OXFORD JOURNAL
OF STUDENT
SCHOLARSHIP

Algorithmic Rearrangements of the Alternating Harmonic Series

Aditya Srivastava

Contact email: aditya.15.srivastava@gmail.com

ABSTRACT

Riemann's Rearrangement Theorem shows that any conditionally convergent series can be rearranged to sum to any target real number. However, standard constructive proofs do not detail the structural characteristics of the block patterns required for prescribed numerical tolerances. In this paper, I develop a Python implementation using continued fractions to select static block patterns—alternating between p positive terms and n negative terms—that approximate target sums for the alternating harmonic series. I evaluate these rearrangements by tracking selected block sizes and block balance metrics $C = \frac{1}{2}\left|\frac{1}{p} - \frac{1}{n}\right|$, comparing algorithmically selected convergents with minimal admissible pairs across different target regions. The results demonstrate that while exact clean representations exist for clean analytical logarithmic targets (e.g., a perfect $1{:}\,2$ ratio for $\frac{1}{2}ln2$), nearby rounded decimal target values require vastly larger block sizes (such as $p = 9466, n = 18931$ at $S = 0.3466$). This happens because these targets are just outside the tolerance window around $\frac{1}{2}ln2$, and stopping at full convergents makes the blocks even larger (the smallest valid pair is $6867{:}\,13733$). Furthermore, negative targets require heavily imbalanced blocks ($p \ll n$), driving C higher. Overall, this work establishes a quantitative framework connecting Riemann rearrangements, rational approximation, and block size structural attributes.

INTRODUCTION

Rearrangements of infinite series reveal a fundamental distinction between finite addition and infinite summation. For a finite collection of real numbers, changing the order of the terms does not affect the result. An infinite series, however, is defined through the limit of its partial sums. Rearranging its terms changes the sequence of partial sums and may therefore alter its limiting behavior.

Let $\sum_{k=1}^{\infty} a_k$ be a convergent series. It is absolutely convergent if

$$\sum_{k=1}^{\infty} \left|a_k\right| < \infty, \tag{1}$$

and conditionally convergent if $\sum a_k$ converges while $\sum |a_k|$ diverges. Every rearrangement of an absolutely convergent series converges to the original sum. Standard treatments of these results can be found in Knopp and Rudin .The possible effects of rearrangement are characterised by the Riemann Rearrangement Theorem . The theorem states that the terms of any conditionally convergent real series can be rearranged to converge to an arbitrarily prescribed real number. The alternating harmonic series is the standard example used to demonstrate this phenomenon:

$$\sum_{k=1}^{\infty} \frac{(-1)^{k+1}}{k} = 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \cdots = ln2. \tag{2}$$

Nineteenth-century work by Pringsheim , later extended by Cowen, Davidson, and Kaufman as well as Riddle , considered rearrangements formed by repeatedly selecting fixed blocks of p positive terms and n negative terms. They showed that the resulting sum is determined by the ratio p/n and is given by

$$S_{p,n} = ln2 + \frac{1}{2} ln\left(\frac{p}{n}\right). \tag{3}$$

Equation (3) can be rewritten explicitly as

$$S_{p,n} = ln\left(2\sqrt{\frac{p}{n}}\right), \tag{4}$$

which directly yields the required exponential ratio expression

$$\frac{p}{n} = \frac{1}{4} e^{2S_{p,n}}. \tag{5}$$

Fixed-block rearrangements can therefore produce an exact target value when the corresponding exponential quantity is rational.

The present work considers the computational problem of constructing structured fixed-block rearrangements for an arbitrary target S and a prescribed error tolerance $\varepsilon > 0$. Continued fractions provide a natural framework for generating rational approximations to $G(S) = \frac{1}{4} e^{2S}$. Transforming the target interval $S \pm \varepsilon$ directly into the exponential domain requires the admissible ratio to satisfy

$$\frac{1}{4} e^{2(S-\varepsilon)} \leq \frac{p}{n} \leq \frac{1}{4} e^{2(S+\varepsilon)}. \tag{6}$$

This study evaluates a continued-fraction algorithm for selecting suitable values of p and n. The principal objective is to analyze the structural properties, minimal admissible pairs, and block metrics of these rearrangements across different regions of the real line.

METHODS

The foundational setup builds upon the framework analyzed by Pringsheim and Riddle .

Fixed-block rearrangements

Consider the alternating harmonic series where positive and negative terms are written as

$$u_j = \frac{1}{2j-1} \quad and \quad v_j = -\frac{1}{2j} \quad (j \geq 1).$$

For fixed positive integers p and n, a (p, n)-block rearrangement repeatedly selects p positive terms followed by n negative terms.

Theorem 1 (Limit of a fixed-block rearrangement). *Let $p, n \in N$. The (p, n)-block rearrangement of the alternating harmonic series converges to $S_{p,n} = ln2 + \frac{1}{2} ln\left(\frac{p}{n}\right)$.*

Proof. After m complete blocks, the rearrangement has used mp positive terms and mn negative terms. Its partial sum at the boundary of the m-th block is:

$$\begin{aligned} S_m(p, n) &= \sum_{j=1}^{mp} \frac{1}{2j-1} - \sum_{j=1}^{mn} \frac{1}{2j} \\ &= H_{2mp-1} - \frac{1}{2} H_{mp-1} - \frac{1}{2} H_{mn}, \end{aligned}$$

where $H_q = lnq + \gamma + o(1)$ as $q \to \infty$. Expanding the harmonic numbers yields:

$$\begin{aligned} S_m(p, n) &= (ln(2mp - 1) + \gamma) - \frac{1}{2}(ln(mp - 1) + \gamma) - \frac{1}{2}(ln(mn) + \gamma) + o(1) \\ &= ln(2mp) - \frac{1}{2} ln(mp) - \frac{1}{2} ln(mn) + o(1) \\ &= ln2 + lnm + lnp - \frac{1}{2} lnm - \frac{1}{2} lnp - \frac{1}{2} lnm - \frac{1}{2} lnn + o(1). \end{aligned}$$

Grouping the logarithmic terms reveals explicit cancellation:

$$\begin{aligned} S_m(p, n) &= ln2 + \left(1 - \frac{1}{2} - \frac{1}{2}\right) ln(m) + \left(1 - \frac{1}{2}\right) ln(p) - \frac{1}{2} ln(n) + o(1) \\ &= ln2 + 0 \cdot ln(m) + \frac{1}{2} ln(p) - \frac{1}{2} lnn + o(1). \end{aligned}$$

The $ln(m)$ and γ terms cancel completely, leaving:

$$\begin{aligned} \lim_{m \to \infty} S_m(p, n) &= ln2 + \frac{1}{2} ln(p) - \frac{1}{2} lnn \\ &= ln2 + \frac{1}{2} ln\left(\frac{p}{n}\right). \end{aligned}$$

In block $m \geq 2$, each of the p positive terms is less than $\frac{1}{2(m-1)p}$ and each of the n negative terms is less than $\frac{1}{2(m-1)n}$ in size. So every partial sum inside the block stays within $\frac{1}{2(m-1)}$ of the sum at the end of the previous block, $S_{m-1}(p, n)$, and this goes to 0 as $m \to \infty$. This means all of the partial sums settle at the same limit. □

For a target S, exact fixed-block representation requires $\frac{p}{n} = G(S) = \frac{1}{4}e^{2S}$. When $G(S)$ is irrational, an admissible tolerance interval is defined by (6).

Continued-fraction approximation and minimal pairs

Let $G(S) = [a_0; a_1, a_2, \ldots]$ be the simple continued fraction of $G(S)$. Convergents p_k/n_k satisfy recurrence relations $p_k = a_k p_{k-1} + p_{k-2}$ and $n_k = a_k n_{k-1} + n_{k-2}$. By Theorems 16 and 17 of Khinchin , convergents provide the best rational approximations of the second kind, minimizing $|n\alpha - p|$ for a given denominator size.

Algorithm 1 (Continued-fraction block selection). Given target S and tolerance ε:

1. Compute $L = \frac{1}{4}e^{2(S-\varepsilon)}$ and $U = \frac{1}{4}e^{2(S+\varepsilon)}$.
2. Generate convergents p_k/n_k for $G(S) = \frac{1}{4}e^{2S}$.
3. Stop at the first index k where $L \le \frac{p_k}{n_k} \le U$, returning $p = p_k, n = n_k$.

Because Algorithm 1 terminates at full convergents, smaller admissible integer pairs—specifically semiconvergents—may also exist within $[L, U]$. We designate the pair with the smallest denominator n in $[L, U]$ as the *minimal admissible pair*; it is found by checking denominators one at a time until one works (Appendix 6).

Block metrics and tolerance criteria

To measure structural properties of selected blocks, we consider two metrics:

1. Block Balance Metric (C): Defined as $C = \frac{1}{2}\left|\frac{1}{p} - \frac{1}{n}\right|$. This metric characterizes the asymmetry between positive and negative term sizes within a block layout. Since $C \approx 1/(2p)$ when n is much bigger than p, C mostly just tracks the size of the smaller block. We keep it as a simple way to describe the shape of a block, and we use the block length $p + n$ (through ΔS below) to describe oscillation.
2. Intra-block Oscillation Amplitude: The partial sums overshoot and undershoot the target within each block. For a total of $N = m(p + n)$ terms, partial sum deviations within a block scale as $\Delta S \approx \frac{p+n}{2N}$. Large block sizes $(p + n)$ significantly increase intra-block oscillation amplitude before settling at block boundaries.

Justification of Varied Tolerances across Experiments

To evaluate the numerical characteristics of the selection mechanism under diverse constraints, different error tolerances ε are assigned across the target regions. Near the high-density logarithmic sum $\frac{1}{2}ln2$, a strict threshold $\varepsilon = 10^{-5}$ isolates the local divergence between decimal inputs and analytical limits. Across the baseline expanse $0.60 \le S \le 0.80$, dual evaluations at $\varepsilon = 10^{-4}$ and $\varepsilon = 10^{-6}$ delineate

how structural bounds expand under precision demands. For negative targets, a looser threshold $\varepsilon = 10^{-3}$ keeps the denominators (which get large because $p \ll n$) from becoming too big. Since ε is different in each experiment, comparing C and block size across regions should only be treated as a rough comparison.

Numerical Results

Three numerical experiments were performed using Python to evaluate block selection under Algorithm 1 and compare them with minimal admissible pairs. Minimal pairs were found by checking denominators one at a time and were then double-checked with high-precision arithmetic. Only some rows are shown in the tables below; the full tables are in Appendix 7.

Behavior near $\frac{1}{2}ln2$

The interval $0.3460 \le S \le 0.3470$ was examined with $\varepsilon = 10^{-5}$. At $S = \frac{1}{2}ln2 \approx 0.34657359$, the ideal ratio $G(S) = 1/2$ is exact for $1:2$. For rounded decimal targets, selected convergents and minimal pairs are shown in Table 1.

Selected numerical results near $\frac{1}{2}ln2$ ($\varepsilon = 10^{-5}$).

Target S	Algorithm 1 ($p:n$)	Minimal Pair ($p:n$)	C
0.3460	435 : 871	429 : 859	0.000575
0.3464	1439 : 2879	1362 : 2725	0.000174
0.3465	3396 : 6793	2991 : 5983	0.000074
0.3466	9466 : 18931	6867 : 13733	0.000026
0.3467	1977 : 3953	1833 : 3665	0.000126
0.3470	586 : 1171	574 : 1147	0.000426

At $S = 0.3466$, Algorithm 1 selected $9466:18931$, whereas the minimal admissible pair in the interval was $6867:13733$.

Behavior around $ln2$

The interval $0.60 \le S \le 0.80$ was examined for $\varepsilon = 10^{-4}$ and $\varepsilon = 10^{-6}$. Results are shown in Tables 2 and 3.

Selected results across $0.60 \le S \le 0.80$ ($\varepsilon = 10^{-4}$).

Target S	Algorithm 1 ($p:n$)	Minimal Pair ($p:n$)	C
0.6000	44 : 53	44 : 53	0.001929
0.6400	107 : 119	89 : 99	0.000471

Target S	Algorithm 1 (p: n)	Minimal Pair (p: n)	C
0.6900	158 : 159	154 : 155	0.000020
0.7000	73 : 72	73 : 72	0.000095
0.7600	8 : 7	8 : 7	0.008929
0.7700	372 : 319	260 : 223	0.000223

Selected results across $0.60 \leq S \leq 0.80$ *(*$\varepsilon = 10^{-6}$*).*

Target S	Algorithm 1 (p: n)	Minimal Pair (p: n)	C
0.6000	293 : 353	293 : 353	0.000290
0.6200	2501 : 2895	1479 : 1712	0.000027
0.6300	52 : 59	52 : 59	0.001141
0.6900	475 : 478	475 : 478	0.000007
0.7000	955 : 942	808 : 797	0.000007
0.7700	751 : 644	751 : 644	0.000111

For $S < ln2, n > p$; for $S > ln2, p > n$. The dominance crossover occurs between $S = 0.6900$ and $S = 0.7000$.

Negative target sums

Negative target sums were evaluated over $-1.5000 \leq S \leq -0.5500$ with $\varepsilon = 10^{-3}$. The results from Algorithm 1, along with the limiting sum $ln2 + \frac{1}{2}ln(p/n)$ and its error, are in Table 4.

Selected results for negative targets ($\varepsilon = 10^{-3}$*).*

Target S	Algorithm 1 (p: n)	Minimal Pair (p: n)	Limiting Sum	Error	C
-1.5000	2 : 161	2 : 161	-1.5010	0.0010	0.246894
-1.2000	10 : 441	6 : 265	-1.2001	0.0001	0.048866
-1.0000	2 : 59	2 : 59	-0.9990	0.0010	0.241525
-0.8000	5 : 99	5 : 99	-0.7997	0.0003	0.094949
-0.6500	3 : 44	3 : 44	-0.6496	0.0004	0.155303
-0.5500	1 : 12	1 : 12	-0.5493	0.0007	0.458333

All negative targets produce $n \gg p$. Because $1/n$ is small when $n \gg p$, $C \approx \frac{1}{2p}$, causing C to increase significantly as p shrinks for negative targets. Since ε is different here than in Experiments 1 and 2, the C values in this table should not be compared directly with the ones in Tables 1–3.

Figure 1 shows the block length $p + n$ for all three experiments, using more target values than the tables.

Block length $log_{10}(p + n)$ versus target S for the pair chosen by Algorithm 1 and the minimal pair. Dashed lines mark $\frac{1}{2}ln2$ and $ln2$.

DISCUSSION

Target sums, ratios, and algorithm artifacts

The fundamental ratio relation $p/n = \frac{1}{4}e^{2S}$ dictates structural shifts across S. Around $S = ln2$, $p/n \approx 1$. For negative targets, e^{2S} decays rapidly, forcing $p \ll n$. A crucial insight arises when comparing full convergents from Algorithm 1 with minimal admissible pairs. Stopping at the first full convergent often yields larger block sizes than strictly necessary to satisfy $L \leq p/n \leq U$. For instance, at $S = 0.3466$ ($\varepsilon = 10^{-5}$), Algorithm 1 selects $9466:18931$, whereas the minimal pair is $6867:13733$. The same up-and-down pattern shows up for the minimal pairs too (Figure 1), so it comes from how far the target is from $\frac{1}{2}ln2$ and not just from the algorithm. Stopping at full convergents only makes the blocks bigger.

Behavior near logarithmic targets

At $S = \frac{1}{2}ln2$, $G(S) = 1/2$ exactly $(1:2)$. However, for the rounded decimal target $S = 0.3466$, $\left|0.3466 - \frac{1}{2}ln2\right| \approx 2.64 \times 10^{-5} > \varepsilon = 10^{-5}$. Thus, the exact $1:2$ ratio falls outside the admissible interval, forcing the algorithm to select large integer representations to approximate the offset decimal.

Oscillation metrics vs. block balance

The metric $C = \frac{1}{2}\left|\frac{1}{p} - \frac{1}{n}\right|$ measures block term asymmetry rather than global convergence rate. For negative targets where p is small ($p = 1, 2, 3$), C is driven higher because $1/(2p)$ dominates. Within each block, maximum partial sum deviations scale with $(p + n)/(2N)$, demonstrating that larger block pairs create wider intra-block oscillations before settling at block boundaries.

CONCLUSION

This work provides an algorithmic analysis of fixed-block rearrangements of the alternating harmonic series. By transforming error tolerances into exact exponential ratio intervals, we established how continued-fraction convergents and minimal admissible pairs approximate target sums. The findings demonstrate that while exact block layouts exist for analytical logarithmic values, decimal approximations induce substantial increases in selected block sizes. Furthermore, negative target sums need very lopsided blocks ($p \ll n$), which makes C larger. However, these blocks are fairly short, so they do not make the oscillations within a block bigger. Future research will focus on dynamic block constructions with proven convergence bounds and systematic search strategies over semiconvergent structures via Stern-Brocot trees.

REFERENCES

Cowen, C. C., Davidson, K. R., and Kaufman, R. P. Rearranging the alternating harmonic series. Amer. Math. Monthly 87 (1980), no. 10, 817–819.

Dybskiy, Y. and Slutsky, K. Riemann Rearrangement Theorem for some types of convergence. *J. Math. Anal. Appl.* 373 (2011), no. 2, 605–613.

Freniche, F. J. On Riemann's rearrangement theorem for the alternating harmonic series. *Amer. Math. Monthly* 117 (2010), no. 5, 442–448.

Khinchin, A. Y. *Continued Fractions*. Dover Publications, Mineola, NY, 1997.

Kilmer, S. and Zheng, S. A generalized alternating harmonic series. *AIMS Math*. 6 (2021), no. 12, 13480–13487.

Knopp, K. *Theory and Application of Infinite Series.* Dover Publications, New York, 1990.

Pringsheim, A. Ueber die Werthveränderungen bedingt convergentor Reihen und Producte. *Mathematische Annalen* 22 (1883), no. 4, 455–502.

Riddle, L. H. Rearranging the alternating harmonic series. *Coll. Math.* J. 19 (1988), no. 5, 319–328.

Riemann, B. Ueber die Darstellbarkeit einer Function durch eine trigonometrische Reihe. *Abhandlungen der Königlichen Gesellschaft der Wissenschaften zu Göttingen* 13 (1867), 87–132.

Rudin, W. *Principles of Mathematical Analysis,* 3rd ed. McGraw–Hill, New York, 1976.

APPENDIX

The following Python script was utilized to run the continued-fraction block selection (Algorithm 1) and to find the minimal admissible pairs by checking denominators one at a time. It produced the results for all of the experiments:

```
import math

def efficiency_approx(target, lower_b, upper_b):
    ###Finds the p and n values
    p_0 = 1
    n_0 = 0
    p_minus1 = 0
    n_minus1 = 1
    current_val = target


    while True:
        a = int(math.floor(current_val))
        p = a*p_0 + p_minus1
        n = a*n_0 + n_minus1
```

```
        ratio = p/n


        if lower_b <= ratio <= upper_b:
            return p,n

        remainder = current_val - a
        if abs(remainder) < 1e-15:
            return p,n

        current_val = 1.0 / remainder
        p_minus1 = p_0
        n_minus1 = n_0
        p_0 = p
        n_0 = n


def minimal_pair(lower_b, upper_b):
    ### Smallest n that has some p with lower_b <= p/n <= upper_b
    n = 1
    while True:
        p = max(1, math.ceil(lower_b * n))
        if p <= upper_b * n:
            return p, n
        n += 1


try:
    lower_range = float(eval(input("Enter the lower bound: "))) # e.g., 0
    upper_range = float(eval(input("Enter the upper bound: "))) # e.g., math.pi
    step = float(eval(input("Enter the step size: ")))# e.g., 0.5
    ep = float(eval(input("Enter the desired accuracy range: ")))
    q = lower_range
    while q<upper_range:
        try:
            G = (1/4) * math.exp(2*(q))
            lower = (1/4) * math.exp(2*(q-ep))
            upper = (1/4) * math.exp(2*(q+ep))

            p, n = efficiency_approx(G, lower, upper)
            mp, mn = minimal_pair(lower, upper)


            if n<=1 or ep>=1:
                efficiency=0
            else:
                efficiency = 0.5 * abs((1.0 / p) - (1.0 / n))
```

```
            print(f"{q:<10.4f} | {p:<8} : {n:<8} | {mp:<8} : {mn:<8} | {efficiency:<12.6f}")

        except ZeroDivisionError:
            print("Calculation error")
        q+=step


except ValueError:
    print("Please enter a valid number")
```

Numerical caveats

The bounds L and U and the check $L \leq p/n \leq U$ use regular double-precision numbers, so a ratio that lands extremely close to an endpoint could be counted wrongly. The stopping test $|r| < 10^{-15}$ also limits how many continued fraction terms can be trusted. In addition, computing e^{2S} overflows near $S \approx 355$ and underflows below $S \approx -372$. Every value in the tables was double-checked with 60-digit decimal arithmetic.

Full Numerical Results

Full results near $\frac{1}{2}ln2$ *(*$\varepsilon = 10^{-5}$*).*

Target S	Alg. 1 $(p: n)$	Minimal Pair $(p: n)$	C
0.3460	435 : 871	429 : 859	0.000575
0.3461	527 : 1055	517 : 1035	0.000475
0.3462	668 : 1337	652 : 1305	0.000375
0.3463	913 : 1827	882 : 1765	0.000274
0.3464	1439 : 2879	1362 : 2725	0.000174
0.3465	3396 : 6793	2991 : 5983	0.000074
0.3466	9466 : 18931	6867 : 13733	0.000026
0.3467	1977 : 3953	1833 : 3665	0.000126
0.3468	1104 : 2207	1058 : 2115	0.000226
0.3469	766 : 1531	744 : 1487	0.000326
0.3470	586 : 1171	574 : 1147	0.000426

Full results across $0.60 \leq S \leq 0.80$ *(*$\varepsilon = 10^{-4}$*).*

Target S	Alg. 1 $(p: n)$	Minimal Pair $(p: n)$	C
0.6000	44 : 53	44 : 53	0.001930
0.6100	94 : 111	83 : 98	0.000815

Target S	Alg. 1 $(p:n)$	Minimal Pair $(p:n)$	C
0.6200	146 : 169	108 : 125	0.000466
0.6300	52 : 59	52 : 59	0.001141
0.6400	107 : 119	89 : 99	0.000471
0.6500	111 : 121	100 : 109	0.000372
0.6600	73 : 78	73 : 78	0.000439
0.6700	190 : 199	127 : 133	0.000119
0.6800	75 : 77	75 : 77	0.000173
0.6900	158 : 159	154 : 155	0.000020
0.7000	73 : 72	73 : 72	0.000095
0.7100	30 : 29	30 : 29	0.000575
0.7200	153 : 145	115 : 109	0.000180
0.7300	183 : 170	141 : 131	0.000209
0.7400	56 : 51	56 : 51	0.000875
0.7500	93 : 83	93 : 83	0.000648
0.7600	8 : 7	8 : 7	0.008929
0.7700	372 : 319	260 : 223	0.000223
0.7800	69 : 58	69 : 58	0.001374
0.7900	142 : 117	125 : 103	0.000752
0.8000	26 : 21	26 : 21	0.004579

Full results across $0.60 \leq S \leq 0.80$ *(*$\varepsilon = 10^{-6}$*).*

Target S	Alg. 1 $(p:n)$	Minimal Pair $(p:n)$	C
0.6000	293 : 353	293 : 353	0.000290
0.6100	304 : 359	304 : 359	0.000252
0.6200	2501 : 2895	1479 : 1712	0.000027
0.6300	52 : 59	52 : 59	0.001141
0.6400	107 : 119	107 : 119	0.000471
0.6500	233 : 254	233 : 254	0.000177
0.6600	569 : 608	569 : 608	0.000056
0.6700	401 : 420	401 : 420	0.000056
0.6800	563 : 578	563 : 578	0.000023
0.6900	475 : 478	475 : 478	0.000007
0.7000	955 : 942	808 : 797	0.000007
0.7100	875 : 846	875 : 846	0.000020
0.7200	153 : 145	153 : 145	0.000180

Target S	Alg. 1 (p: n)	Minimal Pair (p: n)	C
0.7300	957 : 889	957 : 889	0.000040
0.7400	436 : 397	436 : 397	0.000113
0.7500	1591 : 1420	1163 : 1038	0.000038
0.7600	815 : 713	815 : 713	0.000088
0.7700	751 : 644	751 : 644	0.000111
0.7800	809 : 680	809 : 680	0.000117
0.7900	159 : 131	159 : 131	0.000672
0.8000	1081 : 873	1081 : 873	0.000110

Full results for negative targets ($\varepsilon = 10^{-3}$).

Target S	Alg. 1 (p: n)	Minimal Pair (p: n)	C	Limiting Sum	Error
-1.5000	2 : 161	2 : 161	0.246894	-1.5010	0.0010
-1.4500	3 : 218	3 : 218	0.164373	-1.4498	0.0002
-1.4000	4 : 263	3 : 197	0.123099	-1.3998	0.0002
-1.3500	2 : 119	2 : 119	0.245798	-1.3498	0.0002
-1.3000	6 : 323	4 : 215	0.081785	-1.2998	0.0002
-1.2500	3 : 146	3 : 146	0.163242	-1.2493	0.0007
-1.2000	10 : 441	6 : 265	0.048866	-1.2001	0.0001
-1.1500	9 : 359	6 : 239	0.054163	-1.1499	0.0001
-1.1000	9 : 325	6 : 217	0.054017	-1.1002	0.0002
-1.0500	3 : 98	3 : 98	0.161565	-1.0500	0.0000
-1.0000	2 : 59	2 : 59	0.241525	-0.9990	0.0010
-0.9500	4 : 107	4 : 107	0.120327	-0.9501	0.0001
-0.9000	5 : 121	5 : 121	0.095868	-0.9000	0.0000
-0.8500	9 : 197	7 : 153	0.053017	-0.8498	0.0002
-0.8000	5 : 99	5 : 99	0.094949	-0.7997	0.0003
-0.7500	13 : 233	10 : 179	0.036316	-0.7499	0.0001
-0.7000	4 : 65	4 : 65	0.117308	-0.7009	0.0009
-0.6500	3 : 44	3 : 44	0.155303	-0.6496	0.0004
-0.6000	7 : 93	7 : 93	0.066052	-0.6002	0.0002
-0.5500	1 : 12	1 : 12	0.458333	-0.5493	0.0007