

Are Subsequences of Decimal Digits of π Random?

Suman Kumar Sourabh

University Department of Statistics & Computer Applications,
T.M. Bhagalpur University, Bhagalpur – 812007, India

Soubhik Chakraborty

Department of Applied Mathematics, BIT Mesra, Ranchi-835215, India

soubhik@yahoo.co.in

Basant Kumar Das

University Department of Statistics & Computer Applications,
T.M. Bhagalpur University, Bhagalpur – 812007, India

ABSTRACT: A lot has been done on the randomness of the decimal expansion of π with extensive tests of randomness that are used to distinguish good from not-so-good random number generators when applied to the decimal digits of π . π seems to pass these tests as well as some of the best random number generator (RNG) and could well serve as an RNG provided that the digits of π could be easily and quickly produced in the computer [Mar06]. We make an interesting study in the same context in which random substring of arbitrary length are extracted from arbitrary positions a large number of times and each sample is tested for randomness. Our results confirm the randomness of π and a recent claim that “ π is less random than we thought” [TF05] stands refuted. George Marsaglia [Mar06] has also independently refuted the claim but in Marsaglia’s work, the randomness is established on the whole for the first 960 million digits of π . Our study confirms the randomness for arbitrary subsequences also. Finally, the investigation of some functions of π -rather than π itself-is proposed.

KEYWORDS: π , Substring, Run Test for Randomness

Introduction

A lot has been done on the randomness of the decimal expansion of π with extensive tests of randomness that are used to distinguish good from not-so-

good random number generators when applied to the decimal digits of Pi . Pi seems to pass these tests as well as some of the best random number generator (RNG) and could well serve as an RNG provided that the digits of Pi could be easily and quickly produced in the computer [Mar06]. We make an interesting study in the same context in which random substring of arbitrary length are extracted from arbitrary positions a large number of times and each sample is tested for randomness. Our results confirm the randomness of Pi and a recent claim that “ Pi is less random than we thought” [TF05] stands refuted. George Marsaglia [Mar06] has also independently refuted the claim but in Marsaglia’s work, the randomness is established on the whole for the first 960 million digits of pi. Our study confirms the randomness for arbitrary subsequences also. Finally, the investigation of some functions of pi-rather than pi itself is proposed.

1. Test for randomness of decimal digits of Pi for fixed length n which starts from an arbitrary position s

Algorithm:

Step-1: Create a text file say *Pidec.txt* of decimal expansion of Pi.

Step-2: Set MAX = 60,000

Step-3: INPUT n

Step-4: $z = U(0,1)$

$$s = \text{INT}[z * (\text{MAX} - n)] + 1$$

when $n = \text{MAX}$, $s = 1$ otherwise s is a random number between 1 and $\text{MAX} - n$

Step-5: Create an array *arrs* (convert characters to numbers) of n elements starting from s^{th} position of the text file *pidec.txt*.

Step-6: Create an array *arrws* by copying all the elements of array *arrs*.

Step-7: Find the median by the following steps:-

(a) Sort the array *arrs*

(b) If n is odd, median= $(n+1/2)^{\text{th}}$ term.

(c) If n is even, median=mean of $(n/2)^{\text{th}}$ and $(n/2+1)^{\text{th}}$ term

Step-8: From the first digit to the last digit of the unsorted array *arrws*,

(a) Write ‘A’ if the digit is less than the median, or

(b) Write ‘B’ if more, or

(c) if equal, draw a uniform number p between (0 to 1) , if $p > 0.5$ then write ‘B’ otherwise ‘A’

Step-9: Store all characters obtained from step-8 in a string variable *txtstr*.

Step-10: Initialize a new character variable *temp* with the first character of *txtstr* and set $c = 1$

Step-11: From second character onwards, examine with *temp*, if unequal $c = c + 1$ and *temp*=corresponding character. If equal examine the next character. Final value of *c* gives the number of runs in *txtstr*.

Step-12: *c* is asymptotically normal with mean $E(c) = (n+2)/2$ and variance $Var(c) = n(n-2)/4(n-1)$

$$\text{Calculate } U = \{c - E(c)\} / \{Var(c)\}^{1/2}$$

where *U* is $N(0,1)$ for large *n*

Step-13: If $|U| < 1.96$, subsequence may be taken as random at 5% level of significance.

Step-14: Repeat steps 4 to 13 a large number of times for every *n* chosen.

Table 1 gives the run time results.

2. Runtime results

Table 1. Results (the C code is not shown)

n T	50			500			5000			50000		
	s	Random	Test	S	Random	Test	s	Random	Test	s	Random	Test
1	23358	0.017496	Pass	30486	0.013716	Pass	4655	0.013319	Pass	378	0.867603	Pass
2	25780	0.005832	Pass	739	0.009144	Pass	8465	0.259723	Pass	3723	1.126989	Pass
3	26471	0.005832	Pass	1770	0.082297	Pass	10370	0.046617	Pass	6606	1.583152	Pass
4	26818	0.029161	Pass	2456	0.068581	Pass	12909	0.532765	Pass	9086	1.905149	Pass
5	27164	0.005832	Pass	3143	0.013716	Pass	14180	0.606021	Pass	758	0.983880	Pass
6	27856	0.040825	Pass	3486	0.077725	Pass	14814	0.606021	Pass	2488	2.155591	Fail
7	28202	0.034993	Pass	3486	0.077725	Pass	15449	0.532765	Pass	4334	0.894436	Pass
8	28548	0.011664	Pass	3830	0.132590	Pass	16083	0.659297	Pass	6467	0.778159	Pass
9	28893	0.005832	Pass	4173	0.118874	Pass	16402	0.699255	Pass	8024	1.601041	Pass
10	29240	0.011664	Pass	4517	0.004572	Pass	17036	0.812467	Pass	9581	1.216433	Pass
11	998	0.005832	Pass	4861	0.109730	Pass	17671	0.739212	Pass	1023	1.636818	Pass
12	1689	0.017496	Pass	5204	0.036577	Pass	18306	0.759191	Pass	3214	1.779928	Pass
13	2036	0.029161	Pass	5547	0.032005	Pass	18624	0.732552	Pass	4656	0.957047	Pass
14	2382	0.005832	Pass	5547	0.032005	Pass	19258	0.819127	Pass	6328	1.243266	Pass
15	2727	0.000000	Pass	5891	0.068581	Pass	19893	0.652638	Pass	7828	1.493708	Pass
16	3420	0.011664	Pass	6234	0.027432	Pass	20528	0.719233	Pass	8116	1.538430	Pass
17	3765	0.023328	Pass	6577	0.059437	Pass	20846	0.792488	Pass	1230	1.770984	Pass
18	4111	0.017496	Pass	6920	0.027432	Pass	21481	0.519446	Pass	1576	0.930214	Pass
19	4458	0.017496	Pass	7264	0.036577	Pass	22115	0.399574	Pass	2845	1.627874	Pass

n T	50			500			5000			50000		
	Value	Value	Pass	Value	Value	Pass	Value	Value	Pass	Value	Value	Pass
20	4803	0.017496	Pass	7264	0.036577	Pass	22432	0.033298	Pass	4056	0.930214	Pass
21	31793	0.034993	Pass	4646	0.032005	Pass	23068	0.133191	Pass	6362	1.019657	Pass
22	782	0.023328	Pass	6364	0.022860	Pass	10173	0.019979	Pass	6708	0.894436	Pass
23	1129	0.046657	Pass	7737	0.004572	Pass	12395	0.006660	Pass	7862	0.921269	Pass
24	1474	0.023328	Pass	8081	0.022860	Pass	13029	0.492808	Pass	9130	0.903381	Pass
25	1820	0.000000	Pass	8768	0.045721	Pass	13347	0.639318	Pass	2129	1.609985	Pass
26	2167	0.034993	Pass	9111	0.013716	Pass	13982	0.426212	Pass	2475	1.762039	Pass
27	2512	0.029161	Pass	9455	0.013716	Pass	14617	0.692595	Pass	3744	1.824650	Pass
28	2858	0.046657	Pass	9798	0.100586	Pass	15251	0.566063	Pass	4955	0.044722	Pass
29	3551	0.011664	Pass	10141	0.100586	Pass	15886	0.539425	Pass	6281	1.064379	Pass
30	3896	0.058321	Pass	10485	0.050293	Pass	16522	0.692595	Pass	9280	1.413209	Pass
31	5280	0.000000	Pass	10829	0.054865	Pass	16839	0.645978	Pass	837	1.466875	Pass
32	5280	0.000000	Pass	11172	0.013716	Pass	17474	0.725893	Pass	3374	1.860427	Pass
33	5627	0.005832	Pass	11515	0.036577	Pass	18108	0.645978	Pass	6719	1.583152	Pass
34	5973	0.029161	Pass	11859	0.013716	Pass	18426	0.845765	Pass	7468	1.949871	Pass
35	6665	0.023328	Pass	12545	0.077725	Pass	19061	0.699255	Pass	8795	1.413209	Pass
36	7011	0.005832	Pass	12889	0.032005	Pass	19696	0.839105	Pass	121	0.527717	Pass
37	7356	0.005832	Pass	13232	0.128018	Pass	20013	0.699255	Pass	2601	1.779928	Pass
38	7703	0.005832	Pass	13575	0.013716	Pass	20648	0.792488	Pass	2947	0.957047	Pass
39	8049	0.005832	Pass	13920	0.018288	Pass	21283	0.665957	Pass	4042	1.770984	Pass
40	8394	0.017496	Pass	14263	0.064009	Pass	21918	0.512787	Pass	6580	1.842538	Pass
41	2989	0.000000	Pass	27999	0.013716	Pass	22235	0.392914	Pass	6926	1.636818	Pass
42	5065	0.011664	Pass	29373	0.004572	Pass	22870	0.046617	Pass	8079	0.804992	Pass
43	6449	0.023328	Pass	29717	0.027432	Pass	15054	0.725893	Pass	9175	1.574208	Pass
44	6796	0.017496	Pass	30060	0.018288	Pass	16959	0.559404	Pass	213	1.511597	Pass
45	7142	0.011664	Pass	30403	0.009144	Pass	17594	0.712574	Pass	2116	1.073323	Pass
46	7487	0.005832	Pass	30747	0.100586	Pass	18228	0.759191	Pass	4711	1.806761	Pass
47	7834	0.017496	Pass	31433	0.137162	Pass	18864	0.759191	Pass	5057	1.055435	Pass
48	8525	0.011664	Pass	31776	0.045721	Pass	19498	0.852425	Pass	6037	1.440042	Pass
49	8871	0.005832	Pass	32121	0.000000	Pass	19816	0.625999	Pass	7190	0.545606	Pass
50	9218	0.000000	Pass	32464	0.009144	Pass	20450	0.732552	Pass	8344	1.744151	Pass
51	9564	0.023328	Pass	312	0.064009	Pass	21086	0.719233	Pass	9382	1.135934	Pass
52	9909	0.023328	Pass	999	0.013716	Pass	21721	0.539425	Pass	304	0.948102	Pass
53	10256	0.005832	Pass	1342	0.027432	Pass	22038	0.346297	Pass	1573	1.279044	Pass
54	10602	0.005832	Pass	1686	0.013716	Pass	22673	0.159830	Pass	2784	1.601041	Pass
55	10947	0.011664	Pass	2029	0.004572	Pass	23308	0.206447	Pass	4918	1.359543	Pass
56	11294	0.029161	Pass	2373	0.077725	Pass	23625	0.133191	Pass	5264	1.475820	Pass
57	11640	0.005832	Pass	2717	0.009144	Pass	24260	0.352957	Pass	6821	1.243266	Pass
58	11986	0.023328	Pass	3060	0.022860	Pass	24895	0.099894	Pass	7743	1.896205	Pass
59	12332	0.017496	Pass	3403	0.013716	Pass	25530	0.146510	Pass	3106	1.493708	Pass
60	12678	0.052489	Pass	3746	0.187455	Pass	25848	0.099894	Pass	4260	1.923038	Pass

$\frac{n}{T}$	50			500			5000			50000		
61	2082	0.023328	Pass	20918	0.086869	Pass	26482	0.173149	Pass	5413	1.082268	Pass
62	3812	0.011664	Pass	22291	0.109730	Pass	27117	0.519446	Pass	6624	1.538430	Pass
63	4158	0.058321	Pass	22634	0.059437	Pass	27435	0.472829	Pass	7777	0.885492	Pass
64	4505	0.029161	Pass	22978	0.050293	Pass	28070	0.313000	Pass	8758	1.583152	Pass
65	4850	0.017496	Pass	23322	0.054865	Pass	29657	0.452851	Pass	9853	1.663651	Pass
66	6580	0.000000	Pass	23665	0.013716	Pass	254	0.466170	Pass	1410	1.100156	Pass
67	6927	0.017496	Pass	24352	0.096014	Pass	16444	0.685935	Pass	2448	1.681540	Pass
68	7272	0.005832	Pass	24695	0.045721	Pass	18349	0.699255	Pass	3602	1.305877	Pass
69	7618	0.000000	Pass	25038	0.027432	Pass	19301	0.792488	Pass	4640	0.420385	Pass
70	7965	0.005832	Pass	25382	0.050293	Pass	20253	0.765850	Pass	5851	0.670827	Pass
71	8311	0.017496	Pass	25725	0.041149	Pass	20889	0.812467	Pass	7004	0.921269	Pass
72	8656	0.005832	Pass	26068	0.018288	Pass	21523	0.552744	Pass	8100	1.806761	Pass
73	9002	0.034993	Pass	26068	0.018288	Pass	22158	0.472829	Pass	9311	1.279044	Pass
74	9349	0.000000	Pass	26413	0.050293	Pass	22793	0.159830	Pass	291	1.735206	Pass
75	9694	0.023328	Pass	26756	0.054865	Pass	23111	0.039957	Pass	5481	0.697660	Pass
76	10040	0.017496	Pass	27099	0.045721	Pass	23745	0.046617	Pass	6288	1.171711	Pass
77	10387	0.011664	Pass	27442	0.018288	Pass	24380	0.199787	Pass	7557	1.905149	Pass
78	11078	0.017496	Pass	27786	0.009144	Pass	25015	0.099894	Pass	8595	1.162767	Pass
79	11425	0.011664	Pass	28129	0.009144	Pass	25650	0.073255	Pass	9575	1.162767	Pass
80	24920	0.005832	Pass	28472	0.009144	Pass	26285	0.279702	Pass	556	1.735206	Pass
81	26649	0.023328	Pass	6624	0.041149	Pass	26920	0.113213	Pass	1536	1.806761	Pass
82	26995	0.005832	Pass	8340	0.036577	Pass	27237	0.106553	Pass	2401	1.592096	Pass
83	27342	0.040825	Pass	8685	0.059437	Pass	27872	0.519446	Pass	4419	0.715549	Pass
84	27687	0.034993	Pass	9715	0.100586	Pass	28507	0.472829	Pass	6207	1.118045	Pass
85	28033	0.000000	Pass	10401	0.022860	Pass	29142	0.645978	Pass	6957	1.860427	Pass
86	28380	0.017496	Pass	10745	0.032005	Pass	29459	0.566063	Pass	7995	1.842538	Pass
87	28726	0.011664	Pass	11088	0.022860	Pass	57	0.226425	Pass	9033	1.788872	Pass
88	29071	0.023328	Pass	11431	0.000000	Pass	692	0.566063	Pass	1628	1.323766	Pass
89	29418	0.011664	Pass	11775	0.032005	Pass	1644	0.645978	Pass	1974	1.583152	Pass
90	29764	0.000000	Pass	12119	0.041149	Pass	1961	0.699255	Pass	3127	1.556319	Pass
91	30109	0.011664	Pass	12462	0.064009	Pass	13707	0.632659	Pass	4280	0.983880	Pass
92	30455	0.017496	Pass	12805	0.041149	Pass	15612	0.599361	Pass	5145	0.957047	Pass
93	30802	0.040825	Pass	13149	0.118874	Pass	16247	0.599361	Pass	6414	1.332710	Pass
94	31148	0.000000	Pass	13835	0.045721	Pass	17199	0.665957	Pass	7452	1.833594	Pass
95	31493	0.000000	Pass	14179	0.041149	Pass	17834	0.925680	Pass	8317	0.948102	Pass
96	31493	0.000000	Pass	14522	0.041149	Pass	18151	0.699255	Pass	9355	1.565263	Pass
97	31840	0.005832	Pass	14866	0.041149	Pass	18787	0.759191	Pass	278	1.466875	Pass
98	32186	0.005832	Pass	15210	0.000000	Pass	19421	0.879063	Pass	1489	1.440042	Pass
99	32531	0.034993	Pass	15553	0.018288	Pass	20056	0.599361	Pass	3449	1.422153	Pass
100	13285	0.046657	Pass	15896	0.013716	Pass	20691	0.652638	Pass	5871	1.323766	Pass

3. Test for randomness of decimal digits of Pi for an arbitrary length n which starts from a fixed position s

Algorithm:

Step-1: Create a text file say *Pidec.txt* of decimal expansion of Pi.

Step-2: Set $MAX = 60,000$

Step-3: INPUT s

Step-4: $z = U(0,1)$

$$n = \text{INT}[z * (MAX - (a + s) + 1)] + a$$

when $s = MAX$, $n = 1$ otherwise n is a random number between 1 and $MAX - n$

Step-5: Create an array *arrs* (convert characters to numbers) of n elements starting from s^{th} position of the text file *pidec.txt*.

Step-6: Create an array *arrws* by copying all the elements of array *arrs*.

Step-7: Find the median by the following steps:-

(d) Sort the array *arrs*

(e) If n is odd, median= $(n+1/2)^{\text{th}}$ term.

(f) If n is even, median=mean of $(n/2)^{\text{th}}$ and $(n/2+1)^{\text{th}}$ term

Step-8: From the first digit to the last digit of the unsorted array *arrws*,

(d) Write 'A' if the digit is less than the median, or

(e) Write 'B' if more, or

(f) if equal, draw a uniform number p between (0 to 1), if $p > 0.5$ then write 'B' otherwise 'A'

Step-9: Store all characters obtained from step-8 in a string variable *txtstr*.

Step-10: Initialize a new character variable *temp* with the first character of *txtstr* and set $c = 1$

Step-11: From second character onwards, examine with *temp*, if unequal $c = c + 1$ and *temp*=corresponding character. If equal examine the next character. Final value of c gives the number of runs in *txtstr*.

Step-12: c is asymptotically normal with mean $(n+2)/2$
and variance $n(n-2) / 4(n-1)$

$$\text{Calculate } |U| = \left| \frac{c - \left(\frac{n + 2}{2} \right)}{\sqrt{n / 4 \left(\frac{n - 2}{n - 1} \right)}} \right|$$

Where U is $N(0,1)$ for large n

Step-13: If $|U| < 1.96$, subsequence may be taken as random at 5% level of significance.

Step-14: Repeat steps 4 to 13 a large number of times for every s chosen.

Table 2 gives the run time results.

4. Runtime

Table 2. Results (the C code is omitted)

s T	50			500			5000			50000		
	n	Random	Test	n	Random	Test	n	Random	Test	n	Random	Test
1	6206	0.252320	Pass	27700	4.688789	Fail	7899	0.067080	Pass	2857	0.118502	Pass
2	981	0.062943	Pass	8265	0.776746	Pass	13292	0.016972	Pass	3835	0.531438	Pass
3	24499	2.256038	Fail	10483	0.775723	Pass	28839	3.018472	Fail	4411	0.421234	Pass
4	13908	0.867626	Pass	24898	1.468514	Pass	18807	2.174931	Fail	4928	0.029869	Pass
5	26704	2.380553	Fail	15418	1.605910	Pass	29911	2.670364	Fail	5216	0.094016	Pass
6	17497	3.308717	Fail	23999	1.982038	Fail	15436	3.721204	Fail	174	0.009017	Pass
7	1719	1.796005	Pass	3190	0.181333	Pass	20196	1.914837	Pass	576	0.098176	Pass
8	9674	0.624230	Pass	10398	3.298000	Fail	26541	3.225200	Fail	979	0.167633	Pass
9	24892	2.338208	Fail	15890	1.521816	Pass	13970	1.135549	Pass	1555	1.212240	Pass
10	13263	0.840713	Pass	23099	2.266076	Fail	17460	2.019047	Fail	930	0.192351	Pass
11	27097	3.170947	Fail	918	0.033438	Pass	22536	1.814956	Pass	1390	0.556656	Pass
12	10628	0.679893	Pass	6410	1.051571	Pass	129	0.013685	Pass	1584	1.348756	Pass
13	16852	0.405942	Pass	14305	1.053211	Pass	2350	0.810038	Pass	1844	1.743380	Pass
14	31724	1.942620	Pass	20827	3.641050	Fail	5841	0.206750	Pass	3513	0.838872	Pass
15	23209	1.844900	Pass	30094	2.391061	Fail	8379	0.039610	Pass	3686	0.477421	Pass
16	3280	0.024276	Pass	14090	0.987288	Pass	10600	0.026124	Pass	3801	4.199133	Fail
17	13310	0.733236	Pass	20956	1.638568	Pass	13455	0.444412	Pass	3858	0.395886	Pass
18	24377	4.319048	Fail	1864	1.485429	Pass	17580	3.444190	Fail	3974	1.669563	Pass
19	3757	0.015685	Pass	8042	0.413325	Pass	23925	1.452638	Pass	4089	0.353352	Pass
20	11366	1.009692	Pass	14907	2.608752	Fail	13258	0.041071	Pass	4146	0.022556	Pass
21	29350	1.536118	Pass	21429	1.961161	Fail	16748	1.147333	Pass	4261	0.002731	Pass
22	13226	0.957432	Pass	31727	3.993464	Fail	22142	3.824559	Fail	4376	0.004936	Pass

$\frac{s}{T}$	50			500			5000			50000		
23	21181	1.243167	Pass	22932	2.483099	Fail	29757	1.806249	Pass	4491	0.020676	Pass
24	30518	15.587662	Fail	4526	0.760922	Pass	19724	3.066663	Fail	4607	0.134212	Pass
25	14395	0.759116	Pass	11047	0.739065	Pass	27338	2.215566	Fail	4664	0.074600	Pass
26	22003	6.256405	Fail	17569	1.781693	Pass	14767	1.877244	Pass	4779	0.035215	Pass
27	32379	1.951606	Pass	26838	1.733725	Pass	19844	6.433619	Fail	4894	0.031362	Pass
28	822	0.148671	Pass	8089	0.558965	Pass	27458	2.631436	Fail	5009	0.075406	Pass
29	8430	0.380431	Pass	15984	1.370684	Pass	12349	0.006226	Pass	5125	0.027386	Pass
30	17077	0.100730	Pass	22506	2.949886	Fail	16156	1.802004	Pass	5240	0.125541	Pass
31	25032	11.157707	Fail	29500	3.457451	Fail	21233	1.434934	Pass	5355	0.514019	Pass
32	4412	0.270093	Pass	17616	1.370294	Pass	1681	1.514118	Pass	5470	0.118509	Pass
33	11329	0.736601	Pass	22765	1.664535	Pass	3585	0.279227	Pass	139	0.038474	Pass
34	17208	0.905965	Pass	21178	2.224690	Fail	5489	1.158881	Pass	255	0.041112	Pass
35	24470	2.101616	Fail	28730	2.372832	Fail	8027	0.111160	Pass	5457	0.141401	Pass
36	3505	0.002781	Pass	30576	1.820397	Pass	10249	0.129575	Pass	357	0.089925	Pass
37	10076	1.098106	Pass	4406	0.494811	Pass	12786	0.018591	Pass	530	0.046356	Pass
38	15956	1.221292	Pass	12643	1.034148	Pass	16911	2.740389	Fail	702	0.113336	Pass
39	22527	1.651246	Pass	20882	1.911239	Pass	532	0.146486	Pass	818	0.069955	Pass
40	868	0.085257	Pass	416	0.010164	Pass	4022	0.058087	Pass	2854	0.052144	Pass
41	7440	0.521092	Pass	2819	0.419739	Pass	5926	0.193411	Pass	2969	0.018390	Pass
42	13666	0.816538	Pass	6251	0.476800	Pass	7830	0.087025	Pass	3026	0.407752	Pass
43	31303	3.770446	Fail	9685	2.720574	Fail	10051	0.118669	Pass	3084	0.360079	Pass
44	16217	0.035935	Pass	14146	1.110914	Pass	13223	0.052391	Pass	3141	0.053161	Pass
45	24517	2.562739	Fail	31652	3.051246	Fail	23377	1.581573	Pass	3199	0.071224	Pass
46	9431	0.891151	Pass	20798	1.351809	Pass	2556	0.068215	Pass	3256	0.079588	Pass
47	17039	0.972400	Pass	31782	2.255697	Fail	4777	0.028147	Pass	3371	0.403334	Pass
48	23956	1.486950	Pass	18182	3.147619	Fail	6681	0.539467	Pass	3429	0.393835	Pass
49	4719	0.185415	Pass	24360	1.452287	Pass	8902	0.008371	Pass	3487	0.344884	Pass
50	15441	1.002997	Pass	2522	0.452480	Pass	11440	0.000000	Pass	3544	0.517847	Pass
51	22013	1.576566	Pass	5268	0.335984	Pass	14295	1.377119	Pass	4292	0.061089	Pass
52	31696	2.102017	Fail	8014	1.172824	Pass	25400	1.699063	Pass	4407	0.748968	Pass
53	16264	0.455842	Pass	11104	0.751067	Pass	4897	0.304821	Pass	4465	0.105769	Pass
54	26985	5.237989	Fail	14879	1.593252	Pass	7753	0.070865	Pass	4580	0.000000	Pass
55	7403	0.488452	Pass	19342	19.157475	Fail	10609	0.014045	Pass	4638	0.059922	Pass
56	17778	1.038612	Pass	11920	0.799521	Pass	13146	0.061836	Pass	4753	0.049665	Pass
57	26770	5.383250	Fail	15352	0.953459	Pass	16319	1.037208	Pass	4810	0.000000	Pass
58	7534	0.342538	Pass	30799	2.203920	Fail	20761	1.863865	Pass	4925	0.214428	Pass
59	14796	0.056587	Pass	15139	1.453378	Pass	3748	0.257044	Pass	4983	0.056081	Pass
60	21022	1.717170	Pass	24407	14.888486	Fail	5652	0.681854	Pass	5098	0.036451	Pass
61	30360	4.781059	Fail	2912	0.125630	Pass	7873	0.077041	Pass	5213	0.100856	Pass
62	14235	1.857763	Pass	4972	0.470596	Pass	9777	0.042216	Pass	5328	0.099571	Pass
63	21153	2.184866	Fail	7717	0.526474	Pass	12315	0.120497	Pass	5386	0.185064	Pass

$\frac{s}{T}$	50			500			5000			50000		
64	30144	2.552300	Fail	9777	0.597689	Pass	15487	1.210152	Pass	55	0.027740	Pass
65	13329	0.769329	Pass	12179	1.410139	Pass	19929	4.461731	Fail	401	0.122479	Pass
66	20937	1.651237	Pass	15612	2.212913	Fail	1013	0.040702	Pass	3300	0.366194	Pass
67	30621	2.597590	Fail	20418	2.837584	Fail	3233	0.357810	Pass	3473	0.930212	Pass
68	14496	1.477439	Pass	3385	0.372117	Pass	5137	0.131134	Pass	3588	0.576482	Pass
69	21068	1.150020	Pass	6474	0.459437	Pass	7358	0.036976	Pass	3703	0.361801	Pass
70	32481	2.514535	Fail	8877	1.385974	Pass	8945	0.068664	Pass	3818	0.508707	Pass
71	21545	1.324073	Pass	11279	2.028850	Fail	11166	0.009583	Pass	3933	0.451872	Pass
72	30883	3.905638	Fail	14369	0.921445	Pass	14021	4.489502	Fail	4048	0.704377	Pass
73	18218	1.090210	Pass	18831	1.541410	Pass	17828	1.381326	Pass	3667	0.494997	Pass
74	26172	1.883333	Pass	1112	0.004646	Pass	1450	0.627006	Pass	3955	0.781568	Pass
75	7281	0.623228	Pass	4888	0.280029	Pass	24294	1.944092	Pass	4070	1.260629	Pass
76	13160	16.363300	Fail	7634	0.600646	Pass	3791	0.459961	Pass	4127	0.061825	Pass
77	18694	1.822178	Pass	9694	0.620936	Pass	6329	1.132326	Pass	4243	0.021542	Pass
78	26303	1.414069	Pass	13469	0.917527	Pass	8233	0.482208	Pass	4415	0.597387	Pass
79	7412	1.122922	Pass	17245	3.687992	Fail	10137	0.092180	Pass	4530	0.025147	Pass
80	13291	0.696512	Pass	31318	3.066807	Fail	12675	0.133623	Pass	4645	0.014544	Pass
81	19517	3.338297	Fail	16345	1.270674	Pass	16482	1.359631	Pass	4703	0.044464	Pass
82	27471	5.310100	Fail	23210	3.004772	Fail	21559	2.281097	Fail	4818	0.030619	Pass
83	8580	0.460486	Pass	2401	0.822611	Pass	1691	2.697986	Fail	3746	0.506367	Pass
84	14114	0.905573	Pass	4804	0.317501	Pass	4545	0.099204	Pass	4034	1.373892	Pass
85	19993	1.692456	Pass	6864	0.790326	Pass	7401	0.575789	Pass	4092	0.439167	Pass
86	28640	1.743205	Pass	8923	1.042940	Pass	9622	0.008301	Pass	4149	0.814634	Pass
87	10440	1.314080	Pass	11669	0.693319	Pass	12161	0.084404	Pass	4264	0.171903	Pass
88	15973	0.084957	Pass	15101	1.263064	Pass	15967	1.593858	Pass	4322	0.056254	Pass
89	22545	1.723070	Pass	816	0.005229	Pass	27707	1.895508	Pass	4379	0.985759	Pass
90	195	0.014833	Pass	2874	0.102962	Pass	12598	0.091061	Pass	4495	0.007467	Pass
91	7112	0.336154	Pass	4934	0.522450	Pass	19578	1.495863	Pass	4552	0.040006	Pass
92	13338	0.817409	Pass	6994	0.956943	Pass	28144	3.049159	Fail	4610	0.041667	Pass
93	20255	3.521982	Fail	9396	0.774862	Pass	12083	0.217153	Pass	258	0.027568	Pass
94	28901	9.224483	Fail	12486	2.347657	Fail	18428	2.032069	Fail	315	0.037982	Pass
95	10701	4.451213	Fail	24843	2.192531	Fail	25726	1.740154	Pass	430	0.052015	Pass
96	17618	3.451393	Fail	10213	0.850204	Pass	7761	0.158668	Pass	488	0.092353	Pass
97	24536	1.792801	Pass	13302	0.951513	Pass	12838	0.100896	Pass	545	0.154369	Pass
98	15328	1.766623	Pass	17421	1.650401	Pass	20452	2.988445	Fail	603	0.079863	Pass
99	24320	2.278785	Fail	45	0.017149	Pass	584	0.141105	Pass	833	0.092385	Pass
100	1971	2.347290	Fail	1075	0.042353	Pass	2805	0.353069	Pass	891	0.215969	Pass

Concluding remarks

Pseudorandom number generators generally have a cycle whereas digits after the decimal of irrational numbers are non-terminating and, more importantly, non-repeating. Provided these digits can be easily and speedily generated by a computer, must we not recommend pi –or its function which may perhaps behave better, i. e., “look more random” (the Z-value in run test of randomness comes close to zero more often for digits of arbitrary length taken from arbitrary positions) as a random number generator? Given that numbers generated from an arbitrarily selected seed may not be random [M+07], we feel the re-confirmation of pi as being random is crucial in the light of our findings revealing the randomness of sequences of arbitrary (or fixed) length from fixed (or arbitrary) positions. At the time of submitting this paper, we are investigating other functions of pi [Cha07] as possible random number generators which may function better than pi itself.

References

- [TF05] Shu-Ju Tu and Ephraim Fischbach, *A study on the randomness of π* , International Journal of Modern Physics C, 16, No. 2, 281–294, 2005.
- [Mar07] Marsaglia, G. *Refutation of claims such as "Pi is less random than we thought"*, InterStat, <http://interstat.statjournals.net/ YEAR/ 2006/abstracts/0601001.php?Name=601001> (12/24/2007).
- [M+07] Matsumoto, M., Wada, I., Kuramoto, A. and Ashihara, H, *Common defects in initialization of pseudorandom number generators*, ACM Transactions on modeling and Computer Simulation, 17(4), article 15, Sept 2007
- [Cha07] S. Chakraborty, Review of the research paper titled *Common Defects in Initialization of Pseudorandom Number Generators* by Matsumoto, Wada, Kuramoto and Ashihara, ACM Transactions on modeling and Computer Simulation, 17(4), article 15, Sept 2007, published in Computing Reviews, Dec 23, 2007 (www.reviews.com) [recommended as a “best review” by the editor]