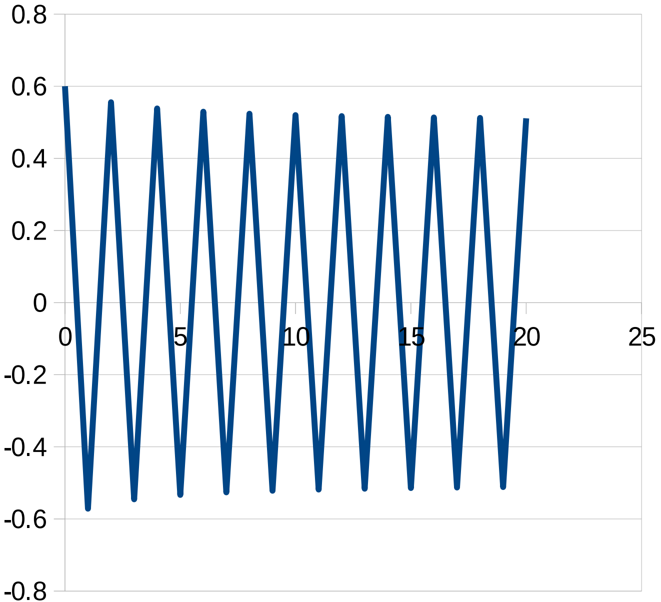
0	0.6
1	-0.57142857
2	0.555555556
3	-0.54545455
4	0.538461538
5	-0.53333333
6	0.529411765
7	-0.52631579
8	0.523809524
9	-0.52173913
10	0.52
11	-0.51851852
12	0.517241379
13	-0.51612903
14	0.515151515
15	-0.51428571
16	0.513513514
17	-0.51282051
18	0.512195122
19	-0.51162791
20	0.511111111



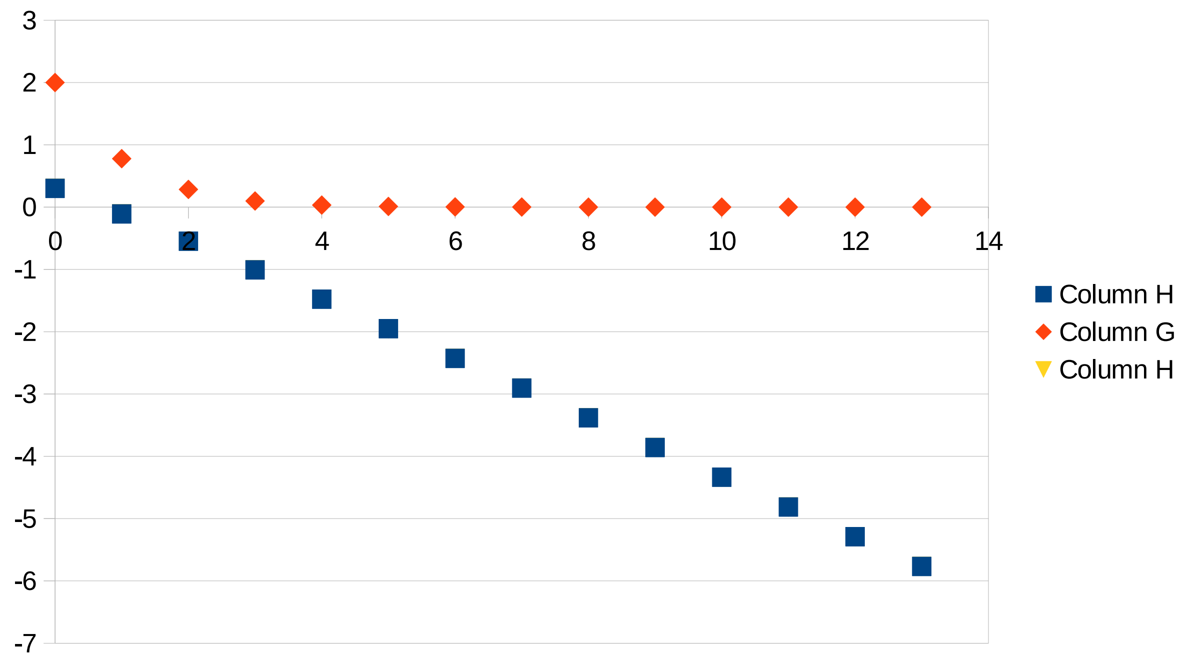
Plot of the sequence a(n)=(-1)^n * (n+3)/(2n+5)

n	xn	error	log10(error)
0	2	2	0.301029996
1	1.5	0.25	-0.60205999
2	1.416666667	0.006944444	-2.15836249
3	1.414215686	6.0073E-06	-5.22132033
4	1.414213562	4.51061E-12	-11.3457643
5	1.414213562	0	Err:502
6	1.414213562	0	Err:502
7	1.414213562	0	Err:502
8	1.414213562	0	Err:502
9	1.414213562	0	Err:502
10	1.414213562	0	Err:502
11	1.414213562	0	Err:502
12	1.414213562	0	Err:502
13	1.414213562	0	Err:502

n	xn	error	log10(error)
0	2	2	0.301029996
1	1.666666667	0.777777778	-0.10914447
2	1.511111111	0.28345679	-0.54751314
3	1.448583878	0.098395252	-1.00702586
4	1.425942167	0.033311064	-1.47741149
5	1.418155254	0.011164324	-1.95216756
6	1.415531111	0.003728327	-2.42848595
7	1.414653154	0.001243547	-2.90533792
8	1.414360139	0.000414601	-3.38236923
9	1.414262426	0.00013821	-3.85946048
10	1.414229851	4.60711E-05	-4.33657173
11	1.414218992	1.53571E-05	-4.81368965
12	1.414215372	5.11906E-06	-5.2908098
13	1.414214166	1.70635E-06	-5.76793068

Columns A-D: Newton-Raphson method of
Solving x^2-2=0.
Starting with an initial guess x0=2 for
The square root of 2, after four iteration
The result is correct up to 10 decimal digits.
Observe that the number of the correct digits
Doubles after each iteration.

Columns E-H: If we are less smart, so a less
Optimal, but a still very reasonable
algorithm is used, we get a quite good
approximation, the error decays like a
geometric sequence.



n	xn	lg(xn)	xn-100	lg(xn-100)
0	222	2.346352974	122	2.086359831
1	246.4	2.391640703	146.4	2.165541077
2	275.68	2.44040526	175.68	2.244722323
3	310.816	2.492503367	210.816	2.323903569
4	352.9792	2.547749114	252.9792	2.403084815
5	403.57504	2.605924298	303.57504	2.482266061
6	464.290048	2.666789375	364.290048	2.561447307
7	537.1480576	2.73009401	437.1480576	2.640628553
8	624.5776691	2.795586453	524.5776691	2.719809799
9	729.4932029	2.86302125	629.4932029	2.798991045
10	855.3918435	2.932165105	755.3918435	2.878172291
11	1006.470212	3.002800926	906.4702122	2.957353537
12	1187.764255	3.074730251	1087.764255	3.036534783
13	1405.317106	3.147774333	1305.317106	3.115716029
14	1666.380527	3.221774182	1566.380527	3.194897275
15	1979.656632	3.296589869	1879.656632	3.274078521
16	2355.587959	3.372099326	2255.587959	3.353259767
17	2806.70555	3.448196853	2706.70555	3.432441013
18	3348.04666	3.524791502	3248.04666	3.51162226
19	3997.655992	3.601805419	3897.655992	3.590803506
20	4777.187191	3.679172259	4677.187191	3.669984752

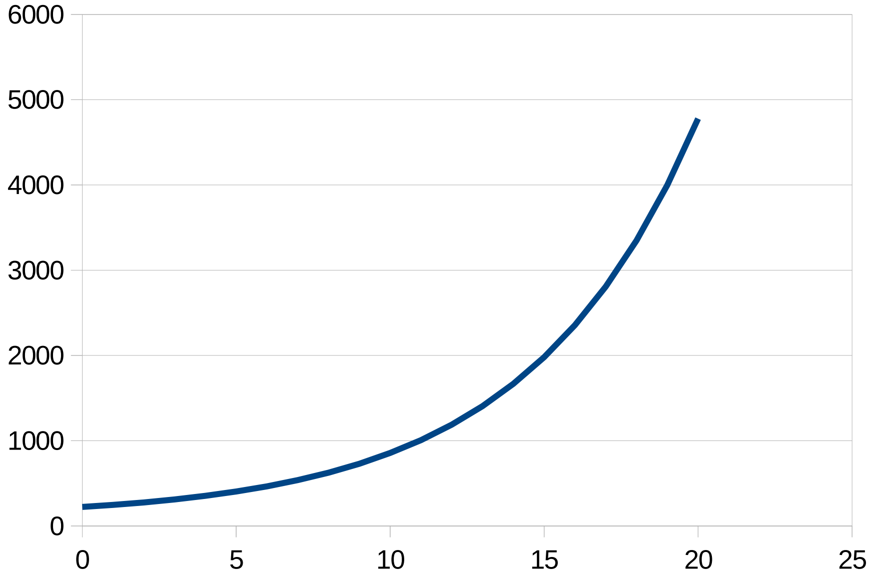
0.067733891

0.067733891	2.283707128
0.001017424	0.011894151
0.995731375	0.028232382
4432.081574	19
3.532667613	0.01514428

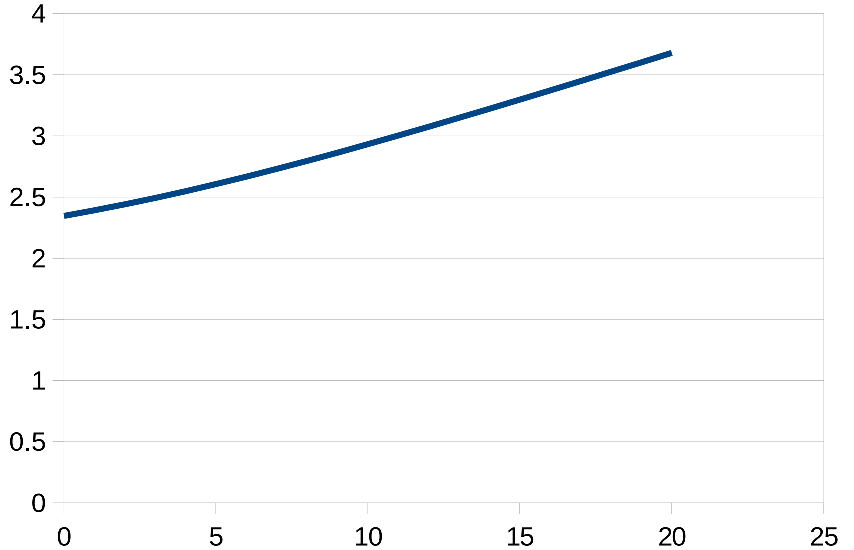
Best fit for lg(xn)

0.079181246	2.086359831
2.5108E-17	2.93523E-16
1	6.96718E-16
9.94539E+30	19
4.827645689	9.22289E-30

Best fit for lg(xn-100)



Plot of n ↔ xn



Plot of n ↔ lg(xn)

n	x	y	error
0	0	1	0
1	0.05	1.05	0.001271096
2	0.1	1.1025	0.002670918
3	0.15	1.157625	0.004209243
4	0.2	1.21550625	0.005896508
5	0.25	1.276281563	0.007743854
6	0.3	1.340095641	0.009763167
7	0.35	1.407100423	0.011967126
8	0.4	1.477455444	0.014369254
9	0.45	1.551328216	0.01698397
10	0.5	1.628894627	0.019826644
11	0.55	1.710339358	0.02291366
12	0.6	1.795856326	0.026262474
13	0.65	1.885649142	0.029891687
14	0.7	1.979931599	0.033821108
15	0.75	2.078928179	0.038071837
16	0.8	2.182874588	0.04266634
17	0.85	2.292018318	0.047628534
18	0.9	2.406619234	0.052983877
19	0.95	2.526950195	0.058759464
20	1	2.653297705	0.064984123

Euler method for y'(x)=y(x), y(0)=1.
The step size is Delta x=0.05, in 20 steps c23 gives the approximate value
2.65329 for y(1)=exp(1)=e=2.71828…
The error is about 0.06498.