# Calculating mathematial constants using Monte Carlo simulations

Bhoris Dhanjal

## Section 1: Naive Monte Carlo Integration

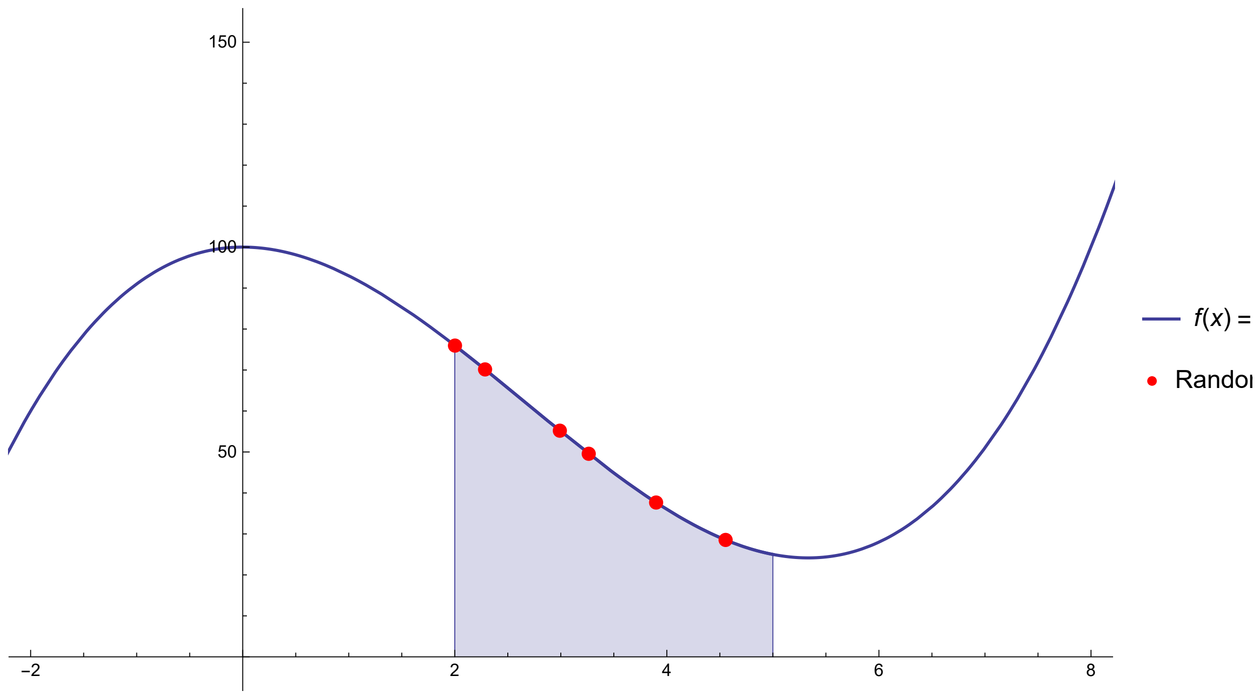
In[•]:=
```
g[x_] = 100 - 8 x^2 + x^3;
```

In[•]:=
```
a = 2; b = 5;
```

In[•]:=
```
IntegralPlot[f_, {x_, L_, U_}, {l_, u_}, opts : OptionsPattern[]] :=
 Module[{col = ColorData[1, 1]},
  Plot[{ConditionalExpression[f, x > l && x < u], f}, {x, L, U}, Prolog →
    {{col, Line[{{l, 0}, {l, f /. {x → l}}}]}, {col, Line[{{u, 0}, {u, f /. {x → u}}}]}},
   Filling → {1 → Axis}, PlotStyle → col, opts]]
```

In[•]:=
```
ExampleRand = RandomReal[{a, b}, {6}];
ExamplePoints = g[ExampleRand];
pexample = Show[IntegralPlot[g[x], {x, -10, 20}, {2, 5},
   PlotLegends → {TraditionalForm[HoldForm[f[x] = 100 - 8 x^2 + x^3]]}],
  ListPlot[Transpose[{ExampleRand, ExamplePoints}], PlotStyle → {Red},
   PlotLegends → {TraditionalForm[HoldForm[Random points on f[x]]]}],
  PlotTheme → "Detailed", PlotRange → {{-2, 8}, {150, 0}},
  AxesOrigin → {0, 0}, ImageSize → Large]
pexamplenolegend = Show[IntegralPlot[g[x], {x, -10, 20}, {2, 5}], ListPlot[
    Transpose[{ExampleRand, ExamplePoints}], PlotStyle → {Red}], PlotTheme → "Detailed",
   PlotRange → {{-2, 8}, {150, 0}}, AxesOrigin → {0, 0}, ImageSize → Large];
```

Out[•]=




In[•]:=
```
points = Table[Rectangle[{a, 0 }, {b, ExamplePoints[[x]]}], {x, 1, 5}]
```

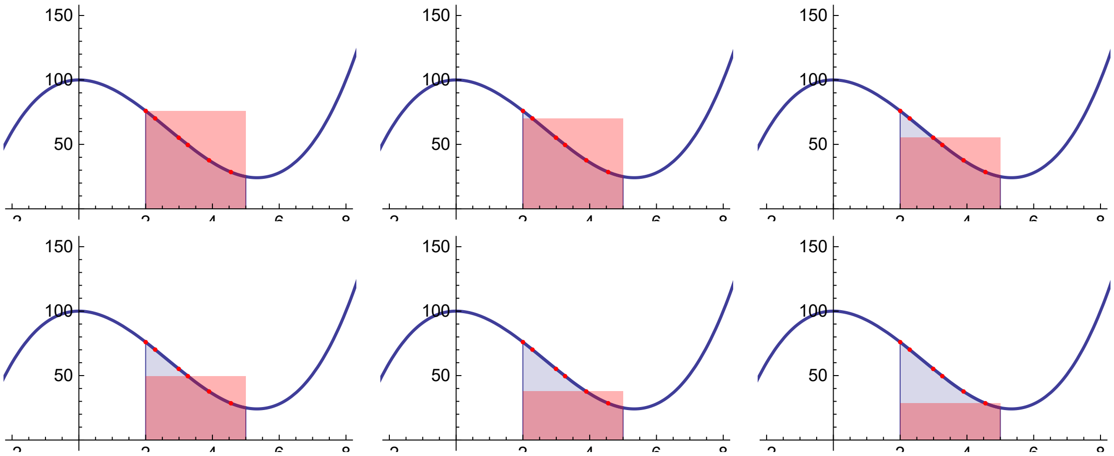
Out[•]=
```
{Rectangle[{2, 0}, {5, 37.6609}],
 Rectangle[{2, 0}, {5, 49.5594}], Rectangle[{2, 0}, {5, 28.5325}],
 Rectangle[{2, 0}, {5, 75.9617}], Rectangle[{2, 0}, {5, 55.1924}]}
```

In[•]:=
```
rectangles[x_] := {Red, Opacity[0.05], EdgeForm[Red], points};
```

In[•]:=
```
g[x_] := Show[pexamplenolegend, Graphics[
    {Red, Opacity[0.3], Rectangle[{2, 0}, {5, ExamplePoints[[x]]}], ImageSize → Medium}]];
```

In[ ]:=
```
GraphicsGrid[{{g[4], g[6], g[5]}, {g[2], g[1], g[3]}}, ImageSize → Large]
```

Out[ ]=



# Section 2: Estimating Pi

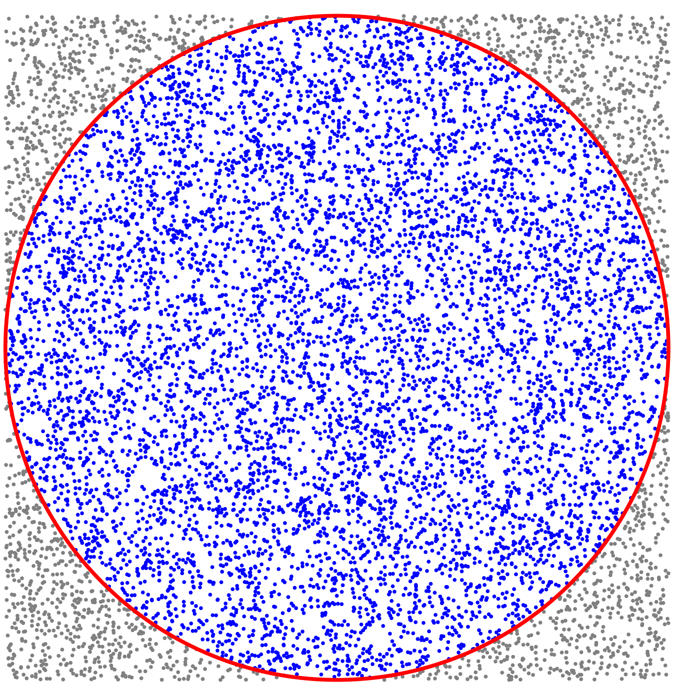
## 2.1 Elementary Method

```
pairs = RandomReal[{-1, 1}, {10000, 2}];
4 Count[Map[Norm, pairs], _?(# ≤ 1 &)] / 10000.
```

Out[ ]= 3.1512

In[ ]:=
```
Graphics[{PointSize[Small], Blue, Point@Select[pairs, Norm[#] ≤ 1 &], Gray,
  Point@Select[pairs, Norm[#] > 1 &], Red, Thick, Circle[]}, AspectRatio → 1]
```

Out[ ]=



In[•]:=
```
approxPi[n_] := 4. Count[Map[Norm, RandomReal[{-1, 1}, {n, 2}]], _? (# ≤ 1 &)] / n
```

In[•]:=
```
RepeatedExperiment = ParallelTable[approxPi[10^6], {10^3}]; // AbsoluteTiming
```

Out[•]=
```
{263.747, Null}
```

In[•]:=
```
mupi = Mean[RepeatedExperiment]
```

Out[•]=
```
3.14154
```

In[•]:=
```
NumberForm[mupi, 10]
```

Out[•]//NumberForm=
```
3.141539388
```

In[•]:=
```
(3.141539388 - Pi) / Pi * 100
```

Out[•]=
```
-0.0016955
```

In[•]:=
```
sigmapi = StandardDeviation[RepeatedExperiment]
```

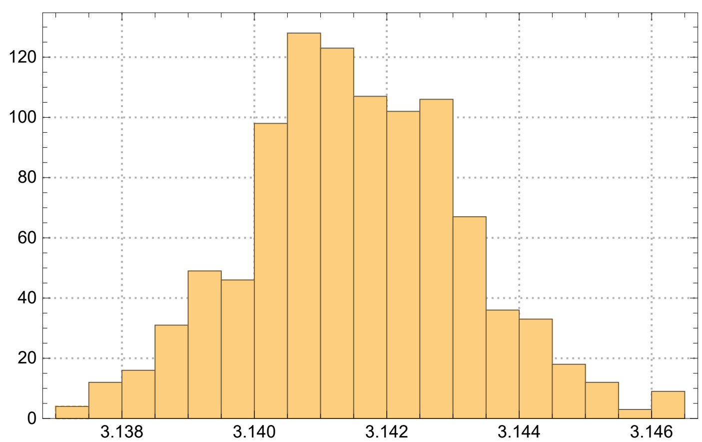
Out[•]=
```
0.00165069
```

In[•]:=
```
NumberForm[sigmapi, 10]
```

Out[•]//NumberForm=
```
0.001650685944
```

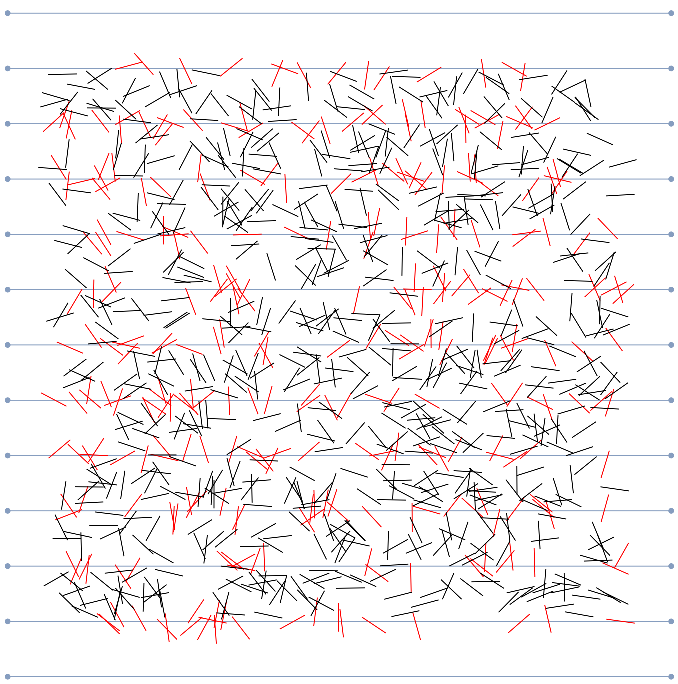
In[•]:=
```
Histogram[RepeatedExperiment, PlotTheme → "Detailed"]
```

Out[•]=




## 2.2 Buffon' s Needle

In[ ]:=
```
d = 0.2; n = 1000;
lines = MeshRegion[Join @@ Table[{{-1 - d, y}, {1 + d, y}}, {y, -1 - d, 1 + d, d}],
   Line[Partition[Range[2 Floor[2 / d + 3]], 2]]];
needles = Table[Line[{pt, RandomPoint[Circle[pt, 0.5 d]]}],
   {pt, RandomReal[{-1, 1}, {n, 2}]}];
overlap = Select[needles, ! RegionDisjoint[lines, #] &];
Show[lines, Graphics[{Red, overlap, Black, Complement[needles, overlap]}]]
N[(n) / Length[overlap]]
```

Out[ ]=


Out[ ]= 3.14465

In[ ]:=
```
NewRepeatedBuffon = ParallelTable[d = 0.2; n = 10000;
    lines = MeshRegion[Join @@ Table[{{-1 - d, y}, {1 + d, y}}, {y, -1 - d, 1 + d, d}],
      Line[Partition[Range[2 Floor[2 / d + 3]], 2]]];
    needles = Table[Line[{pt, RandomPoint[Circle[pt, 0.5 d]]}],
      {pt, RandomReal[{-1, 1}, {n, 2}]}];
    overlap = Select[needles, ! RegionDisjoint[lines, #] &];
    N[(n) / Length[overlap]], {10^3}]; // AbsoluteTiming
```

Out[ ]= {2607.03, Null}

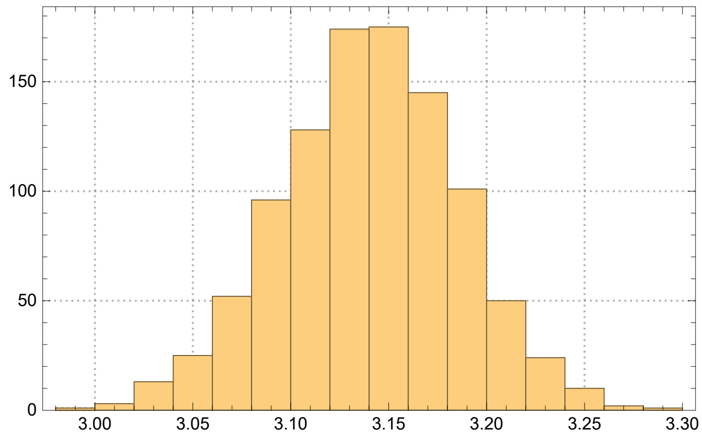
In[ ]:= `Mean[NewRepeatedBuffon]`

Out[ ]= 3.14018

In[ ]:= `StandardDeviation[NewRepeatedBuffon]`

Out[ ]= 0.0451035

In[◦]:=
```
Histogram[NewRepeatedBuffon, PlotTheme → "Detailed"]
```

Out[◦]=



In[◦]:=
```
ScientificForm[100 (Mean[NewRepeatedBuffon] - Pi) / Pi]
```

Out[◦]//ScientificForm=

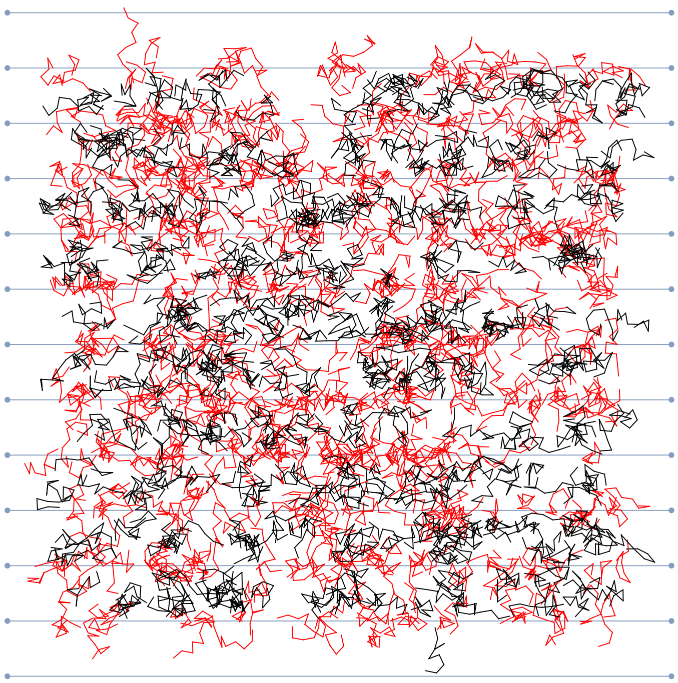
$-4.4808 \times 10^{-2}$

## 2.3 Buffon' s Noodle

In[◦]:=
```
generateNoodle[l_, np_, cent_] :=
 Block[{ls = l / np, pts}, pts = RandomPoint[Circle[{0, 0}, ls], np];
  Line /@ Partition[(cent + #) & /@ Accumulate[pts], 2, 1]]
```

```
d = 0.2; n = 1000;
lines = MeshRegion[Join @@ Table[{{-1 - d, y}, {1 + d, y}}, {y, -1 - d, 1 + d, d}],
   Line[Partition[Range[2 Floor[2 / d + 3]], 2]]];
noodles = Table[generateNoodle[d, 10, pt], {pt, RandomReal[{-1, 1}, {n, 2}]}];
ints = With[{nood = #}, RegionDisjoint[#, lines] & /@ nood] & /@ noodles;
overlap = Extract[noodles, Position[And @@ # & /@ ints, False]];
Show[lines, Graphics[{Red, overlap, Black, Complement[noodles, overlap]}]]
N[(2 n l) / (Count[ints, False, 2] d)]
```

Out[•]=



Out[•]= 3.36984

```
In[•]:= ParallelRepeatedBuffonNoodle = ParallelTable[d = 0.2; n = 10000;
   lines = MeshRegion[Join @@ Table[{{-1 - d, y}, {1 + d, y}}, {y, -1 - d, 1 + d, d}],
      Line[Partition[Range[2 Floor[2 / d + 3]], 2]]];
   noodles = Table[generateNoodle[d, RandomInteger[{2, 10}], pt],
      {pt, RandomReal[{-1, 1}, {n, 2}]}];
   ints = With[{nood = #}, RegionDisjoint[#, lines] & /@ nood] & /@ noodles;
   overlap = Extract[noodles, Position[And @@ # & /@ ints, False]];
   n / Length[overlap], 250]; // AbsoluteTiming
```

Out[•]= {2228.07, Null}

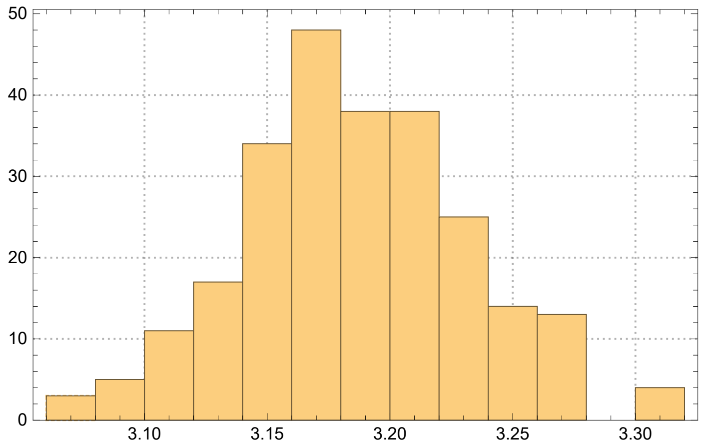
```
In[•]:= noodlemean = Mean[ParallelRepeatedBuffonNoodle];
   N[noodlemean, 15]
   noodlesigma = StandardDeviation[ParallelRepeatedBuffonNoodle];
   N[noodlesigma, 15]
```

Out[•]= 3.18614572347265

Out[•]= 0.0460212421729219

```
In[•]:= Histogram[ParallelRepeatedBuffonNoodle, PlotTheme → "Detailed"]
```

Out[•]=




In[•]:= ScientificForm$\left[100\ \frac{3.1861457234726520996723632985374326486 9 - \text{Pi}}{\text{Pi}}\right]$

Out[•]//ScientificForm=

```
1.4181682603551276098595484017515O1529
```

# Section 3: Euler-Mascheroni Constant and Stieltjes Constants

## 3.1 Estimating Euler-Mascheroni Constant by Gumbel Distribution

```
In[•]:= approxgamma[n_] := Mean[-Log[-Log[RandomReal[{0, 1}, {n}]]]]
```

In[•]:= approxgamma$\left[10^6\right]$

```
Out[•]= 0.577385
```

```
In[•]:= RepeatedGammaExp[n_, repeat_] := ParallelTable[approxgamma[n], {repeat}];
```

In[•]:= gammaresult = RepeatedGammaExp$\left[10^6, 10^4\right]$; // AbsoluteTiming

```
Out[•]= {313.266, Null}
```

```
In[•]:= mugamma = Mean[gammaresult]

Out[•]= 0.577213
```

```
In[•]:= sigmagamma = StandardDeviation[gammaresult]

Out[•]= 0.00128127
```

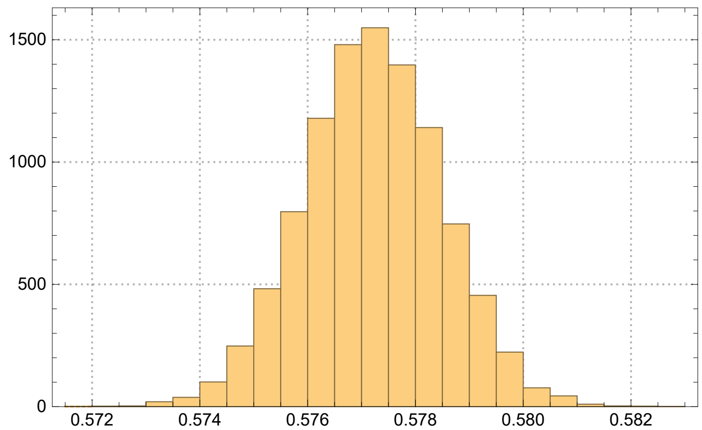
In[•]:= N$\left[100\ \frac{\text{mugamma} - \text{EulerGamma}}{\text{EulerGamma}}\right]$

```
Out[•]= -0.000482422
```

In[●]:=
```
ScientificForm[-0.000482422]
```
Out[●]//ScientificForm=

$-4.82422 \times 10^{-4}$

In[●]:=
```
Histogram[gammaresult, PlotTheme → "Detailed"]
```
Out[●]=




## 3.2 Estimating Stieltjes constants by approximating an integral

In[●]:=
```
Needs["Compile`"]
```

In[●]:=
```
sub = 2000;
tlist = Subdivide[0., 2. Pi, sub];
zeta = Zeta[1. + Exp[I tlist]];
cF = With[{zeta = zeta, tlist = tlist, L = sub / (2. Pi)},
   Compile[{{t, _Real}}, Block[{i, λ}, i = Floor[Mod[t, 2. Pi] L] + 1;
     λ = (t - tlist[[i]]) L;
     (1. - λ) zeta[[i]] + λ zeta[[i + 1]]], CompilationTarget → "C",
    RuntimeAttributes → {Listable}, Parallelization → True, RuntimeOptions → "Speed"]];
sint[n_, x_] := Exp[-n I x] cF[x]
lowerlim = 0.;
upperlim = 2. Pi;
ParallelRepeatedStieltjesIntegral[n_, points_, repeat_] :=
  (((-1)^n n!) / (2. Pi) (upperlim - lowerlim))
   ParallelTable[Mean[sint[n, RandomReal[{lowerlim, upperlim}, {points}]]],
    {repeat}, Method → "CoarsestGrained"]
```

In[●]:=
```
s1 = ParallelRepeatedStieltjesIntegral[1, 10^5, 10^4]; // AbsoluteTiming
```
Out[●]= {52.9252, Null}

In[●]:=
```
Mean[Re[s1]]
```
Out[●]= -0.0728663

In[●]:=
```
StandardDeviation[Re[s1]]
```
Out[●]= 0.00258171

In[●]:=
```
ScientificForm[N[100 (0.07286628889273356 - StieltjesGamma[1])/StieltjesGamma[1]]]
```
Out[●]//ScientificForm=

$-2.00069 \times 10^{2}$

In[●]:=
```
hs1 = Histogram[Re[s1], PlotTheme → "Detailed", PlotLabel → γ1];
```
In[●]:=
```
s2 = ParallelRepeatedStieltjesIntegral[2, 10^5, 10^4]; // AbsoluteTiming
```
Out[●]= {52.1139, Null}

In[●]:=
```
Mean[Re[s2]]
StandardDeviation[Re[s2]]
```
Out[●]= -0.00969426

Out[●]= 0.00523629

In[●]:=
```
ScientificForm[N[100 (-0.009694260647498874 - StieltjesGamma[2])/StieltjesGamma[2]]]
```
Out[●]//ScientificForm=

$4.02199 \times 10^{-2}$

In[●]:=
```
hs2 = Histogram[Re[s2], PlotTheme → "Detailed", PlotLabel → γ2];
```
In[●]:=
```
s3 = ParallelRepeatedStieltjesIntegral[3, 10^5, 15000]; // AbsoluteTiming
```
Out[●]= {92.3126, Null}

In[●]:=
```
Mean[Re[s3]]
StandardDeviation[Re[s3]]
```
Out[●]= 0.00204329

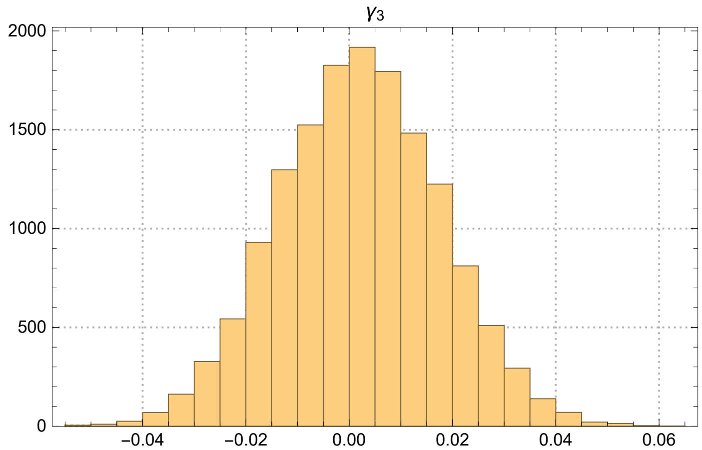
Out[●]= 0.0155079

In[●]:=
```
ScientificForm[N[100 (0.0020432938187206046` - StieltjesGamma[3])/StieltjesGamma[3]]]
```
Out[●]//ScientificForm=

$-5.13216 \times 10^{-1}$

In[•]:=
```
hs3 = Histogram[Re[s3], 20, PlotTheme → "Detailed", PlotLabel → γ3]
```

Out[•]=




In[•]:=
```
s4 = ParallelRepeatedStieltjesIntegral[4, 10^5, 30000]; // AbsoluteTiming
```

Out[•]=
```
{189.212, Null}
```

In[•]:=
```
Mean[Re[s4]]
StandardDeviation[Re[s4]]
```

Out[•]=
```
0.00233049
```

Out[•]=
```
0.0613601
```

In[•]:=

$$\text{ScientificForm}\left[\text{N}\left[100\,\frac{0.002330492470272919\grave{}\ - \text{StieltjesGamma}[4]}{\text{StieltjesGamma}[4]}\right]\right]$$

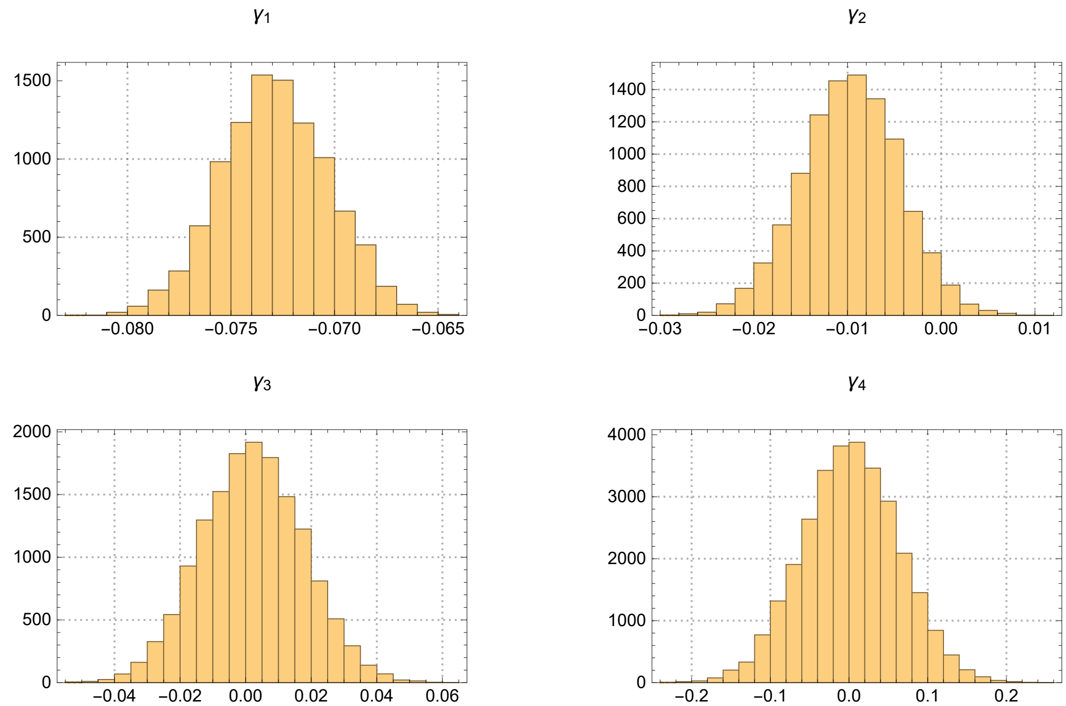
Out[•]//ScientificForm=

$2.20283 \times 10^{-1}$

In[•]:=
```
hs4 = Histogram[Re[s4], PlotTheme → "Detailed", PlotLabel → γ4];
```

In[•]:=
```
GraphicsGrid[{{hs1, hs2}, {hs3, hs4}}]
```

Out[•]=




# Section 4: Estimating Euler’s number

In[•]:=
```
ParallelRepeatedEExp[n_, repeat_] :=
 ParallelTable[Mean[Table[Module[{u = Random[], t = 1}, While[u < 1, u = Random[] + u;
      t++];
     t], {n}]], {repeat}]
```

In[•]:=
```
RepeatedEExp = ParallelRepeatedEExp[10^6, 10^4]; // AbsoluteTiming
```

Out[•]= {1265.84, Null}

In[•]:=
```
muEexp = N[Mean[RepeatedEExp]]
```

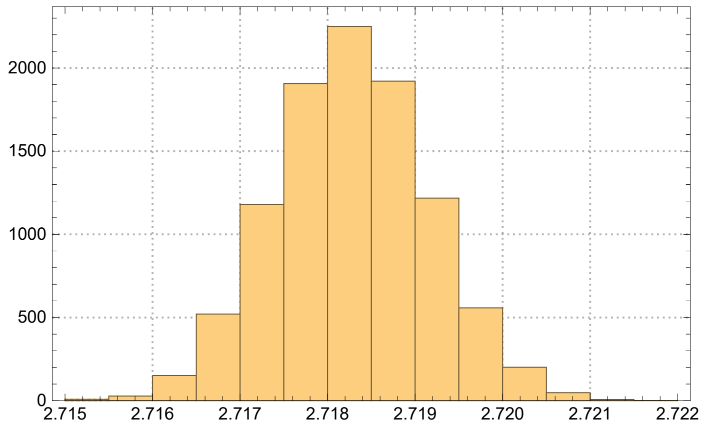
Out[•]= 2.71827

In[•]:=
```
sigmaEexp = N[StandardDeviation[RepeatedEExp]]
```

Out[•]= 0.00087055

In[ ]:=
```
Histogram[RepeatedEExp, PlotTheme → "Detailed"]
```

Out[ ]=




In[ ]:=
```
ScientificForm[N[100 (muEexp - E)/E]]
```

Out[ ]//ScientificForm=

$-3.40522 \times 10^{-4}$

# Section 5: Estimating Phi

In[ ]:=
```
phiint[x_] = 3/2 + 1/(4 Sqrt[x]);
```

In[ ]:=
```
phiblim = 4; phiulim = 5;
```

In[ ]:=
```
RepeatedPhiIntegral[points_, repeat_] := ParallelTable[N[
    (phiulim - phiblim)/points Total[phiint[RandomReal[{phiblim, phiulim}, {points}]]]], {repeat}]
```

In[ ]:=
```
PhiIntegralData = RepeatedPhiIntegral[10^6, 10^4]; // AbsoluteTiming
```

Out[ ]= {274.546, Null}

In[ ]:=
```
muphi = Mean[PhiIntegralData]
```

Out[ ]= 1.61803

In[ ]:=
```
NumberForm[muphi, 10]
```

Out[ ]//NumberForm=

1.618033999

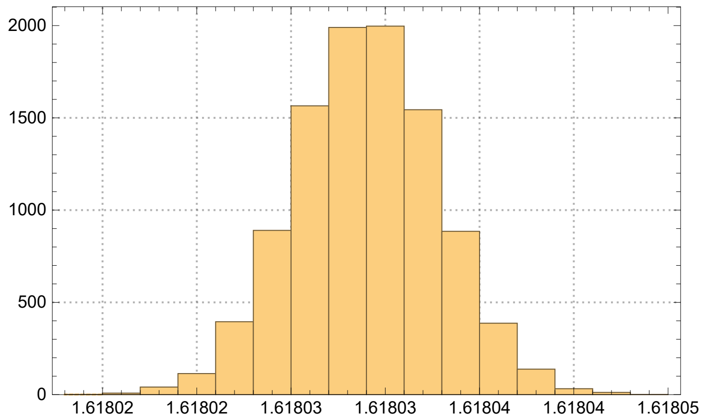
In[ ]:=
```
sigmaphi = StandardDeviation[PhiIntegralData]
```

Out[ ]= $3.79948 \times 10^{-6}$

In[ ]:=
```
N[GoldenRatio, 10]
```

Out[ ]= 1.618033989

In[ ]:= `Histogram[PhiIntegralData, PlotTheme → "Detailed"]`

Out[ ]=



In[ ]:= `ScientificForm[N[100 (muphi - GoldenRatio)/GoldenRatio]]`

$$\text{ScientificForm}\left[N\left[100\,\frac{\text{muphi} - \text{GoldenRatio}}{\text{GoldenRatio}}\right]\right]$$

Out[ ]//ScientificForm=

$6.54299 \times 10^{-7}$