Golden ratio in science, as random sequence source, its computation and beyond

S.K. Sen, Ravi P. Agarwal*

Department of Mathematical Sciences, Florida Institute of Technology, 150 West University Boulevard, Melbourne, FL 32901-6975, United States

Received 19 June 2007; accepted 21 June 2007

Abstract

Some rational as well as some irrational numbers, among all real numbers in mathematics, are very special and have fascinated many human minds. Associated with these numbers are not only the fascinating history but also remarkable physical phenomena observed by critical minds of scientists, artists, architects, engineers, naturalists and spiritualists. The rational number $2^n$ and the irrational number $\pi$—a transcendental number, for example, have very special places in computer science and in mathematics, respectively. Some of the other famous numbers are the Hilbert number $2^{\sqrt{2}} \approx 2.66514414269023$, the Liouville number $\approx 0.1100010000000000000000010000$ which has a 1 in the 1st, 2nd, 6th, 24th, 120th etc. places and 0s elsewhere, the Euler–Mascheroni constant $\gamma = \lim_{n \to \infty} \left( \sum_{k=1}^{n} \frac{1}{k} - \ln n \right) \approx 0.57721566490153$, and the numbers $e^{-\pi/2} \approx 0.207879576350762, \pi e \approx 22.4591577183611$ (believed (not proved) to be a transcendental number) and $e^\pi \approx 23.1406926327793$. Presented here is yet another exceedingly delightful, extensively explored irrational algebraic number $(1 + \sqrt{5})/2 \approx 1.61803398874989$ called the golden ratio $\phi$ and its widespread occurrence in mathematics, specifically geometry, computational science, biology, artistic creations, architecture, nature and beyond. Specifically, digits—even randomly or systematically chosen consecutive digits or consecutive blocks of digits—of golden ratio may be used as a source of uniformly distributed random numbers. Unlike any of the several quasi- and pseudo-random number generators using various methods, we need to use no method here; only we have to pick up the consecutive/nonconsecutive blocks of digits from the stored golden ratio and hence it would be a fastest means of obtaining random numbers. This idea of getting random sequences possibly opens up a new efficient way of solving numerous optimization problems including the NP-hard travelling salesman problem by polynomial-time heuristics such as ant system approaches, genetic algorithms, simulated annealing and other randomized algorithms. Also, whether these random numbers sieved out of the golden ratio are quasi- (more uniformly distributed) or pseudo-random numbers may be studied including its scope among other random number generators. Presented here is the golden ratio along with its computation up to a desired number of digits using the single Matlab command vpa. Also described are its occurrences in sciences in very many ways and a fixed-point iteration scheme besides other methods for its computation. Demonstrated are the uniform pseudo-random distribution of its digits and its capability to perform the Monte Carlo integrations using systematically its consecutive blocks of digits. Mentioned are some of the interesting happenings/occurrences in nature, art and architecture in which the golden ratio has been discovered in an exact/approximate form. The article is our way of viewing this amazing number, the golden ratio, and depicting its beauty. Included are several Matlab programs for the reader with Matlab facilities. These will enable him/her to have a deeper insight into its character in the background of our aesthetic sense and its extraordinary tendency to pop up in diverse situations through quick computation.

© 2008 Elsevier Ltd. All rights reserved.

* Corresponding author. Tel.: +1 321 674 7714; fax: +1 321 674 7412.
E-mail addresses: sksen@fit.edu (S.K. Sen), agarwal@fit.edu (R.P. Agarwal).

0898-1221/S - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2007.06.030
1. Introduction

Golden ratio (gr) $\varphi$ is the algebraic\(^1\) irrational number \((1 + \sqrt{5})/2\) which is given by 1.618033988749894848204586834366381177203091798058 (up to 50 digits) or, equivalently, exactly by the trigonometric expression 0.5/\(\sin(0.1\pi)\).\(^2\) The symbol $\varphi$ (lowercase form) – the first letter of the 450 BC Greek sculptor Phidias (phi) – to represent the golden ratio (the term is believed to be first used by Martin Ohm (1792–1872)) was proposed by Mark Barr, an American mathematician in around 1909. The golden ratio fascinated intellectuals of diverse interests for over 24 centuries. It can be seen, from the foregoing trigonometric expression, that an algebraic number, say $\varphi$, can be expressed in terms of a transcendental number, say $\pi$, where the expression is evidently a transcendental function. An algebraic number, say $\varphi$, cannot be expressed as an algebraic function of a transcendental number, say $\pi$ (pi). The gr $\varphi$ is also known as or is related/connected to golden section, golden mean, golden number, golden proportion, golden cut, golden rectangle, golden triangle, golden angle, golden ellipse, golden string, golden ratio conjugate (=silver ratio), whirling triangles, whirling rectangles, whirling square, phi ($\varphi$), extreme and mean ratio (defined by Euclid in 300 BC), medial section, divine proportion (defined by Pacioli in 15th century), divine section, Fibonacci numbers, Gaullist cross, and mean of Phidias [1]. Further, it has connection with pentagon, pentagram, decagon, dodecahedron, continued fraction, Euclidean algorithm and the Regula Falsi (numerical) method. Though not as well known as the transcendental number\(^3\) $\pi \approx 3.14159265358979$ which is the ratio of the circumference and the diameter of any circle, the gr, has a tendency to turn up in very many places. While it is not possible to mention the connection and occurrence of this amazing number in all possible places and the beauty associated with it, we have made an attempt to retell in our own way many observed facts about it and to explore some of its characters which we have not come across, although these might have been already recorded or known to others. After all the gr conjured up the curiosity/inquisitiveness of numerous analytical minds over centuries; all of them had their own way of observing its beauty and enjoying the thrill of such observations. As a matter of fact, we can find an enormous amount of information recorded in the literature [1–26] on golden ratio and related materials. Yet newer and newer occurrences of the golden ratio or its connection are being discovered.

In Section 2, we discuss in detail what golden ratio (gr) is and demonstrate the occurrence of gr in science while in Section 3, we describe how an arbitrary number of digits of golden ratio can be produced exactly using the Matlab command vpa. We talk about digits of $\varphi$, their statistical behavior, their usage as a random sequence source and Monte Carlo integration in Section 4. This is to demonstrate the fact that $\varphi$ could be a possible source of easily obtaining random numbers without executing any method/algorithm. Section 5 briefly describes the occurrence/connection of $\varphi$ exactly/approximately in nature, artifacts, and architecture. While it is not that $\varphi$ is embedded in a natural/artificial law/rule, it is we, the concerned human beings, who over centuries, discovered/found out the connection of $\varphi$ in numerous natural/artificial phenomena (already occurred/existed or existing) and were amazed. We include conclusions in Section 6. We have also provided, in these sections, except Section 5, Matlab programs for convenience of the reader who has Matlab software. Matlab is a most widely used user-friendly highest level programming language that needs no formal programming knowledge for its use. Just by copying from the text/word document and then pasting the program in the Matlab M file, we can run the program.

2. Golden ratio in science

According to the Collins Gem English Dictionary, the word science means systematic study and knowledge of natural or physical phenomena. Since mathematics is the language of physics or, for that matter, any materials

---

\(^{1}\) An algebraic number is a root of any rational polynomial.

\(^{2}\) There are several possible proofs. A proof may start from a known formula and then continue using the axioms to arrive at the desired statement.

\(^{3}\) Transcendental number is a number that is not the root of any polynomial (of finite degree) whose coefficients are rational (or integer) numbers. That is, it is not the root of any integer polynomial implying that it is not an algebraic number of any degree. Every real transcendental number must be irrational since a rational number is, by definition, an algebraic number of degree 1. Some of the proven transcendental numbers are $\ln 2$, $\pi$, $2\sqrt{2}$, $\sin(1)$, $\Gamma(1/3)$, and $e^7$. 
science including chemistry and biology, we have considered, possibly without any serious mistake, mathematics including geometry, arithmetic, algebra and body of numbers as a branch/part of science, more specifically exact science. Thus, all that we present in this section is broadly under the designation science. The irrational number \( \pi \) = 3.14159265358979 \( \cdots \) = area of a unit circle = (circumference/diameter) of any circle is widely known and extensively occurs in numerous real-world problems. But there is another irrational (algebraic) number \( \phi \) = 1.61803398874989 \( \cdots \) = exactly the solution \((1 + \sqrt{5})/2\) of the quadratic (algebraic) equation \(x^2 - x - 1 = 0\) that has the same natural tendency for popping up and is not as well known as \( \pi \). It can be seen that the other solution of the foregoing equation is \(1/\phi = 0.61803398874989 \cdots\) which is interestingly the exact fractional part of \( \phi \). The following approximation/iteration procedures lead us to their final destination \( \phi \) which is reached in three distinct ways — (i) monotonically increasing toward \( \phi \), (ii) monotonically decreasing toward \( \phi \), and (iii) oscillatorily moving toward \( \phi \) with varying number of iterations.

**Approximation of \( \phi \) by fractions and square roots.** The amazing number \( \phi \), also called the golden ratio (gr), can be expressed exactly by the following infinite series of continued fractions and that of continued square roots.

\[
\phi = 1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \cdots}}}, \quad \phi = \sqrt{1 + \sqrt{1 + \sqrt{1 + \sqrt{1 + \cdots}}}}.
\]

Using the following iterative Matlab program `goldenratiobyfractionsandsquareroots` we can show that the two series — one based on fractions and the other based on square-roots — do approximate the value of gr (\( \phi \)). Approximation by fractions needs 17 iterations to obtain \( \phi \) correct up to 14 decimal places while that by square-roots needs 14 iterations. In each of the fractions- and the square-roots-approximations for \( \phi \), the sequence of approximations monotonically increases and converges to \( \phi \) (in the limit). Standard Matlab provides maximum 14 decimal places accuracy. The concerned program is as follows:

```
clear all ;format long ;phi=1;disp('No. of iterations Approximation of phi by fractions')
for i=1:25, phi=1+1/(1+1/phi);disp([i phi]), p(i)=phi;end;
plot(p);hold on; %Approximation of phi by fractions
clear all;format long;phi=1;disp('No. of Iterations Approximation of phi by square-roots')
for i=1:25, phi=sqrt(1+sqrt(1+phi));disp([i phi]),p(i)=phi;end
plot (p); %Approximation of phi by square roots
```

Issuing the Matlab command `>> goldenratiobyfractionsandsquareroots` we obtain the following result omitting iterations 19–25 as well as trivial zeros and decimal points to conserve space and including the graphs. The graph (Fig. 1) that starts from 1.5 in the base of the square represents the approximation by fractions while the graph that starts from 1.55377397403004 near the left middle portion of the square represents the approximation by square roots.

<table>
<thead>
<tr>
<th>No. of iterations</th>
<th>Approximation of phi by fractions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.5</td>
</tr>
<tr>
<td>2</td>
<td>1.6</td>
</tr>
<tr>
<td>3</td>
<td>1.61538461538462</td>
</tr>
<tr>
<td>4</td>
<td>1.61764705882353</td>
</tr>
<tr>
<td>5</td>
<td>1.61797752808989</td>
</tr>
<tr>
<td>6</td>
<td>1.61802575107296</td>
</tr>
<tr>
<td>7</td>
<td>1.61803278688525</td>
</tr>
<tr>
<td>8</td>
<td>1.61803381340013</td>
</tr>
<tr>
<td>9</td>
<td>1.61803396316671</td>
</tr>
</tbody>
</table>
Each of the foregoing two limits of infinite series indeed equals $\phi$ exactly. For the limit of the series with fractions, let

$$x = 1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \cdots}}}.$$ 

Then we can write $x = 1 + \frac{1}{x}$, i.e. $x^2 - x - 1 = 0$ and this has $\phi$ as one of the two roots. Similarly we can prove that $\phi$ is the limit of the series with square roots.

Approximation of $\phi$ by golden rectangles. A golden rectangle is one whose sides are related to $\phi$ in the following way. Create a new bigger rectangle by swinging the long side around one of its ends so that the new long side is the sum of the old long side and the old short side while the new short side is the old long side. The ratio of the new long side and the new short side will be the value in the process of approaching $\phi$. Constructing successively increasingly bigger rectangles, we obtain the ratio, in the limit, as $\phi$. However, the sequence of ratios is not monotonically increasing as in the case of approximations by continued fractions and by continued square roots. It is damped oscillatory approaching toward its destination to $\phi$ (i.e., converging to $\phi$). For instance, the $13 \times 8$ rectangle is an approximate golden rectangle. A golden rectangle is thus one having sides exactly in the ratio $1 : \phi$. The following Matlab program (starting from $13 \times 8$ rectangle) named goldenratiobygoldenrectangles
clear all; format long g ; n=13;d=8;
for i=1:35, p=(n+d)/n;d1=n;n1=n+d;disp([i d1 n1 p]);
q(i)=p;n=n1;d=d1; end; plot(q)

creates successively increasing big rectangles (each one is a rectangle successively approaching towards the golden rectangle) such that the long side divided by the short side of each rectangle is an approximation of \( \varphi \). Issuing the Matlab command >> goldenratio_by_goldenrectangles we obtain respectively the iteration number, short side, long side, and the ratio (long side/short side = an approximation of \( \varphi \)) as follows (omitting iteration numbers 7–29 and 34, 35 to conserve space). Besides, we get the graph (Fig. 2) of the successive approximations.

<table>
<thead>
<tr>
<th>i</th>
<th>d1</th>
<th>n1</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>13</td>
<td>21</td>
<td>1.61538461538462</td>
</tr>
<tr>
<td>2</td>
<td>21</td>
<td>34</td>
<td>1.61904761904762</td>
</tr>
<tr>
<td>3</td>
<td>34</td>
<td>55</td>
<td>1.61764705882353</td>
</tr>
<tr>
<td>4</td>
<td>55</td>
<td>89</td>
<td>1.61818181818182</td>
</tr>
<tr>
<td>5</td>
<td>89</td>
<td>144</td>
<td>1.61805555555556</td>
</tr>
<tr>
<td>6</td>
<td>144</td>
<td>233</td>
<td>1.61803398874989</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>14930352</td>
<td>24157817</td>
<td>1.6180339887499</td>
</tr>
<tr>
<td>31</td>
<td>24157817</td>
<td>39088169</td>
<td>1.6180339887499</td>
</tr>
<tr>
<td>32</td>
<td>39088169</td>
<td>63245986</td>
<td>1.6180339887499</td>
</tr>
<tr>
<td>33</td>
<td>63245986</td>
<td>102334155</td>
<td>1.6180339887499</td>
</tr>
</tbody>
</table>

It may be noted that successive rectangles are not exact golden rectangles but they approach toward the golden rectangle. In fact, true (exact) golden rectangles are all those whose sides are exactly in the ratio 1:\( \varphi \). Successive rectangles can be constructed starting from 1 \( \times \) 1 square (a particular case of a rectangle (square) where both the long side and the short side are equal). We then get successively 2 \( \times \) 1, 3 \( \times \) 2, 5 \( \times \) 3, 8 \( \times \) 5, 13 \( \times \) 8, 21 \( \times \) 13, 34 \( \times \) 21 rectangles and so on. As a matter of fact, any rectangle or a square can be the starting figure for the foregoing “swinging” process to arrive at the golden rectangle in the limit. Observe that any rectangle which is not a golden rectangle can be the starting rectangle for the destination \( \varphi \) or, equivalently the golden rectangle.

Defining an amplitude as the signed deviation of the foregoing ratio (long side/short side of the rectangle) from \( \varphi \), we show that the successive amplitudes diminish rapidly and vanish finally. The graph of the amplitudes looks just like one side of a water wave decreasing in size and becoming increasingly smaller ripples to vanish finally (Fig. 3). The successive amplitudes and their plot are obtained by the
Fig. 2. Approximation of golden ratio by successive approximate golden rectangles. The long side/short side of each successive approximate golden rectangle oscillatory approaches toward $\phi$.

Fig. 3. The graph of the successive amplitudes each of which is the deviation of the value long side/short side of an approximate golden rectangle. It looks just like one side of a water wave decreasing in size and becoming increasingly smaller ripples to vanish finally.

following Matlab program named `goldenratiobygoldenrectanglesoscillatoryconv` and issuing the command `>> goldenratiobygoldenrectanglesoscillatoryconv`

clear all; format long g; phi=(sqrt(5)+1)/2; n=13;d=8;
for i=1:35, p=(n+d)/n;d1=n;n1=n+d; %disp([i d1 n1 p]);
    q(i)=p;n=n1;d=d1; end; %plot(q);
 a=q-phi*ones(1,35);
 disp('Successive amplitudes of the iterates around phi')
 disp('(oscillatorily convergent) ending in vanishing ripples'),
 format short g; disp([a']);
disp('Graph of oscillating (around phi) amplitudes converging to 0')
 plot(a)

Successive amplitudes of the iterates around phi (oscillatorily convergent) ending in vanishing ripples are as follows:
Fig. 4 is analogously correct up to 14 decimal places. Also we obtain the plot (Fig. 4) of the successive approximations (which oscillate with amplitudes that eventually become zero in the limit) leading to \( \varphi \) as follows: 

\[
\begin{array}{cccc}
-0.0026494 & -2.5583e-008 & -2.4669e-013 \\
0.0010136 & 9.7719e-009 & 9.4147e-014 \\
-0.00038693 & -3.7325e-009 & -3.5971e-014 \\
0.00014783 & 1.4257e-009 & 1.3767e-014 \\
-5.6461e-005 & -5.4457e-010 & -5.3291e-015 \\
2.1567e-005 & 2.0801e-010 & 1.9984e-015 \\
3.1465e-006 & 3.0348e-011 & 2.2204e-016 \\
4.5907e-007 & 4.4276e-012 & 0 \\
-1.7535e-007 & -1.6913e-012 & 0 \\
6.6978e-008 & 6.4593e-013 &
\end{array}
\]

The golden cuboid. While the golden rectangle is a 2-D example/construction, the golden cuboid\(^4\) is analogously a 3-D example/construction. Consider the problem of determining the dimensions of a cuboid of unit volume that has a diagonal of length 2 (units). If \( a, b, c \) are the sides of the cuboid then \( a.b.c = 1, \sqrt{a^2+b^2+c^2} = 2 \). If we choose, without loss of generality, \( b = 1 \), then the positive solution of the foregoing two equations produces \( a = \varphi \), the gr. It is now easy to see that \( a:b:c = \varphi:1:\varphi^{-1} \) and the diagonal of the golden cuboid is \( \sqrt{\varphi^2 + 1 + \varphi^{-2}} = 2 \). Thus \( \varphi, 1, \varphi^{-1} \) are the dimensions of a cuboid of unit volume. The golden cuboid has the exact length \( \varphi \), the exact width 1, and the exact height \( \varphi^{-1} \). The approximate numerical ratio, viz. length:width:height of the golden cuboid is \( \varphi:1:\varphi^{-1} = 1.618033988749891:1:0.618033988749897 \). Also the length \( \varphi \) of the golden cuboid is exactly equal to \( 2\varphi^{-1} + \varphi^{-2} \). The total surface area of the golden cuboid is \( 2(1 \times \varphi) + 2(1 \times \varphi^{-1}) + 2(\varphi \times \varphi^{-1}) = 2(\varphi + 1 + \varphi^{-1}) \approx 6.47213595499957 \) obtained by the Matlab command >> \texttt{format long g; p=1.618033988749891; s=2*(p+1+p^-1)} while four of the six faces of the golden cuboid are golden rectangles. Observe that the exact ratio of the surface area \( 4\pi \times 1^2 \) of the sphere circumscribing the golden cuboid to the surface area \( 2(\varphi + 1 + \varphi^{-1}) \) of the cuboid is \( 4\pi \times 1^2/(2(\varphi + 1 + \varphi^{-1})) = \pi/\varphi \) — yet another interesting connection of \( \pi \) and \( \varphi \). There are many interesting connections of \( \varphi \) that we can discover in many aspects of geometry. In fact, we cannot say at any point in time in future that all the \( \varphi \) connections have been explored in geometry and no new connection is possible. The pure joy of such discoveries varies in intensity/depth from discoverer to discoverer.

Approximation of \( \varphi \) by oscillatory infinite series. The gr \( \varphi \) can be expressed by the following damped oscillatory infinite series that converges to \( \varphi \) in the limit.

\[
\varphi = \frac{13}{8} + \sum_{i=0}^{\infty} \frac{(-1)^{i+1}(2i+1)!}{(i+2)!4^{2i+3}}.
\]

The Matlab program \texttt{goldenratiobyinfiniteseries} is

\footnotesize
\begin{verbatim}
clear all; format long g; p=13/8; s=1;j=0;
for i=0:25,p=p+s*factorial(2*i+1)/(factorial(i+2)*factorial(i)*4^(2*i+3));
disp ([i p]);j=j+1;q(j)=p; s=-s;
end; plot(q)
\end{verbatim}

\end{footnotesize}

Issuing the Matlab command \texttt{>> goldenratiobyinfiniteseries} we obtain the number of iterations and the corresponding series value as follows (omitting iterations 24–25). 20 iterations are needed to get \( \varphi \) correct up to 14 decimal places. Also we obtain the plot (Fig. 4) of the successive approximations (which oscillate with amplitudes that eventually become zero in the limit) leading to \( \varphi \) as follows:

\footnotesize
\begin{verbatim}
-0.0026494 -2.5583e-008 -2.4669e-013
0.0010136 9.7719e-009 9.4147e-014
-0.00038693 -3.7325e-009 -3.5971e-014
0.00014783 1.4257e-009 1.3767e-014
-5.6461e-005 -5.4457e-010 -5.3291e-015
2.1567e-005 2.0801e-010 1.9984e-015
3.1465e-006 3.0348e-011 2.2204e-016
-1.2019e-006 -1.1592e-011 -2.2204e-016
4.5907e-007 4.4276e-012 0
-1.7535e-007 -1.6913e-012 0
6.6978e-008 6.4593e-013
\end{verbatim}

\end{footnotesize}

\(^4\) A cuboid may also be called a rectangular parallelepiped.
Approximation of $\varphi$ by Fibonacci numbers. Consider the sequence of numbers 0, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, and so on where each number (starting from 2) is the sum of the preceding two numbers. The number 2 is the sum of 0 and 1, 3 is the sum of 1 and 2, and so on. These numbers are called Fibonacci numbers and the sequence is called the Fibonacci sequence (name given by E. Lucas (1842–1891)). The sequence obtained by dividing each number (starting from 2) in the Fibonacci sequence by the preceding number converges (oscillatorily) to the golden ratio, also called the golden mean, $\varphi$. The following Matlab program goldenratioofibonacci outputs respectively iteration number $i$, Fibonacci number $F(i-1)$, Fibonacci number $F(i)$, and the ratio $F(i)/F(i-1)$ that is an approximation of $\varphi$ in the

![Graph](image-url)

**Fig. 4.** Approximation of golden ratio $\varphi$ by the oscillatory infinite series. It is fast converging. In 20 iterations, we obtain 14 decimal places accuracy.
Fig. 5. Golden ratio $\phi$ by Fibonacci numbers $F(i)$, where $F(0) = 0$, $F(1) = 1$, $F(2) = F(0) + F(1) = 2$, $F(3) = F(1) + F(2) = 3$, $F(4) = F(2) + F(3) = 5$, and so on. Successive approximations of $\phi$ demonstrating damped oscillatory motion of the approximations that eventually converge to $\phi$.

limit. Also it plots the graph of the successive approximations of $\phi$ demonstrating damped oscillatory motion (Fig. 5) of the approximations that eventually converge to $\phi$.

clear all; format long g; f(1)=0;f(2)=1;
for i=3:43
   f(i)=f(i-1)+f(i-2); p(i)=f(i)/f(i-1);
   disp([i f(i-1) f(i) p(i)]);
end; plot(p)

Issuing the Matlab command >> goldenratio_by_fibonacci we obtain the following output:

<table>
<thead>
<tr>
<th>$i$</th>
<th>$F(i-1)$</th>
<th>$F(i)$</th>
<th>$F(i)/F(i-1)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>3</td>
<td>1.5</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>5</td>
<td>1.66666666666667</td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td>8</td>
<td>1.6</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>13</td>
<td>1.625</td>
</tr>
<tr>
<td>9</td>
<td>13</td>
<td>21</td>
<td>1.61538461538462</td>
</tr>
<tr>
<td>10</td>
<td>21</td>
<td>34</td>
<td>1.61904761904762</td>
</tr>
<tr>
<td>11</td>
<td>34</td>
<td>55</td>
<td>1.61764705882353</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>10946</td>
<td>17711</td>
<td>1.61803398501736</td>
</tr>
<tr>
<td>24</td>
<td>17711</td>
<td>28657</td>
<td>1.6180339901756</td>
</tr>
<tr>
<td>25</td>
<td>28657</td>
<td>46368</td>
<td>1.61803398820533</td>
</tr>
<tr>
<td>26</td>
<td>46368</td>
<td>75025</td>
<td>1.6180339889579</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>39088169</td>
<td>63245986</td>
<td>1.61803398874999</td>
</tr>
<tr>
<td>41</td>
<td>63245986</td>
<td>102334155</td>
<td>1.61803398874989</td>
</tr>
<tr>
<td>42</td>
<td>102334155</td>
<td>165580141</td>
<td>1.61803398874989</td>
</tr>
<tr>
<td>43</td>
<td>165580141</td>
<td>267914296</td>
<td>1.61803398874989</td>
</tr>
</tbody>
</table>
Pascal and Chinese triangles: Fibonacci and hence golden ratio connections If we consider each number of the $n$-row Pascal triangle (created by Blaise Pascal (1623–1662)) as an equally spaced point, then the $n$-row Pascal triangle will be an equilateral triangle. The apex of the triangle is the point representing “1”. Call this apex (vertex) as the 0th ($n = 0$) row of the equilateral (Pascal) triangle. The 0-row Pascal triangle is

\[
\begin{array}{c}
1
\end{array}
\]

The 1-row The Pascal triangle is

\[
\begin{array}{c}
1 \\
1 1
\end{array}
\]

The 5-row Pascal triangle is

\[
\begin{array}{ccccccc}
 & & & & & & 1 \\
 & & & & 1 & 1 & \\
 & & 1 & 2 & 1 & \\
 & 1 & 3 & 3 & 1 & \\
1 & 4 & 6 & 4 & 1 &
\end{array}
\]

The row $n = 1$ has two numbers 1 and 1 placed symmetrically below the row $n = 0$. All other numbers are the sums of the just two above numbers as shown in the foregoing 5-row Pascal triangle. Thus we can construct the $n$-row Pascal triangle for a positive finite integer $n$. It may be seen that the sum of all the numbers in the $n$th row is $2^n$ and $(2^n - 1)$st row has all odd numbers. The coefficients of the binomial expansion $(x + y)^n$ can be readily obtained from the $n$th row of the triangle. For example, when $n = 3$, the coefficients of the binomial expansion of $(1 + x)^3 = x^3 + 3x^2y + 3xy^2 + y^3$ are obtained from the 3rd row, viz., “1 3 3 1” of the Pascal triangle. The $n$th row of the Pascal triangle can be written as

\[
\begin{array}{cccccc}
\binom{n}{0} & \binom{n}{1} & \binom{n}{2} & \cdots & \binom{n}{n-1} & \binom{n}{n}.
\end{array}
\]

It can be easily shown that $\binom{n}{0} + \binom{n}{1} + \binom{n}{2} + \binom{n}{3} + \cdots + \binom{n}{n-1} + \binom{n}{n} = 2^n$, where

\[
\binom{n}{r} = \frac{n!}{(n-r)!r!}.
\]

By displacing the numbers of Pascal triangle to the left we get a Chinese triangle which is a right-angled triangle. Thus the 5-row Chinese triangle is

\[
\begin{array}{cccccc}
1 & \\
1 & 1 & \\
1 & 2 & 1 & \\
1 & 3 & 3 & 1 & \\
1 & 4 & 6 & 4 & 1 &
\end{array}
\]

The Pascal triangle may be used as a computing device to solve problems in permutations $n_P_r = \frac{n!}{(n-r)!}$ and combinations $n_C_r = \frac{n!}{(n-r)!r!}$. For other properties of the Pascal triangle as well as further information on the Chinese triangle, refer [2].

\footnote{For convenience we have called actual $(n + 1)$-row Pascal equilateral triangle as $n$-row Pascal triangle. For instance, in the binomial expansion of $(x + y)^n$, the $n$ goes perfectly well with the terminology that we have used.}
The accuracy is a monotonically increasing linear function of the number of iterations. Following numerical result and the corresponding accuracy versus iteration graph (\(x=0.000001\)) for 14 digit accuracy. If we now vary the accuracy of \(x=5\) or \(x=1\), we will need 36 iterations to get the golden ratio correct up to 14 significant digits. One may replace \(x\) where 35 is the number of iterations used to get 14 significant digit accuracy for the golden ratio. If we take the initial 

The Matlab code named goldenratiobyfixedpointiteration for the scheme is

```matlab
>> format long g; ratios=[1/1 2/1 5/3 8/5 13/8 21/13 34/21 55/34 89/55 144/89 233/144]

1.61797752808989, 1.61805555555556, 1, 2, 1.66666666666667, 1.6, 1.625, 1.61538461538462, 1.61904761904762, 1.61764705882353, 1.61818181818182, 1.61797752808989, 1.61805555555556,
```

which approach oscillatorily to the golden ratio \(\varphi = 1.61803398874989\ldots\). 

Golden ratio by fixed point iteration. A physically concise fixed point oscillatorily convergent iteration scheme to get the golden ratio correct up to 14 significant digits is 

\[x_{i+1} = 1 + \frac{1}{x_i}, \quad i = 1, 2, 3, \ldots, \text{till } |x_{i+1} - x_i|/|x_{i+1}| \leq 0.5 \times 10^{-14},\]

where \(x_0 = 1\) (an initial approximation).

The Matlab code gives the output as 35 1.61803398874999, where 35 is the number of iterations used to get 14 significant digit accuracy for the golden ratio. If we take the initial approximation \(x_0 = 2\), then we get the same solution in 34 iterations. For \(x_0 = 3\), it is 35 iterations while, for \(x_0 \geq 4\), we will need 36 iterations to get the golden ratio correct up to 14 significant digits. One may replace \(x=1\) in the first line of the foregoing by \(x=5\) or \(x=6\) or \(x=100\), and observe that only 36 iterations are all that are required for the 14 digit accuracy. If we take too small an initial approximation, such as \(x=0.000001\), then we need just 37 iterations for 14 digit accuracy. If we now vary the accuracy of \(\varphi\) keeping the initial approximation \(x_0 = 1\), then we get the following numerical result and the corresponding accuracy versus iteration graph (Fig. 6) using the Matlab command. The accuracy is a monotonically increasing linear function of the number of iterations.

<table>
<thead>
<tr>
<th>Accuracy of (\varphi) desired</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>12</th>
<th>14</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of iterations required</td>
<td>11</td>
<td>16</td>
<td>21</td>
<td>26</td>
<td>31</td>
<td>35</td>
</tr>
</tbody>
</table>

A second order (order of convergence = 2) fixed point iteration scheme to get the golden ratio correct up to 14 significant digits is 

\[x_{i+1} = \frac{x_i^2 + 1}{2x_i - 1}, \quad i = 0, 1, 2, 3, \ldots, \text{till } |x_{i+1} - x_i|/|x_{i+1}| \leq 0.5 \times 10^{-14},\]

where \(x_0 = 1\) (an initial approximation).

---

6 Let the order of convergence of a fixed-point iterative scheme be \(k \geq 1\). Roughly speaking, if the \(i\)th iterate \(x_i\) is correct up to \(d\) decimal places, then the \((i+1)\)th iterate \(x_{i+1}\) will be correct up to \(kd\) decimal places. For the Newton method, \(k = 2\). Here if the iterate \(x_i\) is correct up to three decimal places then the next iterate \(x_{i+1}\) will be correct roughly up to six decimal places.
It can be seen that the foregoing scheme is the Newton scheme which will converge for any finite initial approximation $x_0$ since the given function $f(x) = x^2 - x - 1$, whose one of the zeros is the golden ratio, is a polynomial. The foregoing scheme, where $x_0 = 1$, gives us the successive monotonically decreasing convergent iterates as 2, $1.6666666666667$, $1.61904761904762$, $1.61803444782168$, $1.61803398874999$, $1.61803398874989$, $1.61803398874989$ to get 14 significant digit accuracy in just seven iterations.

The Matlab code named as goldenratiowmethod is

```matlab
clear all; format long g; %Newton method to compute golden ratio.
x=1; for i=1:100, x1=(x^2+1)/(2*x-1), if abs((x1-x)/x1)>.5*10^-14,x=x1; else disp([i x1]); break; end; end;
```

By issuing the Matlab command `>> goldenratiowmethod` in the command window, we obtained the foregoing result. If we now take $x_0 = 1000$, i.e. if we replace in the second line of the Matlab code “x=1” by “x=1000”, then we get the successive monotonically decreasing convergent iterates as $500.250625312656$, $250.376563280076$, 125.440782875015, 62.9753938073151, 31.7447708421681, 16.138518926688, 8.3618774364671, 4.5104341940416, 2.66106159796177, 1.86974050988847, 1.61826833841543, 1.61803398874999, 1.61803398874989 to get 14 significant digit accuracy in 15 iterations.

The Newton scheme for any single polynomial equation having distinct roots, will always converge to a root for any chosen finite initial approximation. However, due to finite precision of the computer, the scheme will oscillate around the root if the encountered root is a multiple (repeated) root. This is always so for a (finite degree) polynomial equation since in the Newton scheme

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}, \quad i = 0, 1, 2, \ldots,$$

until $|\frac{x_{i+1} - x_i}{x_{i+1}}| \leq 0.5 \times 10^{-14}$, both the numerator and the denominator $f(x_i), f'(x_i)$ tend to 0 at the multiple root while the numerator $f(x_i)$ tends to 0 faster than the denominator $f'(x_i)$ and these are all known facts (also see deflated Newton method for repeated roots) [5]. The root clusters (closely spaced roots) problem, however, is ill-conditioned with respect to any polynomial-time method [40]. We may, however, use an integer arithmetical method to get exact roots of a rational root-cluster. But such a method is nonpolynomial time [44].

---

7 Usually an initial approximation converges to the nearest zero of a function $f(x)$. For the foregoing function $f(x) = x^2 - x - 1$, the initial approximation $x_0 = 1$ is closer to the zero 1.61803398874989 of $f(x)$, the iterates converge to this zero of $f(x)$, viz., 1.61803398874989.

To obtain the other zero, viz., $-0.61803398874989$, we take $x_0 = -1$ (in the second line of the following Matlab code we replace “x=1” by “x=-1”) which is nearer the zero $-0.61803398874989$ and get the iterates $-0.61803398874989$, $-0.61803398874989$, $-0.61803398874989$, $-0.61803398874989$, $-0.61803398874989$, $-0.61803398874989$, $-0.61803398874989$, $-0.61803398874989$ in six iterations (for 14 significant digit accuracy).
However, the equation $x^2 - x - 1 = 0$ for golden ratio has neither multiple roots nor root clusters. It has no complex roots. So no complex initial approximation for the foregoing Newton scheme is required. Nevertheless if one chooses a complex (imaginary part ≠ 0) initial approximation and evidently use a complex arithmetic, then the iterates will still converge to the golden ratio which is real. For example, if we choose a complex initial approximation, $x_0 = 2 + 3i$, i.e. in the second line of the foregoing Matlab code,\(^8\) if we replace “$x = 1$” by “$x = 2 + 3i$”, then the successive iterates will be

$$
1.33333333333333 + 1.33333333333333i,
1.12734082397004 + 0.329588014981273i,
1.59443486195394 − 0.24539865876194i,
1.59095141116518 − 0.000781073576091225i,
1.61836985396791 + 1.96304078729883e−005i,
1.61803403901069 + 5.8944584000414e−009i,
1.6180339887499 + 2.64983146525741e−016i,
1.61803398874989 + 3.08690656551585e−031i
$$

to get 14 significant digit accuracy in eight iterations.

Although we have discussed the fixed-point iteration schemes in connection with the golden ratio, this discussion is, in general, valid for general single-variable polynomial equation solving using a fixed-point iteration scheme.

There are infinity of fixed-point iterative schemes that are possible. We may readily develop an iterative scheme of higher (>2) order convergence as well as of lower order convergence. Higher order schemes may be developed using more terms of the Taylor series. However, these schemes need not be computationally more economical. Most often, a second-order or a third-order scheme has been found to be computationally most economical. If we use the fixed point iteration scheme $x_{i+1} = x_i^2 − 1$, $x_0 = 2$ then the scheme will diverge. The successive iterates will be $3, 8, 63, 3968, 15745023, \ldots$. If we take $x_0 = 1$, then the foregoing scheme will oscillate producing $0$ and $−1$ alternately without ever converging to the golden ratio. If we choose $x_0 = 1.5$ then the iterates will oscillate and finally will produce $0$ and $−1$ alternately. Exactly this fixed oscillation will happen for $x_0 = 1.6$ as well as for $x_0 = 1.618$ and even for $x_0 = 1.61803398874989$ — a choice of initial approximation extremely close to the golden ratio. If we have a computer with infinite precision (word length is infinite) and with the real arithmetic, and $x_0 = $ the exact golden ratio, then only the scheme “$x_{i+1} = x_i^2 − 1$, $x_0 = $ exact golden ratio” will remain fixed and will produce the value of each iterate as exactly the golden ratio. However, in an artificial-world environment, having a computer with infinite precision, is impossible. A scheme converges/diverges/oscillates depending mainly on the choice of the initial approximation $x_0$.

Other φ connections

(i) **Trisectors of an angle.** Let the trisectors of an angle $3\theta$ of a triangle divide its base (straight line) into three segments of lengths equal to the three members of the Fibonacci sequence $f_n$, $f_{n+1}$, $f_{n+3}$. Observe that

$$\lim_{n→∞} f_{n+1}/f_n = ϕ, \lim_{n→∞} f_{n+3}/f_n = ϕ^3.$$

Then $\lim_{n→∞} \cos \theta = 1/2\sqrt{ϕ^2 + 1} = 1/2\sqrt{ϕ + 2}$. The limiting value of $θ$ is $θ = 0.314159265358979 \ldots$ (in radian) or, equivalently, $18^\circ$.

(ii) **Log of the golden mean.** Consider the equation $\lim_{n→∞} (n^n + (n+a)^n) = \lim_{n→∞} (n+2a)^n$. The value of $a$ that satisfies the equation is $a = log_e ϕ = log_e 1.61803398874989 \approx 0.481211825059603$.

(iii) **Conic sections.** Eccentricity $c$ is a measure of how much the conic section deviates from being circular. The eccentricity $c$ of the next sections are as follows: For a circle, $c = 0$; for a non-circle ellipse, $0 < c < 1$; for a parabola, $c = 1$; for a hyperbola, $1 < c < ∞$, while for a straight line, $c = 1$ or $∞$ depending on the definition of eccentricity used. We have $ϕ = \frac{1+\sqrt{5}}{2}$, while we define $ϕ' = \frac{1−\sqrt{5}}{2}$. Then, for the ellipse $\frac{x^2}{c^2} + \frac{y^2}{2} = 1, c^2 = −ϕ'$; for the parabola $y^2 = 4x$, $c^2 = ϕ + ϕ' = 1$; for the hyperbola $\frac{x^2}{c^2} − \frac{y^2}{2} = 1, c^2 = ϕ$.

---

\(^8\) In Matlab, if a complex number is encountered then automatically the computation will be done appropriately in complex arithmetic. The user, unlike many other programming languages such as C and C++, need not consciously take care of the arithmetic in his program.
(iv) **Golden triangle.** The golden triangle is an isosceles triangle such that the ratio of the hypotenuse (i.e. one of the two equal sides of the triangle) \( a \) to the base (i.e. the smallest side of the triangle) \( b \), viz. \( b/a = \varphi \), the golden ratio. The triangle has the vertex angle \( \theta \) (i.e. the angle between the two equal sides) given by \( \theta = 2 \sin^{-1} \left( \frac{1}{\varphi} \right) = \frac{\pi}{\varphi} \approx 0.62831830717959 \). Observing that \( \varphi = \frac{1 + \sqrt{5}}{2} \approx 1.618033988749895 \) and \( \varphi' = \frac{1 - \sqrt{5}}{2} \approx -0.618033988749895 \), we have the following connections of \( \varphi \) with trigonometric functions:

\[
\begin{align*}
4 \sin^2(\pi/20) &= 2 - \sqrt{\varphi} + 2 \approx 0.0978869674096929, \\
4 \cos^2(\pi/20) &= 2 + \sqrt{\varphi} + 2 \approx 3.90211303259031, \\
4 \sin^2(3\pi/20) &= 2 - \sqrt{\varphi'} + 2 \approx 0.824429495415054, \\
4 \cos^2(3\pi/20) &= 2 + \sqrt{\varphi'} + 2 \approx 3.17557050458495. 
\end{align*}
\]

(v) **Golden ratio conjugate** is also called the *silver ratio* and is the quantity \( 1/\varphi = \frac{\sqrt{5} - 1}{2} \approx 0.618033988749895 \).

(vi) **Golden section** is a straight line segment sectioned into two segments, viz., the longer segment \( a \) and the shorter segment \( b \) according to the golden ratio \( \varphi \) such that \( \frac{a+b}{a} = \frac{a}{b} \). If we choose \( b = 1 \), then \( a = \frac{1 + \sqrt{5}}{2} = \varphi \) as one of the roots. The other root, viz. the other value of \( a \) will be \( 2\varphi, 2\varphi' \). Other names frequently used for or closely related to the golden section are the *golden ratio*, the *golden mean*, the *golden number*, phi(\( \varphi \)), extreme and mean ratio, medial section, divine section, golden proportion, golden cut, mean of Phidias and the *divine proportion*.

(vii) **Order of convergence of regula falsi method.** The regula falsi method also known as the linear interpolation or the method of false position [5] is a fixed point iteration scheme derived from the Newton scheme by replacing the first derivative of the function \( f(x) \) in the nonlinear equation \( f(x) = 0 \) by its backward difference form. Thus the need to compute the analytical derivative which may be amplified in transcendental/trigonometric function – particularly in a product form – is obviated in the regula falsi method. The method computes a solution of the given equation \( f(x) = 0 \) starting from two (supplied) initial approximations \( x_0, \, x_1 \). The order of convergence of this method is less than (as it should be) that of the Newton method which is 2 and is *exactly* \( \varphi \) (\( \approx 1.618033988749894872 \)). The regula falsi method is an important numerical method particularly for solving a system of nonlinear equations — algebraic/transcendental.

(viii) **Logarithmic spiral.** A logarithmic spiral, also known as the equiangular spiral, growth spiral, golden spiral, or Bernoulli spiral was first defined by Descartes and then extensively studied by J. Bernoulli. In polar coordinates \((r, \theta)\), a logarithmic spiral (curve) can be generated by the equation \( \theta = \frac{1}{\cot(\alpha)} \ln(r/a) \), where the angle \( \alpha \) between the tangent and the radial line at the point \((r, \theta)\) is constant [6]. The foregoing equation is derived from the (polar) equation \( r = ae^{b\theta} \), where \( b = \cot \alpha \). If the constant angle \( \alpha \) is such that \( \tan \alpha = \frac{\pi}{\varphi} \), where \( \varphi \) is the golden ratio, then we get \( \alpha \approx 1.2735250220895 \) (in radian) \( \approx 72.967608870385 \) (in degree). Using the equation \( r = ae^{b\theta} \), we get the following logarithmic spirals (Fig. 7) for \( a = 1 \) and \( a = 1.2 \) using the following Matlab code named $\text{goldenratiologarithmspiral}$.

```matlab
b=0.306348962530033; t=0:.01:6*pi, r=exp(t*b);polar(t,r,‘r’);
hold on; a is 1 in the foregoing expression for r. The logarithmic spiral
% for a=1 is represented by red dashed line.
t=0:.01:6*pi, r=1.2*exp(t*b);polar(t,r) % a is 1.2 in expression for r here.
% The logarithmic spiral for a=1.2 is shown by a blue continuous line.

and issuing the Matlab command >> $\text{goldenratiologarithmspiral}$.
```

3. \( \varphi \): **Generation to an arbitrary number of digits by Matlab command vpa**

The golden ratio \( \varphi \) given by \( \varphi = (1 + \sqrt{5})/2 \) can be computed to an arbitrary number of digits using the Matlab command \textit{vpa} denoting variable precision arithmetic. The command to be issued is >> $\text{vpa}((1+\text{sqrt}(5))/2$,}
10 000) if we desire to obtain 10 000 digits of $\phi$. All the 10 000 digits will be produced in one single line in the Matlab command window. The last digit is rounded in a way which is not a conventional rounding [5]. To obtain $\phi$ up to 50 digits, we use the command

>>> phi = vpa('(1+sqrt(5))/2', 50)

The output is $\phi = 1.618033988948346556381177203091798058$.

The following table depicts the rounding aspects of Matlab command vpa.

<table>
<thead>
<tr>
<th>Matlab command vpa</th>
<th>Matlab output</th>
<th>Rounding rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>phi = vpa('(1 + sqrt(5))/2', 10)</td>
<td>1.618033988</td>
<td>Not obeyed</td>
</tr>
<tr>
<td>phi = vpa('(1 + sqrt(5))/2', 12)</td>
<td>1.61803398875</td>
<td>Obeyed</td>
</tr>
<tr>
<td>phi = vpa('(1 + sqrt(5))/2', 14)</td>
<td>1.6180339887499</td>
<td>Obeyed</td>
</tr>
<tr>
<td>phi = vpa('(1 + sqrt(5))/2', 16)</td>
<td>1.618033988749895</td>
<td>Obeyed</td>
</tr>
<tr>
<td>phi = vpa('(1 + sqrt(5))/2', 18)</td>
<td>1.61803398874989485</td>
<td>Obeyed</td>
</tr>
<tr>
<td>phi = vpa('(1 + sqrt(5))/2', 22)</td>
<td>1.618033988749894848204</td>
<td>Not obeyed</td>
</tr>
</tbody>
</table>

However, it is not difficult to get first $k$ digits exactly (i.e. truncating all the digits starting from $(k + 1)$st digits and continuing to $\infty$) just by varying/increasing the number of digits around/after $k$.

4. $\phi$: Statistical behaviour, random sequence source and Monte Carlo integration

We have used several pseudo- and quasi-random number generators [7,8] for Monte Carlo single, double and triple integrations. We have compared the performances of these generators with respect to the quality of the result (error) as well as the cost of producing the result (computational/time complexity) [7,8]. All these methods involve arithmetical/non-arithmetical operations to generate each one of the thousands of random numbers that we produce for this purpose. Here we explore the possibility of obviating executing each such generator having thousands of loops of computation and obtaining readily available random numbers straightway from the digits of golden number $\phi$. In order to achieve this we have statistically (using the chi-square test at 10% significant level) determined the distribution of the digits of $\phi$ and found that randomly chosen consecutive digits, say 50, 100, 150, 200, . . . are uniformly distributed.

The following Matlab program goldenratiochisquaretest1randomnumbersource statistically determines the distribution of the digits 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9 from a randomly chosen set of consecutive digits of golden ratio $\phi$. It is found from the output of the program that these digits are uniformly distributed.
zero=0; one=0; two=0; three=0; four=0; five=0; six=0; seven=0; eight=0; nine=0;

% 10000 digit golden ratio (without first two characters ”1.”) with a blank after each digit is
% partially entered in this program to conserve space. The required 10000 digits can be
% readily obtained using the the program FileConvertG in Appendix. The user needs to
% enter the complete 10000 digits before running/executing this program

goldenratio=[6 1 8 0 3 3 9 8 7 4 9 8 4 8 2 0 4 5 8 6 8 3 4 3 6 5 6 3 8 1 1 7 7 2 0 3 0 9 1 7 9 8 0 5 7 6
2 8 6 2 1 3 5 4 4 8 6 2 2 7 0 5 2 6 0 4 6 2 8 1 8 9 0 2 4 4 9 7 0 7 2 0 7 2 0 4 1 8 9 3 9 1 1 3 7 4
8 4 7 5 4 0 8 8 0 7 5 3 8 6 8 9 1 7 5 2 1 2 6 6 3 3 8 6 2 2 3 5 3 6 9 3 1 7 9 3 1 8 0 0 6 0 7 6 6
7 2 6 3 5 4 4 3 3 8 9 0 8 6 5 9 5 9 3 9 5 8 2 9 0 5 6 3 8 3 2 2 6 6 1 3 1 9 9 2 8 2 9 0 2 6 7 8 8
.
.
6 2 5 8 7 0 6 1 5 4 3 3 0 7 2 9 6 3 7 0 3 8 1 2 7 5 1 5 1 7 0 4 0 6 0 5 0 5 7 5 9 4 8 8 2 7 2 3 8
5 6 3 4 5 1 5 6 3 9 0 5 2 6 5 7 7 1 0 4 2 6 4 5 9 4 7 6 0 4 0 5 5 6 9 5 0 9 5 9 8 4 0 8 8 8 9 0 3 7
6 2 0 7 9 9 5 6 6 3 8 8 0 1 7 8 6 1 8 5 5 9 1 5 9 4 4 1 1 1 7 2 5 0 9 2 3 1 3 2 7 9 7 7 1 1 3 8 0 3];

[m n]=size(goldenratio)
for i=1:((m-1)*rand(1)+1),
for j=1:n,
q=goldenratio(i, j);
if q==0, zero=zero+1; elseif q==1, one=one+1;
elseif q==2, two=two+1; elseif q==3, three=three+1;
elseif q==4, four=four+1; elseif q==5, five=five+1;
elseif q==6, six=six+1; elseif q==7, seven=seven+1;
elseif q==8, eight=eight+1; else nine=nine +1;
end;
end;
end;

obsfreq=[zero one two three four five six seven eight nine], e=ones(1, 10);
totaldigits=zero+one+two+three+four+five+six+seven+eight+nine,
expfreq=(totaldigits/10)*e,
teststatisticchisquare=sum((expfreq-obsfreq). 2./expfreq),
criticalvalueofchisquare=14.684, %at 10% significance level with 9 degrees of freedom
if teststatisticchisquare<criticalvalueofchisquare,
’Accept H0, i.e. randomly selected consecutive digits are uniformly distributed at 10% significance level’,
else
’Reject H0, i.e. randomly selected consecutive digits are not uniformly distributed at 10% significance level’
end;
x=[0 1 2 3 4 5 6 7 8 9]; bar(x, obsfreq)

The output. Issuing the following Matlab command in the command window

>> goldenratiochisquaretest1randomnumbersource

we obtain the output result

m = 120 (randomly selected number of rows of golden ratio); n = 50 (number of fixed columns of golden ratio)
obsfreq = 145 137 159 126 142 130 136 147 138 140 (0 occurs 145 times out of 1400 digits, 1 occurs 137 times out of
1400 digits and so on.)
totaldigits = 1400 (First 1400 consecutive digits, excluding the first digit 1, of the golden ratio have been randomly
chosen.)
expfreq = 140 140 140 140 140 140 140 140 140 140 (0 should occur exactly 140 times out of 1400 digits, 1 should
occur exactly 140 times out of 1400 digits, and so on.)
teststatisticchisquare = 5.4571 (This is the test statistic $\chi^2$-value computed from the data.)
criticalvalueofchisquare = 14.6840 (This is the critical value of $\chi^2$ obtained from the $\chi^2$ table at 10% significance level.)

Accept H0, i.e. randomly selected consecutive digits are uniformly distributed at 10% significance level (Acceptance of the null hypothesis $H_0$ implies that the digits of the golden ratio are a potentially good source of random numbers straightway without executing a random number generator loop thousands of times.)

It may be seen that the chi-square goodness-of-fit test (critical value approach as well as $P$-value approach) must satisfy the following two assumptions:

(a) All expected frequencies are one or greater.

(b) At most 20% of the expected frequencies are less than five.

It may be observed that the chi-square test is always a right-tailed test; also the higher the significance level is, the lower is the critical value of $\chi^2$. A bar diagram depicting the distribution of the ten digits 0, 1, 2, ...., 9 in the first 1400 digits of $\phi$ is as in Fig. 8.

If we desire to see the distribution of the digits of the golden ratio starting from the second digit 6 up to $k$ digits, where $k$ is a multiple of 50, then precede the Matlab program goldenratiochisquaretest1randomnumbersource by the Matlab command

\[
r = \text{input}('Enter a value in [1, 200] of the no. of rows r of golden ratio')
\]

and replace the Matlab command

\[
\text{for } i = 1:((m-1)\text{rand}(1)+1),
\]

by the Matlab command

\[
\text{for } i = 1:r,
\]

The resulting program named as goldenratiochisquaretest1randomnumbersource1 has been executed for all values of the number $r$ of rows of golden ratio in [1, 200] and is found from the chi-square test that the digits are always uniformly distributed, i.e. the test statistic $\chi^2$ is always less than the critical value of $\chi^2$ at 10% significance level. Certainly this is so at 5% as well as 1% significance level.

Matlab program for Monte Carlo integration from the golden ratio using 5-digit random numbers for $\sin x$.

The following Matlab program montecarlofromgoldenratio computes the value of $I = \int_0^{\pi/2} \sin x \, dx \approx \frac{\pi}{2} \frac{1}{N} \sum_{i=1}^{N} \sin(\frac{\pi}{2} t_i)$, where $t_i \ i = 1(1)N$ are $N$ random numbers in [0, 1] retrieved from consecutive five-digit blocks in golden ratio $\phi$. No pseudo-/quasi-random number generation is required thus saving computing time required for executing a generator loop thousands of times. Observe that the exact value of $I$ is $1$. 

Fig. 8. The bar chart for the distribution of the digits 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9 in the first 1400 digits (excluding the very first digit 1) of golden ratio $\phi$. The chart indicates that the digits are fairly uniformly distributed and thus implicitly prompts us to explore the possibility of using the digits of $\phi$ as a random number source. The program goldenratiochisquaretest1randomnumbersource selects randomly the first $N$ digits, where $N$ varies from one run of the program to the other.
Fig. 9. The value of \( I = \int_{0}^{\pi/2} \sin x \, dx = \frac{\pi}{2} \left( -1 \right) \int_{0}^{1} \sin \left( \frac{\pi}{2} t \right) \, dt \approx \frac{\pi}{2} \left( \frac{1}{N} \sum_{i=1}^{N} \sin \left( \frac{\pi}{2} t_i \right) \right) \) for \( N = 50(50)2000 \). The numbers 5, 10, 15, . . . correspond to \( N = 250, 500, 750, \ldots \) while the numbers 0.94, 0.95, 0.96 correspond to the computed values of \( I \). It can be seen that the best value of \( I = 0.992111893494446 \) obtained at \( N = 550 \). \( N \) = number of five-digit random numbers taken consecutively (systematically) from the digits of golden ratio expressed as blocks each of five digits. The exact value of \( I = 1.0 \).

\% golden ratio excluding the first two characters ”1.”
\% 10 000 digit golden ratio (without first two characters ”1.”) with a blank after each block of
\% 5 digits is partially entered in this program to conserve space. The required 10 000 digits
\% can be readily obtained using the the program FileConvertG in Appendix. The user needs to
\% enter the complete 10000 digits before running/executing this program clear all;
goldenratio=[61803 39887 49894 84820 45868 34365 63811 77203 09179 80576
28621 35448 62270 52604 62818 90244 97072 07204 18939 11374
84754 08807 53868 91752 12663 38622 23536 93179 31800 60766
72635 44333 89086 59593 95829 05638 32266 13199 28290 26788
. . .
62587 06154 33072 96370 38127 51517 04060 05057 59488 27238
56345 15639 05265 77104 26459 47604 05569 50959 84088 89037
62079 95663 88017 86185 59159 44111 72509 23132 79771 13803];%[a, b]=size(digits); %format compact; reshape(digits, 250, 4);
[a, b]=size(goldenratio); % format compact; reshape(goldenratio,200, 5)
% prand=digits/10^10; s=0;
reshape(goldenratio, 2000,1);
grrand=goldenratio/10^5;
disp(’ N Integration’);
for N=50:50:2000, s=0;
for i=1:N,s=s+sin(1.5707963268* grrand(i)); end;
format long g; Int(N)=1.5707963268*s/N;
disp([N Int(N)]);
end;
for j=1:40, Inte(j)=Int((j)*50); end;
plot(Inte)

The output results, viz. the integration value and the corresponding graph (Fig. 9) that we obtain by issuing the Matlab command

\texttt{>> montecarlofromgoldenratio}
are as given below. The accuracy of the integration can be readily improved by dividing the interval of integration into subintervals and then summing up all the Monte Carlo subinterval-integrations.

\[
\begin{array}{l}
N & \text{Integration} \\
50 & 0.962407406924503 \\
100 & 0.9429066505134 \\
150 & 0.94447580752357 \\
200 & 0.949644351564261 \\
\vdots & \ddots \\
1050 & 0.988643670772611 \\
1100 & 0.991022822960682 \\
1150 & 0.987114702808239 \\
1200 & 0.988176452938187 \\
\vdots & \ddots \\
1850 & 0.98203299662328 \\
1900 & 0.981171248520982 \\
1950 & 0.978752453412022 \\
2000 & 0.978389319086405 \\
\end{array}
\]

Matlab program for Monte Carlo integration from golden ratio using 5-digit random numbers for \( x^2 \) The following Matlab program \texttt{montecarlofromgoldenratioforxsq} computes the value of \( I = \int_0^1 x^2 dx = 8 \int_0^1 t^2 dt \approx 8(\frac{1}{N} \sum_{i=1}^N t_i^2) \), where \( t_i = 1(1)N \) are \( N \) random numbers in \([0, 1]\) retrieved from consecutive five-digit blocks in golden ratio \( \varphi \). No pseudo-/quasi-random number generation is required thus saving computing time is required for executing a generator loop thousands of times. Observe that the exact value of \( I \) is 8/3. The Matlab program \texttt{montecarlofromgoldenratioforxsq} is exactly the same as the foregoing one except the two lines of code, viz.

\[
\text{for } i=1:N, s=s+grrand(i)^2; \text{ end; } %\text{Integration of } x^2 \text{ between 0 and 2.}
\]

format long g; \texttt{Int(N)=8*s/N}; \texttt{\%Needed for the above integration.}

which replace the following two lines of the foregoing Matlab program:

\[
\text{for } i=1:N, s=s+\sin(1.5707963268*\text{grrand}(i)); \text{ end;}
\]

format long g; \texttt{Int(N)=1.5707963268*s/N;}

The output results, viz. the integration value and the corresponding graph (Fig. 10) that we obtain by issuing the Matlab command

\texttt{\texttt{>> montecarlofromgoldenratioforxsq}}

are as given below. The accuracy of the integration can be readily improved, as in the previous case, by dividing the interval of integration into subintervals and then summing up all the Monte Carlo subinterval integrations.

\[
\begin{array}{l}
N & \text{Integration} \\
50 & 2.925854649552 \\
100 & 2.617613779312 \\
150 & 2.59956824780267 \\
200 & 2.536713302112 \\
\vdots & \ddots \\
1300 & 2.61047808688862 \\
\end{array}
\]
The value of $I = \int_0^2 x^2 \, dx = \int_1^0 t^2 \, dt \approx \frac{1}{N} \sum_{i=1}^{N} t_i^2$ for $N = 50(50)2000$. The numbers 5, 10, 15, ... correspond to $N = 250, 500, 750, ...$ while the numbers 2.5, 2.6, 2.7, ... correspond to the computed values of $I$. It can be seen that the best value of $I = 2.6653610217667$ obtained at $N = 600$. $N =$ number of 5-digit random numbers taken consecutively (systematically) from the digits of golden ratio expressed as blocks each of five digits. The exact value of $I = \frac{8}{3}$.

Matlab program for Monte Carlo integration from the golden ratio using 10-digit random numbers. The following Matlab program `montecarlofromgoldenratio2` computes the value of $I = \int_0^{\pi/2} \sin x \, dx = \int_0^1 \sin(\frac{\pi}{2} t) \, dt \approx \frac{\pi}{2} \sum_{i=1}^{N} \sin(\frac{\pi}{2} t_i)$, where $t_i \, i = 1(1)N$ are $N$ random numbers in [0, 1] retrieved from consecutive ten-digit blocks in golden ratio $\phi$. No pseudo-/quasi-random number generation is required thus saving computing time required for executing a generator loop thousands of time. Observe that the exact value of $I$ is $\frac{\pi}{2}$.

```matlab
%first two characters ”1.” are omitted from golden ratio.
%10000 digit golden ratio (without first two characters ”1.”) with a blank after each block of 10 digits is
%partially entered in this program to conserve space. The required 10000 digits can be
%readily obtained using the the program FileConvertG in Appendix. The user needs to
%enter the complete 10000 digits before running/executing this program

goldenratio=[6180339887 4989484820 4586834365 6381177203 0917980576 2862135448 6227052604 6281890244 9707207204 1893911374 8475408807 5386891752 1266338622 2353693179 3180060766 7263544333 8908659593 9582905638 3226613199 2829026788 6258706154 3307296370 3812751517 0406005057 5948827238 5634515639 0526577104 2645947604 0556950959 8408889037 6207995663 8801786185 5915944111 7250923132 7977113803];
m=200; n=5; for i=1:m, j=1:n, k=(i-1)*n+j; gronerow(k)=gr(i,j);
```
Fig. 11. The value of $I = \int_{0}^{\pi/2} \sin x \, dx = \frac{\pi}{2} \int_{0}^{1} \sin (\pi t) \, dt \approx \frac{\pi}{2} \left( \frac{1}{N} \sum_{i=1}^{N} \sin (\pi t_i) \right)$ for $N = 50(50)1000$. The numbers $2, 4, 6, \ldots$ correspond to $N = 100, 200, 300, \ldots$ while the numbers $0.94, 0.95, 0.96$ correspond to the computed values of $I$. It can be seen that the best value of $I = 0.998813868956653$ (better than the value of $I$ with 5-digit random numbers as it should be) obtained at $N = 550$. $N =$ number of 10-digit random numbers taken consecutively (systematically) from the digits of the golden ratio written as consecutive blocks each of 10 digits. The exact value of $I = 1$.

```
% end; disp(gronerow)
% [a, b]=size(digits); % format compact; reshape(pdigits, 250, 4);
[a,b]=size(goldenratio); % format compact; reshape(goldenratio,200, 5)
% prand=digits/10^10; s=0;
reshape(goldenratio, 1000,1);
grrand=goldenratio/10^10;
disp( ‘N Integration’);
for N=50:50:1000, s=0;
for i=1:N,s=s+sin(1.5707963268*grrand(i));
end;
format long g; Int(N)=1.5707963268*s/N;
disp([N Int(N)]);
end;
for j=1:20, Inte(j)=Int((j)*50); end;
plot(Inte)
```

The output results, viz. the integration value and the corresponding graph that we obtain by issuing the Matlab command $>>$ montecarlofromgoldenratio2 are as given below and in Fig. 11, respectively. The accuracy of the integration can be readily improved, as in previous cases, by dividing the interval of integration into subintervals and then summing up all the Monte Carlo subinterval integrations.

<table>
<thead>
<tr>
<th>$N$</th>
<th>Integration</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>0.962413870508596</td>
</tr>
<tr>
<td>100</td>
<td>0.942913899551367</td>
</tr>
<tr>
<td>150</td>
<td>0.944483364078763</td>
</tr>
<tr>
<td>200</td>
<td>0.949652425483289</td>
</tr>
<tr>
<td>850</td>
<td>0.978375281416709</td>
</tr>
<tr>
<td>900</td>
<td>0.978503139566131</td>
</tr>
<tr>
<td>950</td>
<td>0.975624145854393</td>
</tr>
<tr>
<td>1000</td>
<td>0.981956986757435</td>
</tr>
</tbody>
</table>
In the foregoing Matlab programs, the same 10 000 digit golden ratio (excluding the first two characters “1.”) may appear superfluous or against conservation of space. Since there is no immediate easy way to write the golden ratio with consecutive 10 000 digits (without any blank between any two digits anywhere) as blocks of 5 or 10 digits in Matlab, both the foregoing programs are very convenient for users of Monte Carlo integration. This could be readily done just by copying and pasting one of the two programs depending on the accuracy requirement and then replacing the current function $\sin x$ or $x^2$ by the given function to be integrated. The Monte Carlo method here retrieves readily systematically the random numbers from the stored digits of the golden ratio without use of any random number generators. We may store 100 000 or more digits of golden ratio instead of just 10 000 digits stored/shown here.

We may, however, choose random numbers each of fixed block of digits from golden ratio in a non-consecutive uniform manner or even randomly. For better accuracy larger blocks need to be chosen.

5. **$\varphi$ in nature, artifacts, and architecture**

The Greek mathematicians Pythagoras (about 582 BC–507 BC) and Euclid (about 330 BC–275 BC), the Italian mathematician Fibonacci (about 1175–1250), also known as Leonardo of Pisa, the German Lutheran mathematician J. Kepler (1571–1630), the British mathematical physicist R. Penrose (1931) are just a few names over the past 25 centuries, who have spent countless hours over this simple yet amazing number, the golden ratio and its properties. Not only mathematicians but also musicians, psychologists, architects, historians, biologists, artists, and mystics have pondered over the omnipresence of this number. Although it is not as well known as $\pi$, the golden ratio has stimulated the thought process of intellectuals of all disciplines like possibly no other number in mathematics. For a brief account as well as further links, refer [45]. The Italian mathematician Luca Pacioli (about 1445–1517) may be credited with starting the modern history of the golden ratio $\varphi$ in around 1509. We just mention below some of the numerous connections of $\varphi$ in nature, artifacts and architecture.

The golden ratio appears in the geometry of regular pentagons and pentagons. Phidias built, in 5th century BC, the Parthenon (a temple of Athena) statues that appear to embody the golden ratio. Plato (427 BC–347 BC) proposed five regular solids – tetrahedron, cube, octahedron, dodecahedron, and icosahedron – some of which have golden ratio connection. For example, an icosahedron (polyhedron with 20 faces) with edge length 2 in three dimensional Cartesian coordinates has the 12 vertices $(0, \pm 1, \pm \varphi)$, $(\pm 1, \pm \varphi, 0)$, $(\pm \varphi, 0, \pm 1)$, where $\varphi$ is the golden ratio [46]. The Swiss naturalist C. Bonnet (1720–1793) discovered that there were two successive Fibonacci series in the spiral **phyllotaxy** (arrangement of the leaves on the shoot) of a plant going clockwise and anticlockwise. R. Penrose (b. 1931) discovered a symmetrical pattern which uses golden ratio in *aperiodic tiling* (tiling which never repeats itself) resulting in new discoveries on quasicrystals (aperiodic structures that are capable of producing diffraction).

During the 14–16th century, the aesthetics (a branch of philosophy of art known as axiology or value theory) of golden ratio developed. Consequently, book designers, artists and architects were encouraged to adopt the golden ratio in the dimensional relationships of their works yielding pleasing harmonious proportions. The golden ratio is sometimes used in modern artifacts such as stairs, buildings and woodworks.

As to architecture, the front structure of the Parthenon (temple) depicts golden rectangles in its proportions. It is probably not that the architect consciously made the design keeping golden rectangles in mind. It is possibly because of other consideration such as the stability and the aesthetic sense. Archeologists have found that the Acropolis (edge of a high city) of Athens including the Parthenon that several of its geometrical proportions are golden ratio approximately. A dimensional analysis of the Mosque of Uqba (oldest mosque located in Kairoun, Tunisia and built in 670 AD) reveals that the designers had consistently applied golden ratio throughout the design.

As to art/painting, the canvas of Sir Lawrence Alma-Tadema (b. 1836), a fine Victorian Dutch painter “The Roses of Heliogabalus (1888)” has the dimensions 213 cm × 132 cm — an almost perfect golden rectangle. In an illustration, Leonardo Da Vinci probably consciously applied golden ratio to the human face. Some think that in his creation of Mona Lisa, he employed the golden ratio. Piet Cornelis Mondrian (1872–1944), a Dutch painter employed the golden section in his geometrical paintings. However, a dimensional/geometrical study on 565 works of art of various eminent painters performed in 1999 inferred statistically that the mean ratio of the two sides of their paintings is 1.34 with a minimum value of 1.04 and a maximum value of 1.46.

In nature, the rabbit population seems to grow in such a way that we tend to get a feel that there is a similarity of the rabbit sequence with the Fibonacci sequence which, in the limit, produces the golden ratio. However, such an observation is crude and not possibly very enchanting.
6. Conclusions

Enormous yet non-exhaustive information is available on golden ratio. There has been a highly impressive wealth of information recorded in the internet as well as in books [1–4,6,9–26] on golden ratio and related materials. Yet the newer occurrences of the golden ratio or its connections are being discovered. It appears impossible to claim that the information on golden ratio/connected materials is complete and no more discovery (or no more different way of viewing) on this subject in future is possible. In spite of its (golden ratio’s) tendency to pop up in many many places, it is much less known than \( \pi \).

No exact representation of \( \varphi \) in a number system — The numbers such as the golden ratio \( \varphi = (1 + \sqrt{5})/2 \) and the ratio (the circumference divided by the diameter of any circle = area of the unit circle) \( \pi = 2 \int_{-1}^{1} \sqrt{1-x^2} \, dx \) have exact real existence. Can we capture this exactness in finite number of digits in any number system — The golden ratio \( \varphi \), an algebraic number, cannot be expressed exactly in any so far known number system such as the fixed radix (base), the variable radix and the negative radix systems [27–29]. In any number system, it will have an infinite number of digits. It can be seen that the decimal rational number \( 1/3 = 0.333333 \ldots \) has no exact representation in radix 10 (decimal) number system while it has exact representation, viz., 0.1 in radix 3 number system in which the three symbols conventionally used are 0, 1 and 2. Although it may/may not be impossible to devise a number system in which \( \varphi \) can be exactly represented in a finite number of digits, such a number system could have difficulty or face impossibility to represent any conventional number which we are familiar with. This is probably because of the randomness of its infinity of digits not only in a decimal system but also in other number systems known to us. However, possibly any decimal irrational number cannot be exactly represented/captured either in any existing number system with fixed/variable/negative radix or in any yet-to-be discovered number system. Although such a number does have real existence and nature has its exact representation, our so far known number systems fail to exactly represent them.

Numerical name versus alphabetical name. We the human beings/the living beings are psychologically much more comfortable with easily pronounceable (and possibly sometimes having some meaning) name. This name is almost always made up of alphabetical characters (letters) in any language and not of numbers in a number system (say, decimal number system). Such a non-numerical name is relatively easy to remember, pronounce and spell correctly. We have not known, in general, names made up of digits in, say, the decimal system. Often/sometimes, however, a prisoner may be given a fixed number and subsequently known/called by that number rather than by his original name at least inside the prison. We have read the famous story named The Bishop’s Candlesticks in which the convict Jean Valjean (a peasant) who, after 19 years of imprisonment for stealing food for his starving family, was released. He was called/known in the prison by the number 24 601 assigned to him and not by his original name [19]. He later became a force for good in the world although it became too difficult for him to escape the past. So far as \( \varphi \) is concerned, we are more comfortable and probably need less effort/energy to call it by the alphabetical name(s) such as the golden ratio, the golden number and the divine proportion rather than by the non-exact decimal number 1.6180339887498948482045868343656381177203091798058 (up to 50 digits). It may not be out of place to mention that there are some numbers such as the decimal number 1729 = 7 \times 13 \times 29 which are immediately recognized as interesting (not a dull) numbers with their important character and nature by an exceptional human mind such as the Indian mathematician Srinivasa Ramanujan (1887–1920) [30]. Ramanujan recognized readily this taxi cab number 1729 as the smallest number expressible as the sum of two cubes in two different ways, viz., \( 1^3 + 12^3 = 9^3 + 10^3 \) [31]. No alphabetical name has been given to the number 1729 which is best known by itself. To Ramanujan, each number is said to be his personal friend.

Golden ratio versus \( \pi \) as random sequence source. Considerable amount of work has been done on the distribution of digits of \( \pi \) as well as on computing a billion or more digits of \( \pi \) [32–38]. Are the infinite digits of the algebraic number golden ratio \( \varphi \) equally random as those of the transcendental number \( \pi \)? Are the digits of \( \pi \), if used as a random sequence source, produce better results in Monte Carlo integration than what the digits of \( \varphi \) have done? Are the digits of \( \pi \) statistically equally uniformly distributed as those of golden ratio? If we have to choose between \( \pi \) and \( \varphi \) for a randomized algorithm/heuristic, then which one we should choose? These questions could be probably satisfactorily answered considering a large (size \( \geq 30 \)) sample following the procedures in [7,8]. We are yet to make a detailed study of these, but our preliminary investigation suggests that the digits of \( \pi \) appear to perform better than those of \( \varphi \) so far as the quality of the Monte Carlo integration is concerned. The detailed study not only of \( \pi \) and \( \varphi \), but also of other transcendental and algebraic numbers in terms of their utility in real world applications, specifically in NP-hard problems [39,40] such as the travelling salesman problem (TSP) will appear elsewhere. Consider those algebraic and
transcendental numbers having non-repetitive/non-recursive infinite digits. How do these numbers compare as random sequence sources with popular quasi- and pseudo-random number generators [7,8,41,42] in terms of the quality of the output of, say, the ant algorithm or the genetic algorithm or the simulated annealing for TSP? — A detailed study with respect to each of the algorithms will be made. We have already seen statistically that quasi-random sequences are better suited for s-dimensional domain/space scanning [7,8,41,42]. The pseudo-random sequences, on the other hand, are the bases for many cryptographic applications in which easy deciphering without the knowledge of the key(s) is unacceptable. Quasi-random sequences which focus on maintaining uniformity (thus reducing randomness) are less desirable than pseudo-random sequences in these applications.

**Very many ways of looking at the golden ratio: Do we really create these ways?** There are indeed possibly unlimited ways to view/compute the golden ratio. It is not impossible to find a relation/link among these ways or to derive one way from the other. Also, it is probably impossible to show that so far all the ways have been explored. Under these circumstances the question “How original is the new contribution on $\phi$?” is subjective. Some may consider/derive a possible relationship of the new contribution and feel that it is nothing new while others may look at it in a different angle and discover its beauty in their own way and enjoy the new contribution. So far as the computation of $\phi$ is concerned, two or more different algorithms will be linked at least in terms of the goal, viz. the output for $\phi$. Physically the steps could be different, but the cost of computation (computational/time complexity) as well as the quality of the result/output (relative error) could be the same or different. A fast (i.e. polynomial-time) algorithm to compute $\phi$ or a solution of any other problem with lower order of complexity and/or better quality of the result than that of a best one existing is definitely more desirable and is given a place in literature. In the absolute sense, however, we believe that there is no creation of knowledge or matter or anything – non-existing – that is possible. Things (material or non-material) which do not exist cannot be created. Creating something out of nothing has never been experienced/shown by any known human being nor is it non-violation of laws of nature (both physical and non-physical/spiritual) that we know of. We do see creation of something out of nothing **only in magic!** We only discover/find ways/knowledge out of ever existing knowledge/ways which is unfathomable. It is possibly not out of place to quote Swami Vivekananda: “All knowledge that the world has ever received comes from the mind; the infinite library of the universe is in your own mind” [43]. Thus whatever has been presented here is our own way of viewing $\phi$. We are, like any other contributor, not exhaustive in dealing with $\phi$, nor can anybody be exhaustive now or in future. We believe that there will be readers who will find our ways interesting. Probably these will induce inquisitiveness and further innovation particularly with high level user-friendly programming languages such as the Matlab and the Maple. We are confident that many more newer beauties and utilities of $\phi$ will be discovered by critical minds of scientists and mathematicians for others to enjoy/benefit.

*The motive.* The purpose of this article is to provide the reader with information so as to prompt his/her critical mind to explore latent beauty as well as the utility of $\phi$ in a real-world environment. This article has also the potential to discover in several other numbers such as the exceedingly important transcendental number $\pi$ their real world application in addition to mathematical beauty. Specifically, which of $\phi$, $\pi$ and any other number (algebraic/transcendental) having infinite number of non-recurrent digits is most ideal as a quasi- or a pseudo-random number source is certainly an important study. To solve many challenging problems such as the NP-hard [39,40] TSP which has tremendous real-world utility, we do need excellent random numbers in large/huge quantities. Such excellent random numbers should be readily (practically requiring no significant computing resource) available. The idea of using a randomized algorithm/evolutionary approach/ genetic algorithm/ heuristic algorithm is mainly to solve NP-hard (intractable) problems in polynomial-time. A deterministic polynomial-time algorithm (which needs no random numbers) for such problems is yet unknown [40]. Also, the article, we believe, has the potential to encourage researchers to discover many other as yet unexplored properties/characters in these numbers.

**Superimposition of Matlab rand on choice of random numbers from golden ratio.** It is possible to randomly select blocks of digits from golden ratio and use them for any randomized/heuristic algorithm. In such a situation, the number of the use of “rand” will be relatively small. One may study if such a combination/superimposition really does help to procure random numbers out of the golden ratio or $\pi$ or any other transcendental/algebraic number. The blocks of, say, 10 digits sieved out from $\phi$ sequentially/systematically possibly do solve the requirement of a random sequence equally well (without the use of rand).

**Digits of golden ratio: How many computed and any pattern—** In December 1996, ten million decimal digits of the golden ratio have been computed using Maple needing 29 min 16 sec on an SGI R10000 195 MHz computer [24]. In May 2000, 1.5 billion decimal digits of the golden ratio have been computed in less than three hours of
computation [24]. In December 2005, 25,000 binary digits of the golden ratio have been computed using Maple [25]. No specific pattern has been so far discovered in the digits of the golden ratio in any number system. This fact has encouraged us to use, like that in the digits of \( \pi \), the consecutive/contiguous blocks of digits of the golden ratio as a pseudo-random sequence and use this sequence for multiple Monte Carlo integration and other randomized algorithms for many real-world problems. The first 500 bits (binary digits) of the golden ratio excluding the very first integral digit “1” is presented below inside the Matlab program named \texttt{goldenratiochisquaretest1randomnumbersource1binary} for the reader to get a feel about the distribution of 0s and 1s in these digits.

```matlab
r=input('Enter a value in [1, 10] of the no. of rows r of binary golden ratio ’)
zero=0; one=0;
for i=1:r
    for j=1:n,
        q=goldenratio(i, j);
        if q==0, zero=zero+1; else q==1, one=one+1;
    end;
end;
obsfreq=[zero one], e=ones(1, 2);
totaldigits=zero+one;
expfreq=(totaldigits/2)*e,
teststatisticchisquare=sum((expfreq-obsfreq).^2/expfreq),
criticalvalueofchisquare=2.706, %at 10% significance level with one degree of freedom
if teststatisticchisquare<criticalvalueofchisquare,
'Accept H0, i.e. selected consecutive digits are uniformly distributed at 10% significance level’, else
'Reject H0, i.e. selected consecutive digits are not uniformly distributed at 10% significance level’
end;
x=[0 1]; bar(x, obsfreq)
```

If we consider first 200 bits (binary digits) of the golden ratio, we obtain, by issuing the Matlab command

```matlab
>> goldenratiochisquaretest1randomnumbersource1binary
```

and setting \( r = 4 \), the observed frequencies 99 0s and 101 1s. The expected frequencies are 100 0s and 100 1s. The test statistic \( \chi^2 = 0.02 \) while the critical value of \( \chi^2 \) at 10% significance level with one degree of freedom is 2.706. So we accept the null hypothesis \( H_0 \). That is, the bits 0 and 1 are uniformly distributed. The corresponding bar chart is shown in (Fig. 12).

If we now consider the first 500 bits of the golden ratio, we obtain, by issuing the Matlab command

```matlab
>> goldenratiochisquaretest1randomnumbersource1binary
```

and setting \( r = 10 \), the observed frequencies 264 0s and 236 1s. The expected frequencies are 250 0s and 250 1s. The test statistic \( \chi^2 = 1.568 \) while the critical value of \( \chi^2 \) at 10% significance level with 1 degree of freedom is 2.706. So we accept the null hypothesis \( H_0 \). That is, the bits 0 and 1 are uniformly distributed. The corresponding bar chart is as shown in (Fig. 13).
The character of the golden ratio in any number system such as that with any positive integral radix (base), negative radix, and variable radix \([27–29]\) possibly remains invariant. The statistical experiment done by us demonstrates this fact for both radix 2 and radix 10.

**Computation of \(\phi\) to an arbitrary number of digits by Matlab command \texttt{vpa}**. Thousands of digits of \(\phi\) are already available in the internet. One may possibly use these electronic digits. This could be cumbersome as well as sometimes could involve errors. Alternatively, the Matlab variable precision arithmetic command \texttt{vpa} is very conveniently used to compute \(\phi\) to an arbitrary number of decimal digits, say 50 000 digits. These digits are produced errorless in one line without any blank between any two digits or any two blocks of digits. This is an extremely useful command not only to compute \(\phi\) but also to compute numerous algebraic and transcendental numbers to an arbitrary number of digits subject, however, to the software, hardware, and possibly sometimes time limitations. This single three-letter command described in Section \(3\) is indeed extremely useful for the reader to explore \(\phi\) as well as many other numbers such as \(\pi\) to a much deeper extent using a widely available over a billion FLOPS (floating-point operations per second) computing device along with a good available Matlab version.

**Inserting a required blank after each fixed block of digits**. To conserve space we have omitted most of the digits in several foregoing Matlab programs, which must be entered before running the program. The general Matlab program File ConvertG along with the Matlab command \texttt{vpa} will readily produce the required digits with required blanks among blocks of digits. These programs will obviate the need for the reader to learn involved Matlab file manipulation program(s). This will definitely save lots of time for the serious reader/user and possibly eliminate frustration as programming may not be his/her cup of tea.

**Beauty of golden ratio: A subjective feeling of an individual**. Whenever a person visualizes scenery or goes through a sequence of digits or a poem or listens to a music, he/she tries to comprehend/find out similarities/exactness/a rhythm/a pattern/a non-quantified beauty in it consciously or unconsciously or subconsciously so that he/she can appreciate it in his/her own way. When he/she reads a modern poem, he/she tries to extract
the significance/message/beauty from it in his/her own way. Once he/she does so, he/she will have the satisfaction and enjoyment of discovering the beauty. This enjoyment is subjective and varies from person to person. If he/she is unable to extract the beauty/message or to comprehend it, the poem will appear to him/her as beautyless/messageless while to another person who has grasped the same poem in his/her own way might see beauty in it or might find it not so enchanting. It is so with a music or with scenery. However, there could be several minds with critical analysing power who could gauge in their own way the beauty to a varied extent. Although an analogy of a poem with the golden ratio is not proper in any other respect, it is not very much out of place or improper when we talk about beauty with respect to both. When we are able to comprehend or correlate or link the golden ratio with something in nature/artifacts, we derive enjoyment of our own discovery. It is impossible to say that all the discoveries of the character of the golden ratio and its connection to nature, art, architecture, science and beyond have been made over centuries by numerous analytical/scrupulous minds and no more (future) discoveries are possible. True it is that future discoveries may be linked/correlated with the past ones without much difficulty, yet these could/would provide sources of fresh enjoyment/beauty to numerous minds who have interest in golden ratio.

References cited from internet are appropriately checked. We, like any other reader, have rather easily detected sometimes some errors in some of the internet literature. It is widely felt that the information that we get from the internet are not that reliable as that we get through a journal paper/a book as it is not known to have passed through a responsible/reasonable review process. However, a serious reader can sieve out the useful information from the internet literature using logic and own judgement and cross-verifying with the existing literature. It would be wrong to suggest that any information posted in the internet is useless/wrong. We have carefully after proper examination/evaluation sieved out the useful information from the cited websites.

Relation between $\pi$ and $\phi$ and their easy-to-remember values. It is interesting to note that $\pi \sqrt{\phi}$ is approximately 4. Using the Matlab command

$$\text{vpa('pirootphi=2*asin(1)*sqrt((sqrt(5)+1)/2)',20)}$$

we obtain the value up to 20 digits as $\text{pirootphi} = 3.9961675861352626668$. However, using bisection/trial-and-error, a better integral relationship will be obtained if the power of $\phi$ is 0.501992 instead of 0.5. The required Matlab command would be then

$$\text{vpa('pirootphi=2*asin(1)*((sqrt(5)+1)/2)\^0.501992',20)}$$

that produces the value $\text{pirootphi} = 4.0000000448593202880$ (up to 20 digits). It is not our objective to determine the power of $\phi$ to obtain an integral value very accurately as this is always possible for any two numbers. Our motive is to find out a simple easily memorizable rational approximation for $\phi$ for real world/school rough calculation just as we had used $22/7$ for $\pi$. We find this approximation as $8/5$ for $\phi$. It is interesting to note that $8/5$ km is approximately 1 mile.

Golden ratio connection: Fibonacci numbers and Pascal triangle in ancient India. Pingala, an Indian mathematician of 5th ca. century B.C., the younger brother of Panini, the great grammarian, who presented the first known description of a binary number system [47], gave the basic ideas of Fibonacci numbers [48]. The 10th century mathematician Halayudha’s commentary includes a presentation of the Pascal triangle [49] called meru-prastaara [50]. For further details, refer [51–54].

High speed progression of Fibonacci numbers. Fibonacci numbers progress with a discrete starting number analogous to the continuous physical quantity called velocity ($v$) with discrete (possibly non-uniform) acceleration ($a$). A Fibonacci number seems analogous to the distance $s = vt + at^2$. This needs exploration.

Acknowledgement

The authors wish to thank Gholam Ali Shaykhian for writing the general Matlab program FileConvertG that introduces a single blank after each block of $k$ digits in a slightly different form.

Appendix

The following general Matlab program FileConvertG takes as input a.txt file and produces as output a file with one blank after each block of $k$ digits. Although it appears that inserting blanks is manually a simple task, it is indeed significantly involved in Matlab programming. There seems to be no one Matlab command by which we can achieve the file with desired blanks. For a large file, say a data file consisting of thousands of digits available in
an electronic form, the following program inserts the blanks appropriately and correctly in the file thus completely obviating possible human errors and the need for enormous manual labour. A need for inserting a blank after each block of consecutive \(k\) digits is to achieve a random sequence with each random number consisting of \(k\) digits. This will enable desired precision in the computation involving a randomized algorithm. Smaller the \(k\) is, the higher is the possible error and vice versa.

% Reformatting.txt file: Permits inserting a blank in the file after each block of \(k\) digits.
function f = FileConvertG()
cle;
% show.txt files
system('dir *.txt');
% system('dir *.asv');
% system('dir *.*');
disp(' Press Enter key to continue . . . .
pause;
disp(sprintf('

n'));
inputFileName=input('Enter Input File Name: ', 's');
%Input file name will have extension.txt
disp(sprintf('

n
n'));
outputFileName=input('Enter Output File Name: ', 's');
disp(sprintf('

n'));
numberOfColumns=input('Enter Number of columns for display Matrix: ');
disp(sprintf('

n
n'));
numberOfKdigits=input('Enter Number of Digits in each Element: ');

fid1 = fopen(inputFileName);
fid2 = fopen(outputFileName,'w');
St = fscanf(fid1,'%s');
loopCount =length(St)/ numberOfKdigits;
S=length(St);
initial=1;
final=numberOfKdigits;
Columns = 0;
Line = "
for i=1:numberOfKdigits:S
    Temp=St(1,initial:final);
    initial = initial+numberOfKdigits;
    final = final+numberOfKdigits;
    if final>S % last number is not \(K\)-digits
        final = S;
    end
    Columns = Columns +1;
    Line = strcat(Line, sprintf(' %s', Temp));
if Columns == numberOfColumns
    disp(sprintf(' %s\n',Line));
    fprintf(fid2, '%s\n',Line);
    Columns = 0; % Reset column counter
    Line="
end
References

[21] http://www.joyofpi.com/pi.html, (for first one million consecutive (with one blank after every ten digits excluding the first two characters 3.) digits of π).
[33] http://www.geom.uiuc.edu/~huberty/math5337/groupe/digits.html, 1997 (Includes 100000 contiguous digits of π (with no blank in any line)).
[34] http://www.angio.net/pi/piqtery. (The pi-search page).


[38] http://www.cecm.sfu.ca/organics/papers/borein/index.html. (Contains information for Ref. [37]).


