

# Extracting Hexadecimal Digits of $\pi$

*William Malone*



**William Malone** is currently a senior Mathematics major at Ball State University. Upon graduation he plans to pursue graduate studies starting in the fall of 2005. The content of this paper was a result of a project he did with Dr. Hanspeter Fischer in the fall of 2004.

How would you go about finding, say, the billionth digit of  $\pi$ ? There are many algorithms, that can calculate the billionth digit of  $\pi$  in various bases within a reasonable amount of time on a powerful computer system. However, they usually rely on calculating *all* the digits of  $\pi$  less than and including one billion. This necessarily involves arithmetic of huge numbers, which is typically implemented by means of Fast Fourier Transforms. There are also very elegant new algorithms that allow us to compute many digits of  $\pi$  on a personal computer. The software package *Mathematica*, for example, uses a fast converging series technique, developed by the Chudnovsky brothers in 1987, to compute all the decimal digits of  $\pi$  less than a given number [3]. However, it is not feasible to go beyond 10 million decimal digits with this method on a personal computer, because of speed and storage limitations.

In 1997, David Bailey, Peter Borwein and Simon Plouffe discovered a formula for  $\pi$ , which allows us to extract any given hexadecimal digit of  $\pi$  by means of a strikingly simple method, without ever computing the digits leading up to it, in essentially linear time and logarithmic storage space [1]. Indeed, it could be programmed on a hand-held calculator. While 16 is a very natural base for computers, its occurrence in this context is rather coincidental, as we shall see below. Recall that, in base 16, the familiar decimal expansion

$$\begin{aligned}\pi &= 3.141592\dots \\ &= 3 \cdot 10^0 + 1 \cdot 10^{-1} + 4 \cdot 10^{-2} + 1 \cdot 10^{-3} + 5 \cdot 10^{-4} + 9 \cdot 10^{-5} + 2 \cdot 10^{-6} \dots\end{aligned}$$

becomes

$$\begin{aligned}\pi &= 3.243F6A\dots \\ &= 3 \cdot 16^0 + 2 \cdot 16^{-1} + 4 \cdot 16^{-2} + 3 \cdot 16^{-3} + 15 \cdot 16^{-4} + 6 \cdot 16^{-5} + 10 \cdot 16^{-6} \dots\end{aligned}$$

Although this so-called *hexadecimal* expansion is slightly more compact, if we printed the first one billion hexadecimal digits of  $\pi$  on one strip of paper, it would still be over 1,000 miles long.

In this note, we will describe the method of Bailey-Borwein-Plouffe and use it to compute a short consecutive sequence of hexadecimal digits of  $\pi$  well beyond the one billion hexadecimal digit mark. The algorithm is based on the following series representation of  $\pi$ :

**Theorem (Bailey-Borwein-Plouffe).**

$$\pi = \sum_{k=0}^{\infty} \frac{1}{16^k} \left( \frac{4}{8k+1} - \frac{2}{8k+4} - \frac{1}{8k+5} - \frac{1}{8k+6} \right)$$

**Proof.** We begin with an integral whose value will be shown to equal  $\pi$ . Consider

$$\int_0^{\frac{1}{\sqrt{2}}} \frac{4\sqrt{2} - 8x^3 - 4\sqrt{2}x^4 - 8x^5}{1 - x^8} dx.$$

Upon substituting  $y = \sqrt{2}x$  we obtain

$$\int_0^1 \frac{4\sqrt{2} - \frac{8y^3}{(\sqrt{2})^3} - \frac{4\sqrt{2}y^4}{(\sqrt{2})^4} - \frac{8y^5}{(\sqrt{2})^5}}{1 - \frac{y^8}{(\sqrt{2})^8}} \frac{dy}{\sqrt{2}},$$

which in factored form reads

$$\int_0^1 \frac{16(y-1)(y^2+4)(y^2+2y+2)}{(y^2+2y+2)(y^2-2y+2)(y-\sqrt{2})(y+\sqrt{2})(y^2+2)} dy.$$

Reducing the fraction we can simplify this to

$$\int_0^1 \frac{16y-16}{(y-\sqrt{2})(y+\sqrt{2})(y^2-2y+2)} dy.$$

Finally, we use the method of partial fractions:

$$\int_0^1 \frac{16y-16}{(y-\sqrt{2})(y+\sqrt{2})(y^2-2y+2)} dy = \int_0^1 \frac{2}{y-\sqrt{2}} + \frac{2}{y+\sqrt{2}} - \frac{4(y-2)}{y^2-2y+2} dy$$

(Integration of the first two terms is straightforward and for the third term we use the trigonometric substitution  $y-1 = \tan \theta$ .)

$$\begin{aligned} &= \left[ 2\ln(\sqrt{2}-y) \right]_{y=0}^{y=1} + \left[ 2\ln(\sqrt{2}+y) \right]_{y=0}^{y=1} - 4 \int_{-\frac{\pi}{4}}^0 \frac{(\tan \theta - 1)}{\tan^2 \theta + 1} \sec^2 \theta d\theta \\ &= 2\ln(\sqrt{2}-1) - 2\ln(\sqrt{2}) + 2\ln(\sqrt{2}+1) - 2\ln(\sqrt{2}) - 4 \int_{-\frac{\pi}{4}}^0 \tan \theta - 1 d\theta \\ &= 2\ln((\sqrt{2}-1)(\sqrt{2}+1)) - 4\ln(\sqrt{2}) - 4 \left[ -\ln(\cos \theta) - \theta \right]_{\theta=-\frac{\pi}{4}}^{\theta=0} \\ &= 2\ln(2-1) - 4\ln(\sqrt{2}) - 4 \left( -\ln 1 - 0 + \ln \frac{1}{\sqrt{2}} - \frac{\pi}{4} \right) = \pi. \end{aligned}$$

It remains to show that this integral equals the sum as claimed. Notice that

$$\frac{1}{1-x^8} = \sum_{k=0}^{\infty} x^{8k} \quad (\text{for } 0 \leq x \leq \frac{1}{\sqrt{2}}),$$

so that

$$\int_0^{\frac{1}{\sqrt{2}}} \frac{x^{p-1}}{1-x^8} dx = \int_0^{\frac{1}{\sqrt{2}}} \sum_{k=0}^{\infty} x^{p-1} x^{8k} dx \quad (\text{for any } p \geq 1).$$

Because the sum is a geometric series and hence uniformly convergent, we can pull the summation outside the integral and evaluate it as follows:

$$\sum_{k=0}^{\infty} \int_0^{\frac{1}{\sqrt{2}}} x^{p-1} x^{8k} dx = \sum_{k=0}^{\infty} \left[ \frac{x^{8k+p}}{8k+p} \right]_{x=0}^{x=\frac{1}{\sqrt{2}}} = \frac{1}{\sqrt{2}^p} \sum_{k=0}^{\infty} \frac{1}{16^k(8k+p)}.$$

Doing this for each of the four terms, we obtain

$$\begin{aligned} \int_0^{\frac{1}{\sqrt{2}}} \frac{4\sqrt{2} - 8x^3 - 4\sqrt{2}x^4 - 8x^5}{1-x^8} dx \\ = \sum_{k=0}^{\infty} \frac{1}{16^k} \left( \frac{4}{8k+1} - \frac{2}{8k+4} - \frac{1}{8k+5} - \frac{1}{8k+6} \right). \quad \square \end{aligned}$$

**The Algorithm.** Let  $n$  be a fixed (large) positive integer. By the above Theorem,

$$16^n \pi = \sum_{k=0}^{\infty} 16^{n-k} \left( \frac{4}{8k+1} - \frac{2}{8k+4} - \frac{1}{8k+5} - \frac{1}{8k+6} \right).$$

If we look at the fractional part of the left hand side, it yields the hexadecimal expansion of  $\pi$  starting at position  $n+1$ . So, the same is true of the sum on the right:

$$(16^n \pi) \bmod 1 = \left( \sum_{k=0}^{\infty} 16^{n-k} \left( \frac{4}{8k+1} - \frac{2}{8k+4} - \frac{1}{8k+5} - \frac{1}{8k+6} \right) \right) \bmod 1$$

We can break this sum apart into terms where 16 is raised to a non-negative power and terms where 16 is raised to a negative power. In the second sum, we need to compute only the first twenty, or so, terms because it converges very rapidly and we are interested only in a small window of hexadecimal digits:

$$\begin{aligned} (16^n \pi) \bmod 1 \approx & \left( \sum_{k=0}^n 16^{n-k} \left( \frac{4}{8k+1} - \frac{2}{8k+4} - \frac{1}{8k+5} - \frac{1}{8k+6} \right) \right) \bmod 1 \\ & + \left( \sum_{k=n+1}^{n+20} 16^{n-k} \left( \frac{4}{8k+1} - \frac{2}{8k+4} - \frac{1}{8k+5} - \frac{1}{8k+6} \right) \right) \bmod 1. \end{aligned}$$

The point of the algorithm is that we can drastically simplify computations in the first sum by using modular arithmetic. Since all we are interested in is the fractional part of this expression, we get away with replacing each numerator by its residue modulo the denominator. For example,

$$\sum_{k=0}^n 16^{n-k} \frac{4}{8k+1} \bmod 1 = 4 \sum_{k=0}^n \frac{16^{n-k} \bmod (8k+1)}{8k+1} \bmod 1.$$

This allows us to calculate  $16^m$  by modular exponentiation, accounting for the impressive time savings. Here is how: first express  $m$  in binary. Next, compute the least positive residues (remainders) of  $16^2, 16^4, 16^8, \dots, 16^{2^j}$ , where  $(j+1)$  is the number of digits in the binary expansion of  $m$ . Do this sequentially, always squaring the previous result, and reducing mod  $8k+1$  in every step. Then multiply the relevant powers, one at a time, always reducing mod  $8k+1$ , to obtain  $16^m \bmod 8k+1$ . This procedure cuts back on the number of necessary arithmetic steps on a logarithmic scale [2, pp. 147–149].

For example, if we wanted to compute  $16^{16461} \bmod 46534$ , we would express 16461 in binary and get  $16461 = 2^{14} + 2^6 + 2^3 + 2^2 + 2^0$ . Hence,

$$16^{16461} \bmod 46534 = 16^{2^{14}} \cdot 16^{2^6} \cdot 16^{2^3} \cdot 16^{2^2} \cdot 16 \bmod (46534)$$

This modular exponentiation is implemented in *Mathematica* by the standard command `PowerMod[16,16461,46534]`.

**An Example.** On November 23, 2004, we programmed the above algorithm for  $n = 1011232004$ . Here is our *Mathematica* code:

```
In[1]:=
FirstSum =
  Function[{n,p},s=0;k=0;
    While[k<n+1,s=s+N[PowerMod[16,(n-k),8k+p]/(8k+p),20];k++];s];
SecondSum =
  Function[{n,p},s=0;k=n+1;
    While[k<n+21,s=s+N[16^(n-k)/(8k+p),20];k++];s];
n=1011232004;
N[Mod[4*FirstSum[n,1]+4*SecondSum[n,1]
  -2*FirstSum[n,4]-2*SecondSum[n,4]-FirstSum[n,5]-SecondSum[n,5]
  -FirstSum[n,6]-SecondSum[n,6],1]]
Out[1]= 0.2047
In[2]:= RealDigits[Out[1],16]
Out[2]= {{3, 4, 6, 7, 3, 6, 12, 4, 1, 8, 1, 13, 1}, 0}
```

Converting the fractional output .2047... to hexadecimal, we obtain

346736C4181D1

which is the consecutive sequence of hexadecimal digits of  $\pi$  starting with position  $n + 1 = 1011232005$ , accurate to more than 10 places. It took a little over 2 days to run this program.

## References

- [1] D. Bailey, P. Borwein and S. Plouffe, *On the Rapid Computation of Various Polylogarithmic constants*, *Mathematics of Computation* **66** (1997) 903–913.
- [2] K. Rosen, *Elementary Number Theory and Its Applications* (5th Edition), Addison Wesley (2005).
- [3] E. Weisstein, <http://mathworld.wolfram.com/PiFormulas.html>