# How to Compute $\pi$ to 1'000 Decimals

Walter Gander

ETH and HKBU

Qian Weichang College

Shanghai University

April 2016

## Why this talk?

$$\pi = \frac{\text{circumference}}{\text{diameter}} \text{ of a circle}$$

Using a computer algebra system e.g. MAPLE we get

```
Digits:=1000;
                                      1000
evalf(Pi);
```

3.141592653589793238462643383279502884197169399375105820974944
59230781640628620899862803482534211706798214808651328230664709
38446095505822317253594081284811174502841027019385211055596446
22948954930381964428810975665933446128475648233786783165271201
90914564856692346034861045432664821339360726024914127372458700
66063155881748815209209628292540917153643678925903600113305305
48820466521384146951941511609433057270365759591953092186117381
93261179310511854807446237996274956735188575272489122793818301
19491298336733624406566430860213949463952247371907021798609437
02770539217176293176752384674818467669405132000568127145263560
82778577134275778960917363717872146844090122495343014654958537
10507922796892589235420199561121290219608640344181598136297747
71309960518707211349999998372978049951059731732816096318595024
45945534690830264252230825334468503526193118817101000313783875
28865875332083814206171776691473035982534904287554687311595628
63882353787593751957781857780532171226806613001927876611195909
216420199

## Questions

- How does MAPLE do this?
- Can we develop a program?
- What kind of algorithms are available?

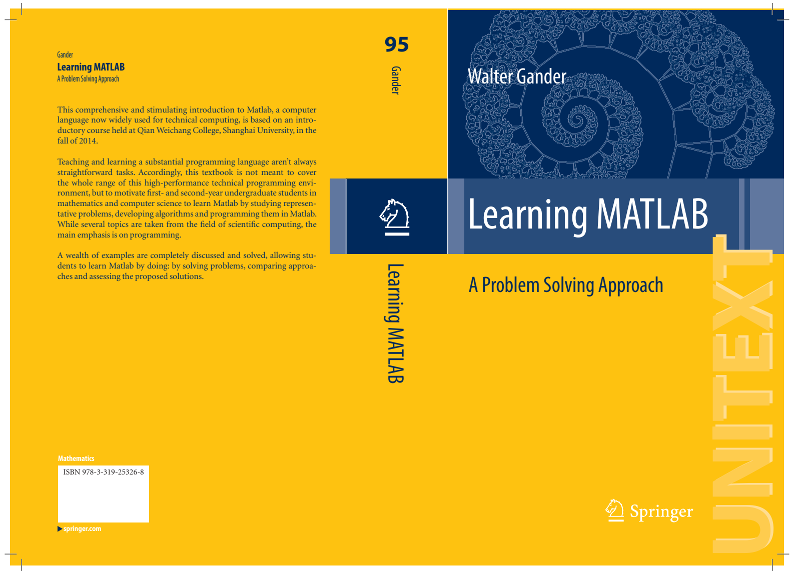
# Student Programming Problem

- MATLAB-course Fall 2014 at Qian Weichang College
- Script has become a book:

  WALTER GANDER, Learning Matlab - A Problem Solving Approach.

  Springer, 2015, ISBN: 978-3-319-25327-5, 149p.

  `http://www.springer.com/us/book/9783319253268`



## $\pi$ is a Fascinating Number

- Unsolved problem in ancient Greece:

  construct with compass and straightedge a square with same area as a given circle

- Galois theory proved: impossible

- We know circle area is $\pi r^2$

- $\pi = 3.1415\ldots$ a transcendental number

- Idiomatic expression:
  "squaring the circle" $\iff$ try to do the impossible!

# Calculations of $\pi$ (incomplete selection – many contributors)

| | | |
|---|---|---|
| Archimedes of Syracuse (287-212 BC) | 22/7=3.14... | Greek |
| Claudius Ptolemy (90-168 ) | 3.1416 | Roman |
| Zu Chongzhi (429-500) | 355/113 =3.141592... | Chinese |
| Muhammad ibn Musa al-Khwarizmi (780-850 ) | 3.1416 | Arab |
| Jamshid al-Kashi (1380-1429) | 3.1415926535897 | Iranian |
| François Viète (1540-1603) | 3.14159265 | France |
| Adriaan van Roomen (1561-1615) | 3.1415926535897932 | Belgium |
| Ludolph van Van Ceulen (1540-1610) | 35 decimals | German-Dutch |
| John Machin (1686-1751) | 100 decimals | British |

| Calculations using computers[a] | decimals | |
|---|---|---|
| Takahashi Kanada Sept 1999 | 206 158 430 000 | Japan |
| Yasumasa Kanada 2002 | 1 241 100 000 000 | Japan |

Formula used by John Machin: $$\pi = 16 \arctan \frac{1}{5} - 4 \arctan \frac{1}{239}$$

[a] http://www-history.mcs.st-andrews.ac.uk/HistTopics/Pi_chronology.html

## Machin-like Formulas

- Carl Størmer, Norwegian mathematician (1874-1957), developed Machin-like formulas:

$$\frac{\pi}{4} = 6\arctan\frac{1}{8} + 2\arctan\frac{1}{57} + \arctan\frac{1}{239}$$

$$\frac{\pi}{4} = 3\arctan\frac{1}{4} + \arctan\frac{1}{20} + \arctan\frac{1}{1985}$$

$$\frac{\pi}{4} = 44\arctan\frac{1}{57} + 7\arctan\frac{1}{239} - 12\arctan\frac{1}{682} + 24\arctan\frac{1}{12943}$$

- Takahashi Kanada (206'158'430'000 decimals) used a formula by K. Takano (1982)

$$\frac{\pi}{4} = 12\arctan\frac{1}{49} + 32\arctan\frac{1}{57} - 5\arctan\frac{1}{239} + 12\arctan\frac{1}{110443}$$

- Derivation of such formulas:

`https://en.wikipedia.org/wiki/Machin-like_formula`

- IEEE-Standard for Floating Point Numbers since 1986: using 64 bits, about 16 decimal digits.

$$
\begin{array}{llll}
S & \overbrace{EEEEEEEEEEE}^{e} & & \overbrace{FFFFF\cdots FFFFF}^{m} \\
0 & 1 \qquad\qquad\quad 11 & & 12 \qquad\qquad\quad 63
\end{array}
$$

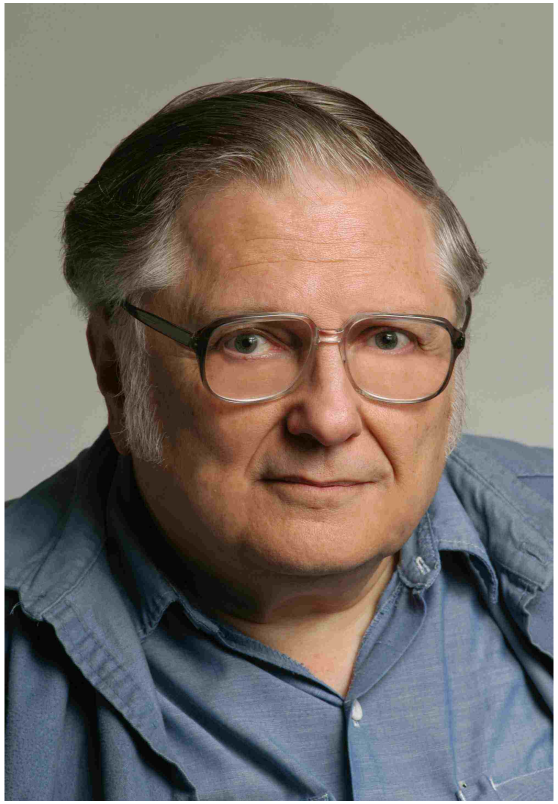
| | |
|---|---|
| S | 1 bit for the sign |
| e | 11 bits for the exponent |
| m | 52 bits for the mantissa |

- normal case: $0 < e < 2047, \quad (2^{11} = 2048)$

$$\tilde{a} = (-1)^S \times 2^{e-1023} \times 1.m$$

- Examples:

$$
\begin{aligned}
\pi &= 3.141592653589793e{+}00 \\
1/7 &= 1.428571428571428e{-}01 \\
\sqrt{2} &= 1.414213562373095e + 00 \\
e^{10} &= 2.202646579480672e + 04
\end{aligned}
$$



William Kahan (*1933)
Father IEEE FP-Standard

## Computing With More Digits

- Hardware: IEEE-Standard, floating point numbers only 16 digits.
- For more digits use software: simulate multiple precision arithmetic.
- Data structure: store digits of a multiple precision number in a integer array. In MATLAB use `uint32` (unsigned integers).
- Perform integer operations with the array elements.
  In MATLAB e.g. `mod` and `idivide`.
- Example: Division $14/3 = 4\ R\ 2 = 4\frac{2}{3}$. In MATLAB:
  $\texttt{idivide}(14, 3) = 4$ and $\texttt{mod}(14, 3) = 2$

## Teach Division to a Computer (red: remainder)

$$
\begin{array}{cccccccccc}
1 & 2 & 3 & 4 & : & 7 & = & 0 & 1 & 7 & 6 \\
0 & & & & & & & & & & \\
\cline{1-1}
{\color{red}1} & 2 & & & & & & & & & \\
 & 7 & & & & & & & & & \\
\cline{1-2}
 & {\color{red}5} & 3 & & & & & & & & \\
 & 4 & 9 & & & & & & & & \\
\cline{2-3}
 & & {\color{red}4} & 4 & & & & & & & \\
 & & 4 & 2 & & & & & & & \\
\cline{3-4}
 & & & {\color{red}2} & & & & & & &
\end{array}
$$

Thus we computed

$$\frac{1234}{7} = 176 \quad R = 2$$

DivideExample.m

# Divide a Mutiprecision Number by a Small Integer

```
function a=Divide1(p,a)
% divides the interger array a by integer number p
n=length(a);
remainder=a(1);
for i=1:n-1
  a(i)=idivide(remainder,p);
  remainder=mod(remainder,p)*10+a(i+1);
end
a(n)=idivide(remainder,p);
```

```
n=50; p=7
a=uint32(zeros(1,n));
a(1)=1; sprintf('%01d',a)
a=Divide1(p,a);sprintf('%01d',a)
```

```
p = 7
ans = 10000000000000000000000000000000000000000000000000
ans = 01428571428571428571428571428571428571428571428571
```

## Using a Formula of Machin

$$\pi = 24 \arctan \frac{1}{8} + 8 \arctan \frac{1}{57} + 4 \arctan \frac{1}{239}$$

- Check formula with floating point hardware:

```
>> 24*atan(1/8)+8*atan(1/57)+4*atan(1/239)
ans =
   3.141592653589793
```

- We have no multiple precision function $\arctan$.
  Solution: compute $\arctan$ with Taylor series

$$\frac{d}{dx} \arctan x = \frac{1}{1+x^2} = \sum_{k=0}^{\infty} (-1)^k x^{2k}, \quad |x| < 1, \qquad \text{geom. series}$$

By integration $\Longrightarrow$ $\arctan x = \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k+1}}{2k+1} = x - \frac{x^3}{3} + \frac{x^5}{5} - \cdots$

# Program arctan Function (Test first with FP-arithmetic)

```
function s=ArcusT(p)
% ARCUST computes the function arctan(1/p) for integer p>1
s=1/p; t=1/p; k=1;
while abs(t)>1e-16*s
  k=k+2; t=-t/p/p;       % update power
  s=s+t/k;               % add term to sum
end
```

```
for p=2:2:10
   [ArcusT(p) atan(1/p)]
end
ans = 0.463647609000806   0.463647609000806
ans = 0.244978663126864   0.244978663126864
ans = 0.165148677414627   0.165148677414627
ans = 0.124354994546761   0.124354994546761
ans = 0.099668652491162   0.099668652491162
```

## Multiprecision *arctan*

By comparing with function `ArcusT`:

- The only variables to become multiprecision numbers are `s` and `t`
- `k` and `p` are simple integer variables
- Operations:
  - Multiprecision addition and subtraction
  - Division of multiprecision number by small integer number
- Terminate summation if multiprecision `t` is all zero

## Addition is simple:

```
function r=Add(imin,s,a);
% ADD adds the multiprecision number a to s without carry. It is
% supposed that s>a and that imin leading components of a are zero
n=length(s);
r=s;
for i=imin+1:n
  r(i)=s(i)+a(i);
end
```

imin = 3

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 |
| 0 | 0 | 0 | 6 | 5 | 3 | 7 | 8 | 2 | 1 |
| 1 | 2 | 3 | 10 | 10 | 9 | 14 | 16 | 11 | 1 |

## but we need a carry-operation afterwards:

```
function s=Carry(s);
% CARRY normalizes the component of s such that 0 <= s(i) < 10
% and moves the carry to the next component
n=length(s);
for i=n:-1:2
  while s(i)>=10
    s(i)=s(i)-10; s(i-1)=s(i-1)+1;
  end
end
```

1 2 4 1 1 0 5 7 1 1

# Subtraction

TestSub

We have to be careful, not to compute negative intermediate results.

`uint32` are unsigned integers!

```
function r=Sub(a,b)
% SUB computes r=a-b where a and b are multiprecision numbers
% with a>b.
n=length(a);
r=a;
for i=n:-1:1
  while a(i)<b(i)                   % need  to borrow from left
    a(i)=a(i)+10; b(i-1)=b(i-1)+1;
  end
  r(i)=a(i)-b(i);
end
```

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 2 | 2 | 4 | 1 | 1 | 0 | 5 | 7 | 1 | 1 |
| 0 | 0 | 4 | 1 | 1 | 0 | 5 | 7 | 2 | 1 |
| 2 | 1 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 0 |

# Multiplication and Division by Small Integer

- Multiplication

  multiply each digit, after that a carry-operation.

$$
\begin{array}{rccccccccccc}
\mathsf{r} = & 0 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
\mathsf{s} = 24\,\mathsf{r} = & 0 & 0 & 24 & 48 & 72 & 96 & 120 & 144 & 168 & 192 & 216 \\
\mathsf{s} = \mathsf{Carry(s)} = & 0 & 2 & 9 & 6 & 2 & 9 & 6 & 2 & 9 & 3 & 6
\end{array}
$$

  pay attention to overflow: `uint32` range: $0$ to $4'294'967'295 = 2^{32} - 1$

- Division: Skip leading zeros like in addition and update $imin$

$$
\begin{array}{rccccccccccl}
\mathsf{r} = & 0 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \quad imin = 2 \\
\mathsf{s} = \mathsf{r}/17 = & 0 & 0 & 0 & 0 & 7 & 2 & 6 & 2 & 1 & 6 & 4 \quad imin = 4
\end{array}
$$

# Division by Small Integer by skipping leading zeros

```
function [s,imin]=Divide(imin,p,s)
% DIVISION divides the multiple precision number s by the integer
% number p.  The first imin components of s are zero. imin is updated
% after the division.
n=length(s);
if imin<n                          % if imin=n => s=0
  first=1;
  remainder=s(imin+1);
  for i=imin+1:n
    s(i)=idivide(remainder,p);
    if s(i)==0
      if first                     % update imin
         imin=i;
      end
    else
      first=0;
    end
    if i<n
      remainder=mod(remainder,p)*10+s(i+1);
    end
  end
end
```

# Multiprecision arctan

```
function s=AtanMultPrec(n,p)
% ATANMULTPREC computes n decimal digits of the function
%  value s=arctan(1/p) where p>1 is an integer number
s=uint32(zeros(1,n));
s(1)=1;
imin=0;                         % imin counts leading zeros in t
[s,imin]=Divide(imin,p,s);      % s=1/p
t=s;                            % first term
k=1;
sig=1;                          % the sign of the term
while imin<n
  k=k+2;
  [t,imin]=Divide(imin,p^2,t);  % new nominator of term
  h=Divide(imin,k,t);           % division without change of imin
  sig=-sig;                     % change sign
  if sig<0
    s=Sub(s,h);                 % subtract or
  else
    s=Add(imin,s,h);            % add term h to s
    s=Carry(s);
  end
end
```

# Test of AtanMultPrec

```
n=20
for p=[4,8,20,57,239, 682,1985,12943]
  s=AtanMultPrec(n,p);sprintf('%01d',s)
  atan(1/p)
end
```

| p | AtanMultPrec | atan |
|---|---|---|
| 4 | 02449786631268641544 | 0.244978663126864 |
| 8 | 01243549945467614352 | 0.124354994546761 |
| 20 | 00499583957219427615 | 0.0499583957721943 |
| 57 | 00175420600574024877 | 0.017542060057402 |
| 239 | 00041840760020747239 | 0.00418407600202075 |
| 682 | 00014662746090119564 | 0.00146627460901 2 |
| 1985 | 00005037782949130856 | 5.03778294913085 7e-04 |
| 12943 | 00000772618402233023 | 7.72618402233023 7e-05 |

## Compute $\pi$ with first formula of Størmer:

$$\frac{\pi}{4} = 6 \arctan \frac{1}{8} + 2 \arctan \frac{1}{57} + \arctan \frac{1}{239}$$

```
function s=Pi(n)
% PI computes n decimal digits of pi using the
%  first formula of Stormer.

t1=24*AtanMultPrec(n,8);      % generate the 3 terms
t2=8*AtanMultPrec(n,57);
t3=4*AtanMultPrec(n,239);
s=t1;
s=Add(1,s,t2);                % add the terms
s=Add(1,s,t3);
s=Carry(s);
```

# Results

Stoermer1

```
>> tic; s=Pi(1000); toc,sprintf('%01d',s)
Elapsed time is 28.777756 seconds.
ans =
```

3141592653589793238462643383279502884197169399375105820974944
59230781640628620899862803482534211706798214808651328230664709
38446095505822317253594081284811174502841027019385211055596446
22948954930381964428810975665933446128475648233786783165271201
90914564856692346034861045432664821339360726024914127372458700
66063155881748815209209628292540917153643678925903600113305305
48820466521384146951941511609433057270365759591953092186117381
93261179310511854807446237996274956735188575272489122793818301
19491298336733624406566430860213949463952247371907021798609437
02770539217176293176752384674818467669405132000568127145263560
82778577134275778960917363717872146844090122495343014654958537
10507922796892589235420199561121290219608640344181598136297747
71309960518707211349999998372978049951059731732816096318595024
45945534690830264252230825334468503526193118817101000313783875
28865875332083814206171776691473035982534904287554687311595628
63882353787593751957781857780532171226806613001927876611195909
216419964

Maple

216420199

s=Pi(1020)

216420198938095257201065489200

- Størmer, second formula

$$\frac{\pi}{4} = 3\arctan\frac{1}{4} + \arctan\frac{1}{20} + \arctan\frac{1}{1985}$$

```
function s=Pi2(n)
% PI2 computes n decimal digits of pi using the
% second Stoermer formula
s=12*AtanMultPrec(n,4);
t2=4*AtanMultPrec(n,20);
t3=4*AtanMultPrec(n,1985);
s=Add(1,s,t2);
s=Add(1,s,t3);
s=Carry(s);
```

```
>> tic,s=Pi2(1000); toc, sprintf('%01d',s)
Elapsed time is 37.637790 seconds.
```

...216420196

- Størmer, third formula

$$\frac{\pi}{4} = 44 \arctan \frac{1}{57} + 7 \arctan \frac{1}{239} - 12 \arctan \frac{1}{682} + 24 \arctan \frac{1}{12943}$$

```
function s=Pi3(n)
% PI3 computes n decimal digits of pi using the
% third Stoermer formula
s=176*AtanMultPrec(n,57);
t2=28*AtanMultPrec(n,239);
t3=96*AtanMultPrec(n,12943);
s=Add(1,s,t2);
s=Add(1,s,t3);
s=Carry(s);
t4=48*AtanMultPrec(n,682);
s=Sub(s,t4);
```

```
>> tic,s=Pi3(1000); toc, sprintf('%01d',s)
Elapsed time is 21.987254 seconds.
```

...21641 9244

- Machin

$$\pi = 16 \arctan \frac{1}{5} - 4 \arctan \frac{1}{239}$$

```
function s=machin(n)
% MACHIN computes n decimal digits of pi using the
%  formula Machin
s=16*AtanMultPrec(n,5);    % generate the 2 terms
t=4*AtanMultPrec(n,239);
s=Sub(s,t);
s=Carry(s);
```

```
>> tic,s=machin(1000); toc, sprintf('%01d',s)
Elapsed time is 25.225382 seconds.
```

...21420124

## Latest News

- Check the web page of Boris Gourévitch
  www.pi314.net/eng/index.php The world of $\pi$
  He cites Kanada's 1 241 100 000 000 digit record of 2002
  (1.2 trillion digits)

- Newer results by Alexander J. Yee & Shigeru Kondo on
  www.numberworld.org/misc_runs/pi-5t/details.html
  - August 2, 2010: 5 trillion (=5000 billions) Digits of Pi
  - October 17, 2011: record improved to 10 trillion $= 10^{13}$ digits.
    (Printed would fill 2 million books with 1'000 pages and 5'000 digits per page.)

## Appendix: Derivation of Machin Formula

We use the rule for addition of $\arctan$

$$\arctan(u) + \arctan(v) = \arctan\left(\frac{u+v}{1-uv}\right) \pmod{\pi}\,, \quad uv \neq 1\,.$$

it is derived from the tangent addition formula

$$\tan(\alpha+\beta) = \frac{\tan(\alpha)+\tan(\beta)}{1-\tan(\alpha)\tan(\beta)}$$

by letting

$$\alpha = \arctan(u)\,, \quad \beta = \arctan(v).$$

$$\begin{aligned} 2\arctan\frac{1}{5} &= \arctan\frac{1}{5} + \arctan\frac{1}{5} \\ &= \arctan\frac{1\times 5 + 1\times 5}{5\times 5 - 1\times 1} = \arctan\frac{10}{24} = \arctan\frac{5}{12} \end{aligned}$$

Iterating:

$$\begin{aligned} 4\arctan\frac{1}{5} &= 2\arctan\frac{1}{5} + 2\arctan\frac{1}{5} = \arctan\frac{5}{12} + \arctan\frac{5}{12} \\ &= \arctan\frac{5\times 12 + 5\times 12}{12\times 12 - 5\times 5} = \arctan\frac{120}{119} \end{aligned}$$

Using $\frac{\pi}{4} = \arctan(1)$ we get

$$\begin{aligned}4\arctan\frac{1}{5} - \frac{\pi}{4} &= 4\arctan\frac{1}{5} - \arctan\frac{1}{1} = 4\arctan\frac{1}{5} + \arctan\frac{-1}{1}\\ &= \arctan\frac{120}{119} + \arctan\frac{-1}{1} = \arctan\frac{120\times 1 + (-1)\times 119}{119\times 1 - 120\times(-1)}\\ &= \arctan\frac{1}{239}\end{aligned}$$

Thus

$$\frac{\pi}{4} = 4\arctan\frac{1}{5} - \arctan\frac{1}{239}$$