387960020114207096 30
//   = 4946041176255201878775086487573351061418968498177 / 1713890704284179071348818450107179358670574362388550 4
//   = ( 7681 * 1492993 * 4313027139808909476126333579645696967 69 ) / ( 2^6 * 3^5 * 11^2 * 17 * 257 * 577 * 1409 * 11777 * 665857 * 2393857 * 2448769 *
55780318173953 )
PRETTYPRINT c 50


piApproximation = 1 / (c*s)

PRETTYPRINT piApproximation  86
// piApproximation = 3.14159_26535_89793_23846_26433_83279_50288_41971_69399_37510_58209_74944_59230_78164_06286_20899_86280_3
//               = 1063508829885643955702523868829133008241125216586197637972442088222758794531096590230104814300659761592544166161243665275176690233286697474699317739 52
//               / 338525374596355722607287291604043659118422478307131374205586143760945503544628107920145576571186806698919567447328724392872446016127314023761664676 85
//               = ( 2^80 * 3^83 * 11^39 * 257 * 577 * 1409 * 11777 * 665857 * 2393857 * 2448769 * 55780318173953 )
//                 / ( 5 * 7^2 * 13 * 19 * 23 * 29 * 31 * 37 * 41 * 43 * 7681 * 1492993 * 4313027139808909476126333579645696967 69 *
                  838549873976734283142691189373245057507371516097566767496640304132808663454176616756 21 )
//           pi =
                 3.14159_26535_89793_23846_26433_83279_50288_41971_69399_37510_58209_74944_59230_78164_06286_20899_86280_34825_
                                 34211_70679_82148_08651_32823_06647_09384_46095_50582_23172_53594_08128
//                 5    10   15   20   25   30   35   40   45   50   55   60   65   70   75   80   85   90   95  100  105  110  115  120  125  130  135  140  145  150
```

Script Execution

READ ComputePiRamanujan.txt
Reading... C:\Users\Michael\workspace\CalcLib\scripts\sqrtCompute.txt

sqrt_polynomial =

2 - x^2

sqrt_poly_derivative =

- 2 * x

Reading... C:\Users\Michael\workspace\CalcLib\scripts\SqrtIteration.txt

sqrt_iteration =

-0.5000000000000000 = -1 / 2 = (- 1) / (2)

sqrtx_squared =

2.2500000000000000 = 9 / 4 = (3^2) / (2^2)

sqrtx =

1.5000000000000000 = 3 / 2 = (3) / (2)

sqrt_iteration =

0.0833333333333334 = 1 / 12 = (1) / (2^2 * 3)

sqrtx_squared =

2.0069444444444445 = 289 / 144 = (17^2) / (2^4 * 3^2)

sqrtx =

1.4166666666666667 = 17 / 12 = (17) / (2^2 * 3)

sqrt_iteration =

0.0024509803921569 = 1 / 408 = (1) / (2^3 * 3 * 17)

sqrtx_squared =

2.0000060073048828 = 332929 / 166464 = (577^2) / (2^6 * 3^2 * 17^2)

sqrtx =

1.4142156862745099 = 577 / 408 = (577) / (2^3 * 3 * 17)

sqrt_iteration =

0.0000021238998199 = 1 / 470832 = (1) / (2^4 * 3 * 17 * 577)

sqrtx_squared =

2.0000000000045110 = 443365544449 / 221682772224 = (665857^2) / (2^8 * 3^2 * 17^2 * 577^2)

sqrtx =

1.4142135623746900 = 665857 / 470832 = (665857) / (2^4 * 3 * 17 * 577)

sqrt_iteration =

1.5948618246059560E-12 = 1 / 627013566048 = (1) / (2^5 * 3 * 17 * 577 * 665857)

sqrtx_squared =

2.0000000000000001 = 786292024016459316676609 / 393146012008229658338304 = (257^2 * 1409^2 * 2448769^2) / (2^10 * 3^2 * 17^2 * 577^2 * 665857^2)

sqrtx =

1.4142135623730951 = 886731088897 / 627013566048 = (257 * 1409 * 2448769) / (2^5 * 3 * 17 * 577 * 665857)

sqrt_iteration =

8.9929283216504540E-25 = 1 / 1111984844349868137938112 = (1) / (2^6 * 3 * 17 * 257 * 577 * 1409 * 665857 * 2448769)

sqrtx_squared =

2.0000000000000001 = 2473020588127600939387543243786675530709484249089 / 1236510294063800469693771621893337765354742124544 =
(11777^2 * 2393857^2 * 55780318173953^2) / (2^12 * 3^2 * 17^2 * 257^2 * 577^2 * 1409^2 * 665857^2 * 2448769^2)

sqrtx =

1.4142135623730951 = 1572584048032918633353217 / 1111984844349868137938112 =
(11777 * 2393857 * 55780318173953) / (2^6 * 3 * 17 * 257 * 577 * 1409 * 665857 * 2448769)

sqrt_iteration =

2.8592838433339520E-49 = 1 / 349737925575794117202085185207056291943796421 2608 =
(1) / (2^7 * 3 * 17 * 257 * 577 * 1409 * 11777 * 665857 * 2393857 * 2448769 * 55780318173953)

sqrtx_squared =

2.0000000000000001 =
244633233172119409772934044199283472792460911293342685727738504492736114554842536340586908523 23329 /
1223166165860597048864670220996417363962304556466713428638692522463680572774212681702934542616 1664 =
(7681^2 * 1492993^2 * 43130271398089094761263335796456969 6769^2) /
(2^14 * 3^2 * 17^2 * 257^2 * 577^2 * 1409^2 * 11777^2 * 665857^2 * 2393857^2 * 2448769^2 * 55780318173953^2)

sqrtx =

1.4142135623730951 = 494604117625520187877508648757335106141896849 8177 / 349737925575794117202085185207056291943796421 2608 =
(7681 * 1492993 * 43130271398089094761263335796456969 6769) / (2^7 * 3 * 17 * 257 * 577 * 1409 * 11777 * 665857 * 2393857 * 2448769 * 55780318173953)

Script interrupted in iteration 7
Assertion "Convergence Complete" has been validated, TOLERANCE >|| sqrt_iteration
*** Script has terminated

s =

1103.000026831974573463813408882133056336936405463 88967 =
116354295547844200479625540962705305445031010498388307062857519290687784871920308555177681218916232885 /
105488932654005296215338934589125951145247950655030308540653845613856673595635654924408113432887296 =
(5 * 7^2 * 13 * 17 * 19 * 23 * 29 * 31 * 37 * 41 * 43 * 838549873976734283142691189373245057507371516097566767496640304132808663454176616 75621) / (2^74 * 3^78 * 11^37)

c =

0.000288585565222547709172878017387960020114207 09630 = 494604117625520187877508648757335106141896849 8177 /
1713890704284179071348818450107179358670574362 3885504 =
(7681 * 1492993 * 43130271398089094761263335796456969 6769) / (2^6 * 3^5 * 11^2 * 17 * 257 * 577 * 1409 * 11777 * 665857 * 2393857 * 2448769 * 55780318173953)

piApproximation =

3.141592653589793238462643383279502884197169399375105820974944592307816406286208998 62804 =
1063508829885643955702523868829133008241125216586197637972442088222758794531096590230104814300659761592544166161243665275176690233286697474699317 73952 /
338525374596355722607287291604043659118422478307131374205586143760945503544628107920145576571186806698919567447328724392872446016127314023761664 67685 =
(2^80 * 3^83 * 11^39 * 257 * 577 * 1409 * 11777 * 665857 * 2393857 * 2448769 * 55780318173953) /
(5 * 7^2 * 13 * 19 * 23 * 29 * 31 * 37 * 41 * 43 * 7681 * 1492993 * 43130271398089094761263335796456969 6769 *
838549873976734283142691189373245057507371516097566767496640304132808663454176616 75621)

Represent Irrational Numbers with Primes

Compute irrational values out to arbitrary numbers of significant digits. This can be seen as an alternate way to describe an irrational number that carries additional attributes such as the specification of the equation and the number of terms that have been used to compute it. The sample below uses the Ramanujan series to calculate 1/pi. This series converges very rapidly. Using just 2 iterations of the summation results in an accurate double float equivalent calculation.

Script Source

```
1/pi = 2*\2 / 9801 * ( SUMMATION [0 <= k <= INFINITY] ( (4*k)! * (1103 + 26390*k) / ((k!)^4 * 396 ^ (4*k)) ) )

!! series (n) = SUMMATION [0 <= k <= n] ( (4*k)! * (1103 + 26390*k) / ((k!)^4 * 396 ^ (4*k)) )

calc series 1
// 1103.0000268319743490 = 1130173253125 / 1024635744 = ( 5^5 * 7 * 4423 * 11681 ) / ( 2^5 * 3^7 * 11^4 )

calc sqrt 2
// 1.4142135623730951 = 886731088897 / 627013566048 = ( 257 * 1409 * 2448769 ) / ( 2^5 * 3 * 17 * 577 * 665857 )

c = 2 * sqrt 2 / 9801

calc 1 / (c * series 1)
```

Script Execution (ComputePrimePI.txt)

```
3.141592653589793 =
        3148377737809732379398656 / 1002159759385796058053125 =
        (
                2^9 * 3^12 * 11^6 * 17 * 577 * 665857
        )
                /
        (
                5^5 * 7 * 257 * 1409 * 4423 * 11681 * 2448769
        )
```

As mentioned above we can attribute this result as a 2 term approximation of the calculation using the Ramanujan series. This can be taken to imply that additional precision can be accomplished by using this as an interim result to which additional terms can be added