Calculating mathematial constants using Monte Carlo simulations

https : // github.com/BhorisDhanjal/MonteCarloMathsConstants

- Bhoris Dhanjal

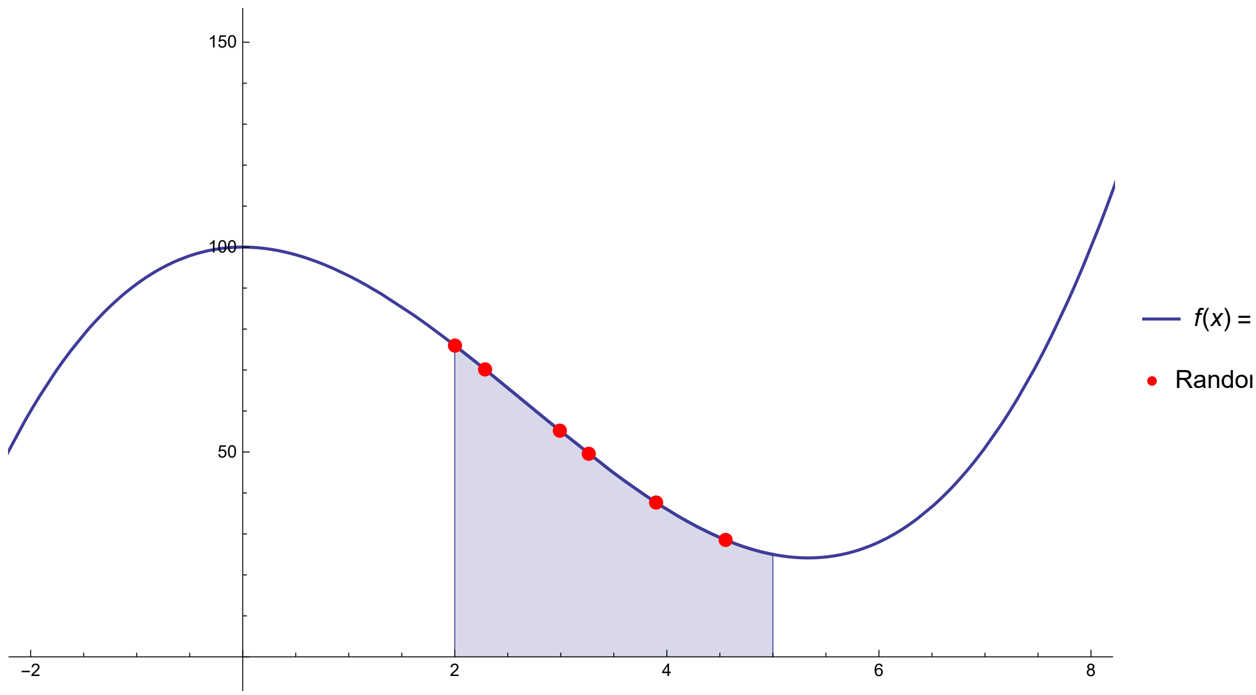
Section 1: Naive Monte Carlo Integration

```
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}]
```

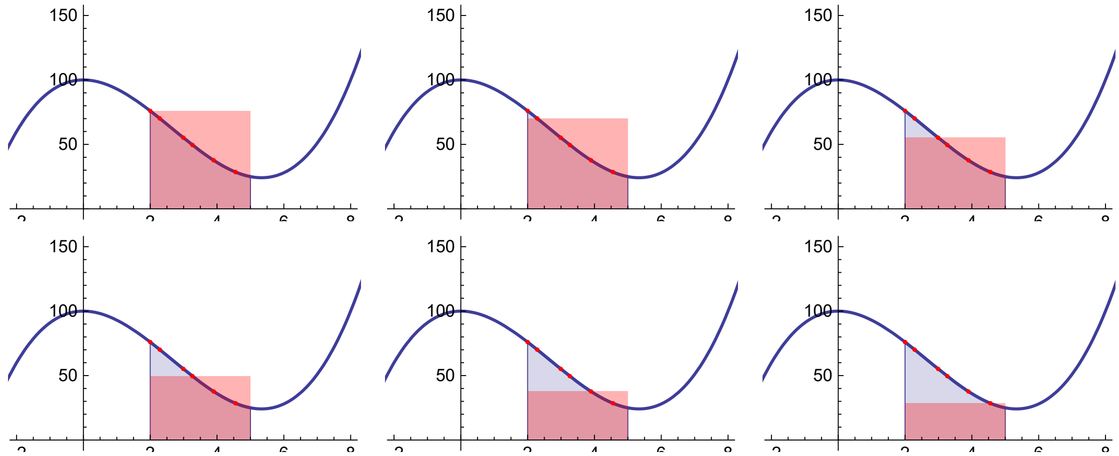
Out[●]=
```
{Rectangle[{2, 0}, {5, 37.6609}],
 Rectangle[{2, 0}, {5, 49.5594}], Rectangle[{2, 0}, {5, 28.5325}],
 Rectangle[{2, 0}, {5, 75.9617}], Rectangle[{2, 0}, {5, 55.1924}]}
```

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

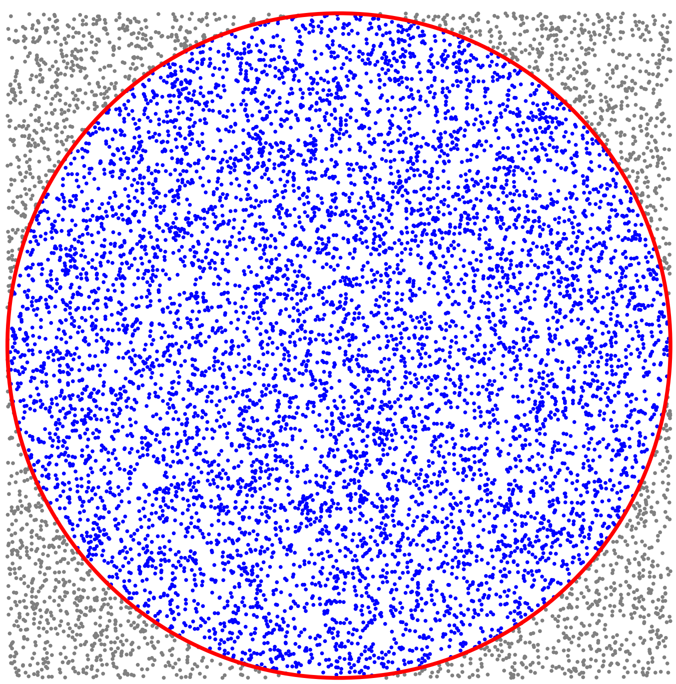
2.1 Elementary Method

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

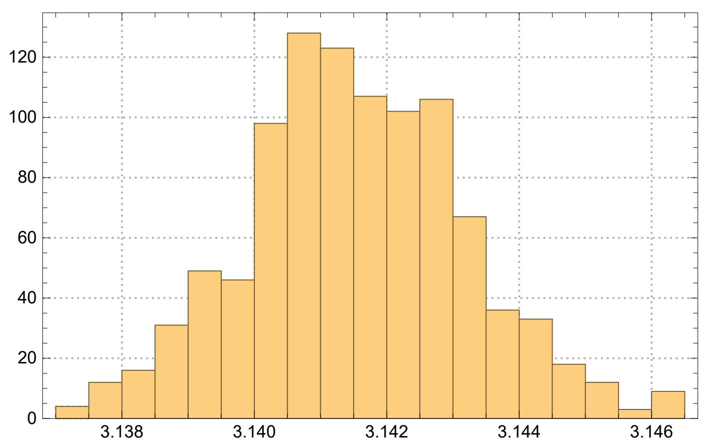
Out[•]= 3.1512

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

Out[•]=

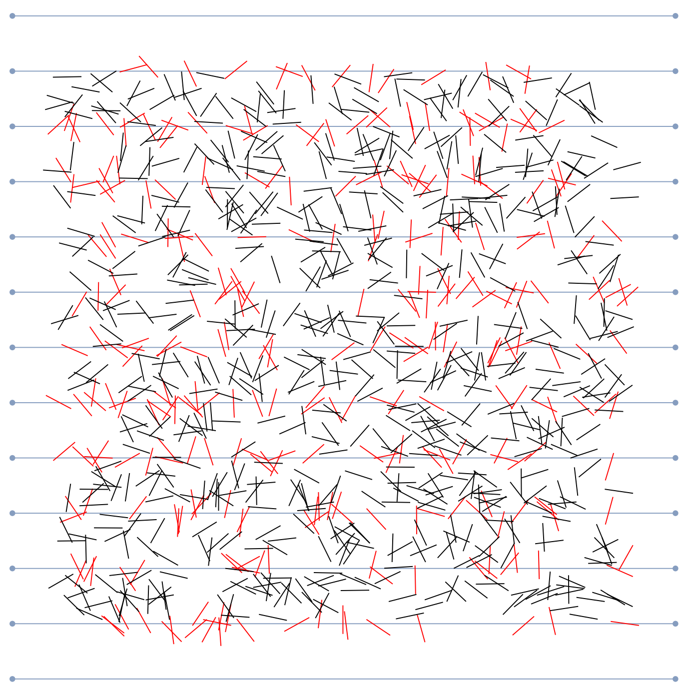


```
In[]:= approxPi[n_] := 4. Count[Map[Norm, RandomReal[{-1, 1}, {n, 2}]], _? (# ≤ 1 &)] / n

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]`

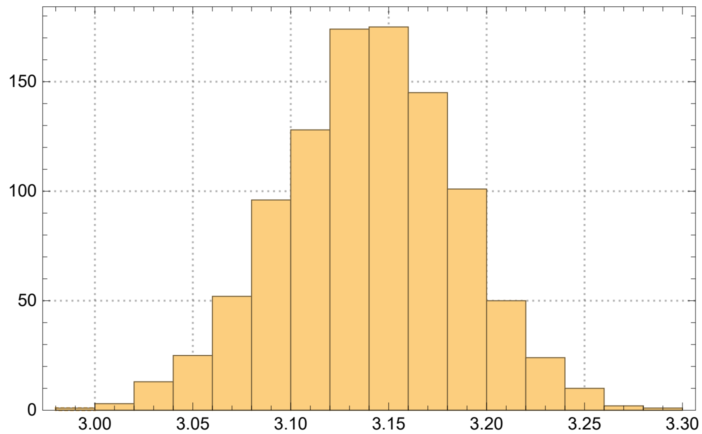
Out[ ]= 3.14018

In[ ]:= `StandardDeviation[NewRepeatedBuffon]`

Out[ ]= 0.0451035

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

Out[∘]=



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

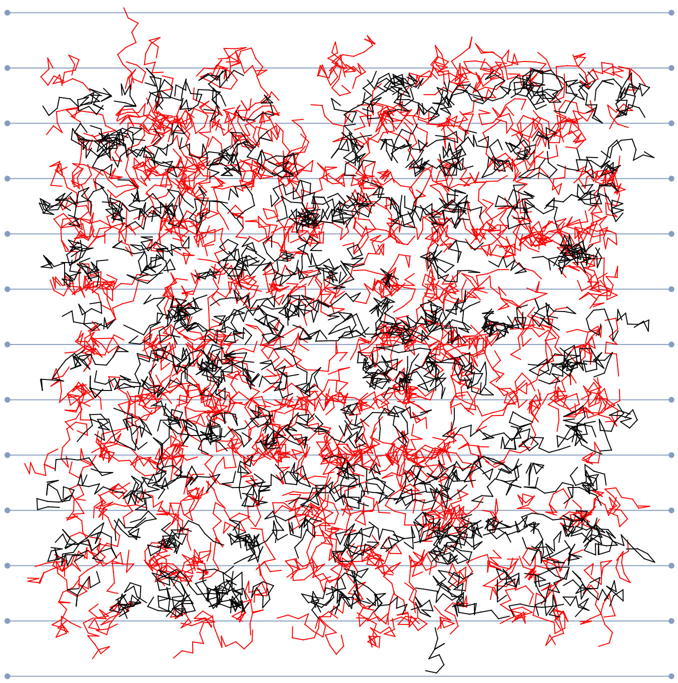
Out[∘]//ScientificForm=

-4.4808×10^{-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}

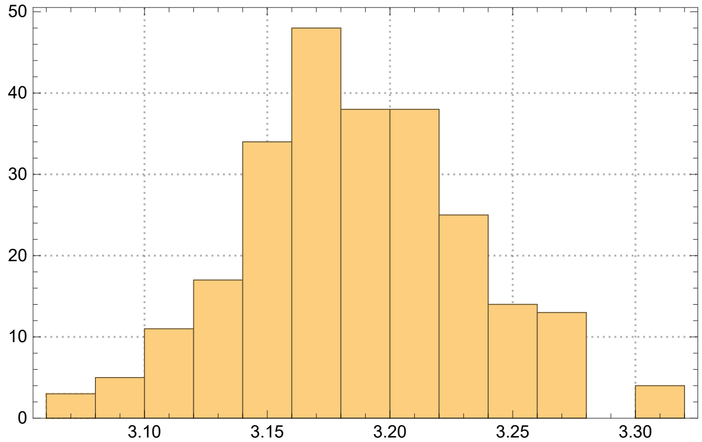
```
In[ ]:= noodlemean = Mean[ParallelRepeatedBuffonNoodle];
    N[noodlemean, 15]
    noodlesigma = StandardDeviation[ParallelRepeatedBuffonNoodle];
    N[noodlesigma, 15]
```

Out[ ]= 3.18614572347265

Out[ ]= 0.0460212421729219

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

Out[•]=



In[•]:=
$$\text{ScientificForm}\left[100\,\frac{3.18614572347265209967236329853743264869 - \text{Pi}}{\text{Pi}}\right]$$

Out[•]//ScientificForm=
```
1.41816826035512760985954840175150152
```

Section 3: 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]]
```

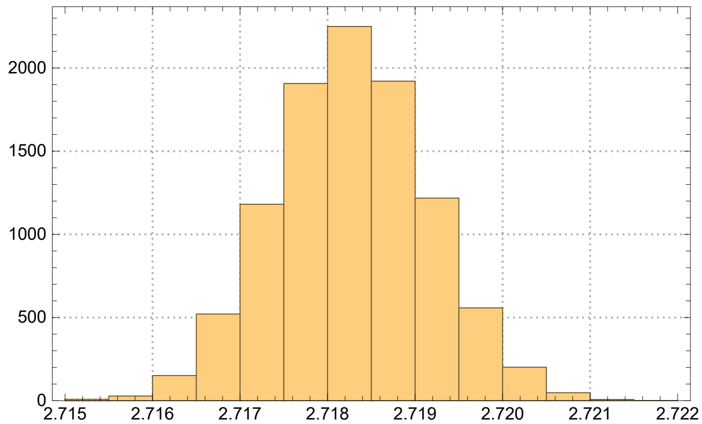
Out[•]= 2.71827

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

Out[•]= 0.00087055

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

Out[•]=



In[•]:= $\text{ScientificForm}\left[N\left[100\ \frac{\text{muEexp} - E}{E}\right]\right]$

Out[•]//ScientificForm=

-3.40522×10^{-4}

Section 4: Euler-Mascheroni Constant and Stieltjes Constants

4.1 Estimating Euler-Mascheroni Constant by Gumbel Distribution

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

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

Out[•]= 0.577385

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

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

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

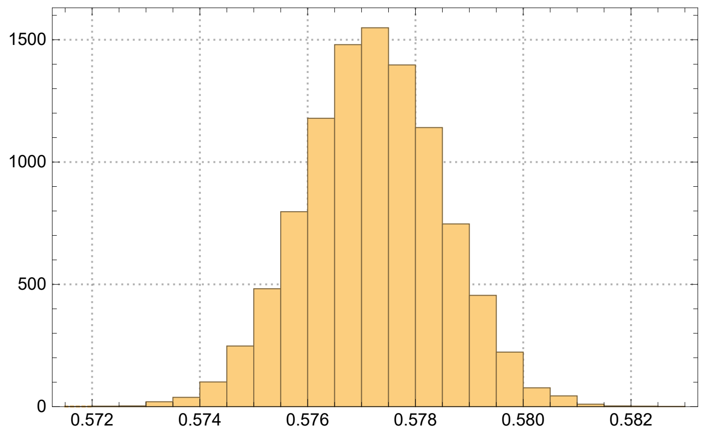
In[•]:=
```
ScientificForm[-0.000482422]
```

Out[•]//ScientificForm=

-4.82422×10^{-4}

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

Out[•]=



4.2 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=

-2.00069×10^{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=

4.02199×10^{-2}

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

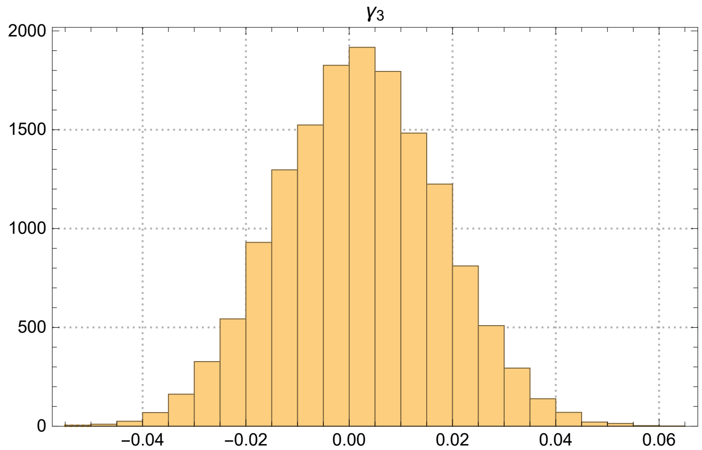
In[]:=
```
Mean[Re[s3]]
StandardDeviation[Re[s3]]
```
Out[]= 0.00204329

Out[]= 0.0155079

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

-5.13216×10^{-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[•]:=
```
ScientificForm[N[100 (0.002330492470272919` - StieltjesGamma[4]) / StieltjesGamma[4]]]
```

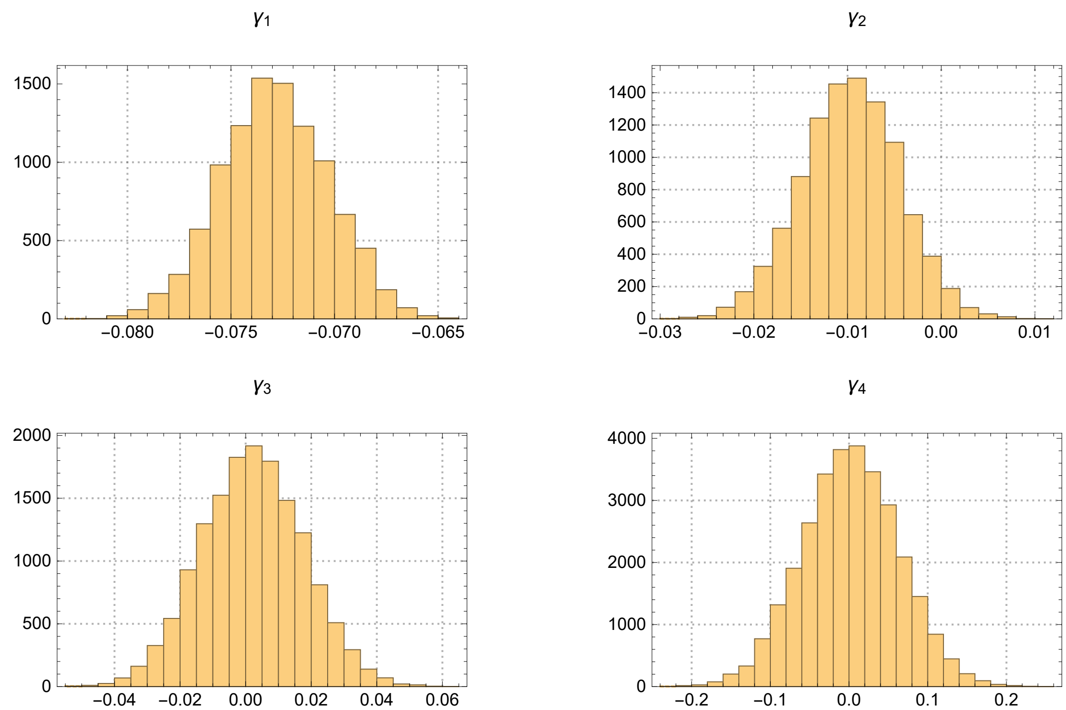
Out[•]//ScientificForm=

2.20283×10^{-1}

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

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

Out[∘]=



Section 5: Estimating Phi

$$\text{phiint[x\_] := } \frac{3}{2} + \frac{1}{4\,\text{Sqrt[x]}};$$

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]
```

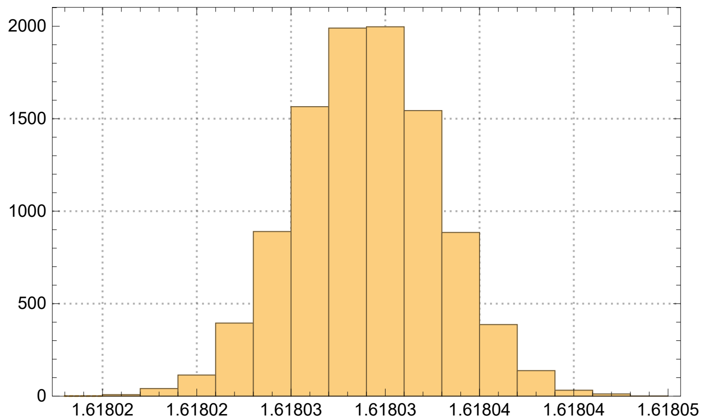
Out[∘]= 3.79948×10^{-6}

In[•]:= `N[GoldenRatio, 10]`

Out[•]= `1.618033989`

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

Out[•]=



In[•]:= $\texttt{ScientificForm}\left[\texttt{N}\left[100\ \frac{\texttt{muphi - GoldenRatio}}{\texttt{GoldenRatio}}\right]\right]$

Out[•]//ScientificForm=

6.54299×10^{-7}

Section 6: Variance Reduction

6.1 Antithetic variables

In[•]:= $\texttt{int[x\_, y\_] :=}\ \frac{\texttt{Sin[x]}}{\texttt{x}}\ \frac{\texttt{Sin[y]}}{\texttt{y}}$

In[•]:= `ClearAll[x, y]`

In[•]:=
```
naivemc = ParallelTable[ SeedRandom[1];
    (1 / (1 points)) Total[int @@@ RandomReal[{0, 1}, {1 points, 2}]], {points, 1, 10000}]; //
  AbsoluteTiming
```

Out[•]= `{55.8768, Null}`

In[•]:=
```
antimc = ParallelTable[SeedRandom[1];
    1/(2 points) Total[int @@@ (RandomReal[{0, 1}, {1 points, 2}]) +
      int @@@ (-RandomReal[{0, 1}, {1 points, 2}])], {points, 1, 10^4}]; // AbsoluteTiming
```

Out[•]= `{118.535, Null}`

In[•]:= `ColorData[97, "ColorList"]`

Out[•]= {■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■}

In[•]:= `pnaive = ListLinePlot[naivemc, PlotLegends → {"Naive"}];`

In[•]:=
```
panti = ListLinePlot[antimc, PlotStyle → ■, PlotLegends → {"Antithetic variables"}];
```

In[•]:=
```
real = Plot[y = SinIntegral[1]^2, {x, 0, 10000}, PlotRange → {{0, 10000}, {0.893, 0.897}},
    PlotStyle → {Red, Dashed}, PlotLegends → {"Actual Value"}];
```

In[•]:=
```
Show[real, pnaive, panti, PlotTheme → "Detailed", ImageSize → Large]
```

Out[•]=

6.2 Control variate

In[•]:=
```
ClearAll[x, y]
```

In[•]:=
```
Series[Sin[x]/x Sin[y]/y, {x, 0, 2}, {y, 0, 2}]
```

Out[•]=
$$\left(1-\frac{y^2}{6}+O[y]^3\right)+\left(-\frac{1}{6}+\frac{y^2}{36}+O[y]^3\right)x^2+O[x]^3$$

In[•]:=
```
SeriesCoefficient[Sin[x]/x Sin[y]/y, {x, 0, n}, {y, 0, m}]
```

Out[•]=
$$\begin{cases} \frac{\left((-i)^m+i^m\right)\left((-i)^n+i^n\right)}{4\,(1+m)!\,(1+n)!} & n \geq -1 \;\&\&\; m > -1 \\ 0 & \text{True} \end{cases}$$

In[•]:=
```
Sum[x^n y^m (Cos[m π/2] Cos[n π/2])/((1 + m)! (1 + n)!), {n, 0, 2}, {m, 0, 2}]
```

Out[•]=
$$1-\frac{x^2}{6}-\frac{y^2}{6}+\frac{x^2 y^2}{36}$$

In[•]:= FullSimplify$\left[\left((-i)^m + i^m\right)\left((-i)^n + i^n\right)\right]$

Out[•]= $4\,\mathrm{Cos}\left[\frac{m\pi}{2}\right]\,\mathrm{Cos}\left[\frac{n\pi}{2}\right]$

In[•]:= Integrate$\left[\frac{1}{36}\left(-6 + x^2\right)\left(-6 + y^2\right), \{y, 0, 1\}, \{x, 0, 1\}\right]$

Out[•]= $\frac{289}{324}$

In[•]:= Simplify$\left[\left(1 - \frac{x^2}{6}\right) + \left(\frac{-1}{6} + \frac{x^2}{36}\right) y^2\right]$

Out[•]= $\frac{1}{36}\left(-6 + x^2\right)\left(-6 + y^2\right)$

```
In[•]:= N[SinIntegral[1]^2]
        N[289/324]
```

Out[•]= 0.895073

Out[•]= 0.891975

In[•]:= cvint[a_, b_] := $\frac{\mathrm{Sin}[a]}{a}\,\frac{\mathrm{Sin}[b]}{b} - \frac{1}{36}\left(-6 + a^2\right)\left(-6 + b^2\right)$

In[•]:= cvmc = ParallelTable$\Big[$SeedRandom[1]; $\frac{1}{1\,\mathrm{points}}$ Total$\left[\right.$cvint @@@ (RandomReal[{0, 1}, {1 points, 2}])$\left.\right]$ + $\frac{289}{324}$, {points, 1, 10^4}$\Big]$; // AbsoluteTiming

Out[•]= {130.448, Null}

In[•]:= anticvmc = ParallelTable$\Big[$SeedRandom[1]; $\frac{1}{2\,\mathrm{points}}$ Total$\left[\right.$cvint @@@ (RandomReal[{0, 1}, {1 points, 2}]) + cvint @@@ (- RandomReal[{0, 1}, {1 points, 2}])$\left.\right]$ + $\frac{289}{324}$, {points, 1, 10^4}$\Big]$; // AbsoluteTiming

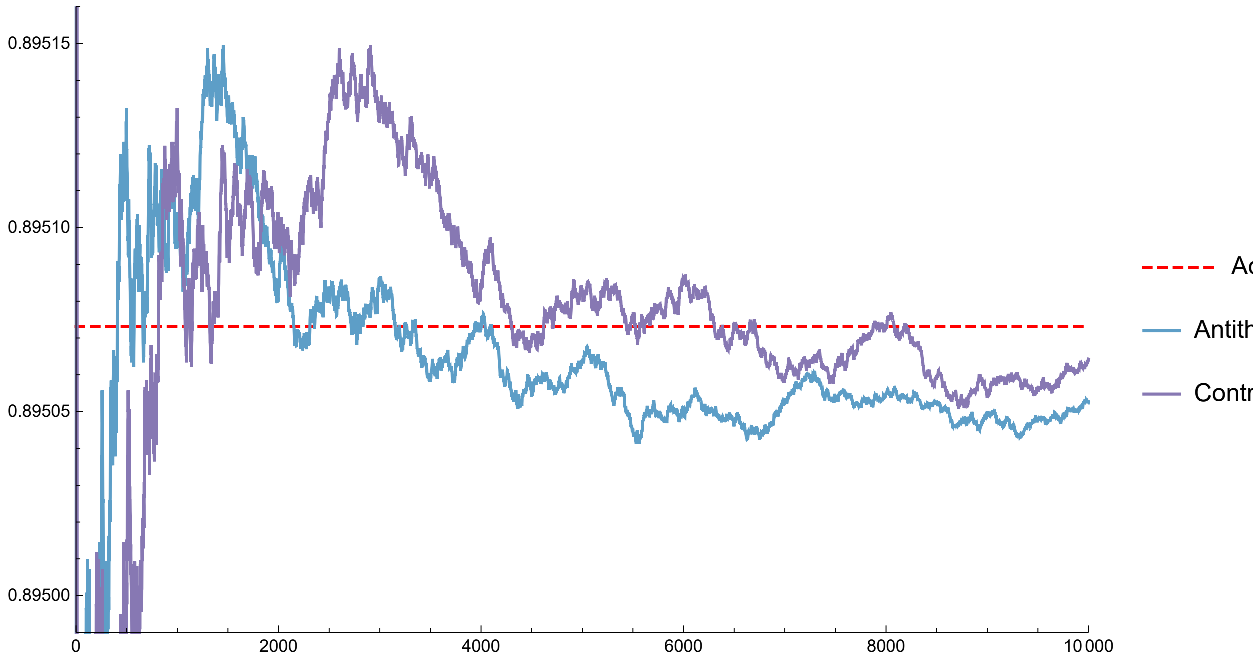
Out[•]= {316.621, Null}

```
In[•]:= pcv = ListLinePlot[cvmc, PlotLegends → {"Control variate"}, PlotStyle → ■];

In[•]:= panticv = ListLinePlot[anticvmc,
          PlotLegends → {"Antithetic control variate"}, PlotStyle → ■, PlotRange → Full];
```

In[•]:= cvreal =
  Plot$\left[\right.$y = SinIntegral$[1]^2$, {x, 0, 10000}, PlotRange → {{0, 10000}, {0.89499, 0.89516}},
   PlotStyle → {Red, Dashed}, PlotLegends → {"Actual Value"}$\left.\right]$;

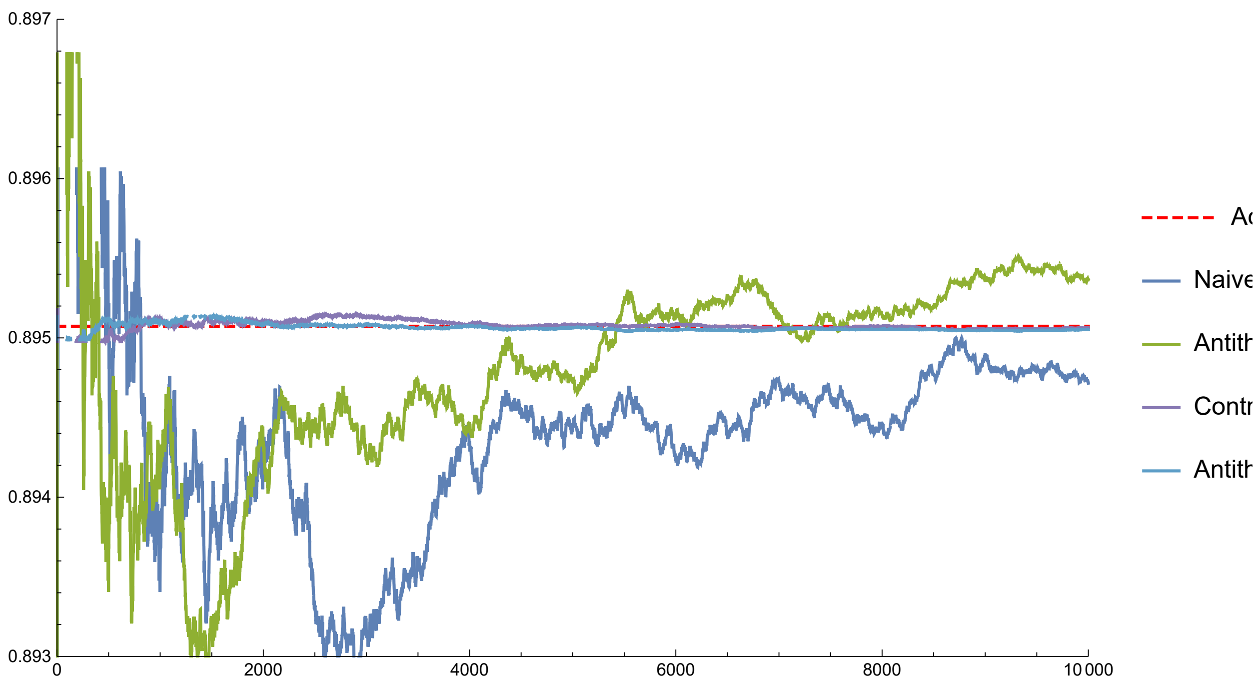
In[•]:=
```
Show[cvreal, panticv, pcv, ImageSize → Large]
```

Out[•]=



In[•]:=
```
Show[real, pnaive, panti, pcv, panticv, ImageSize → Large]
```

Out[•]=



Section 7: Quasi Monte Carlo

7.1 Low discrepancy sequences

7.1.1 Halton sequence

In[]:=
```
halton[base_, len_] := Table[With[{digits = Reverse@IntegerDigits[n, base]},
    Sum[base^(-ii) * digits[[ii]], {ii, Length[digits]}]], {n, len}]
```

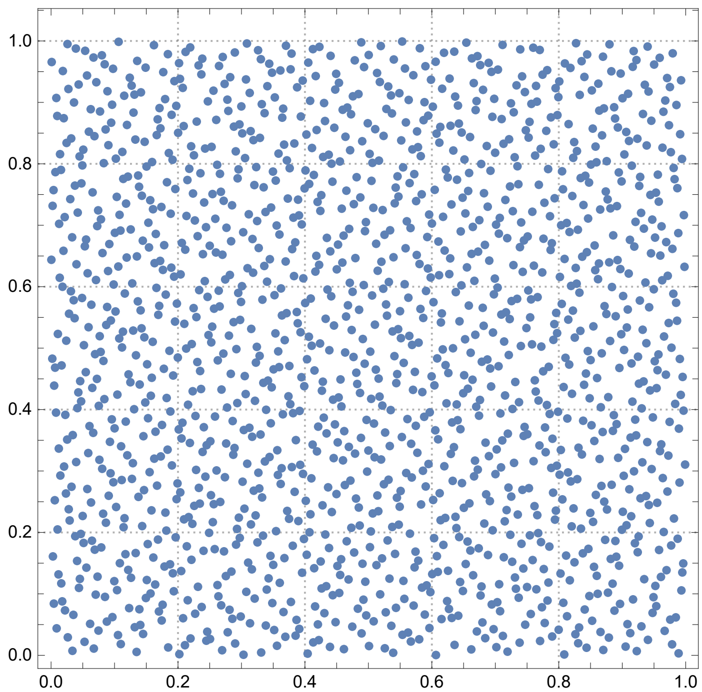
In[]:=
```
h2 = Transpose@{halton[Prime[1], 1500], halton[Prime[2], 1500]};
```

In[]:=
```
h2example = Transpose@{halton[Prime[10], 1500], halton[Prime[11], 1500]};
```

In[]:=
```
h3 = Transpose@{halton[Prime[2], 1500], halton[Prime[3], 1500], halton[Prime[5], 1500]};
```

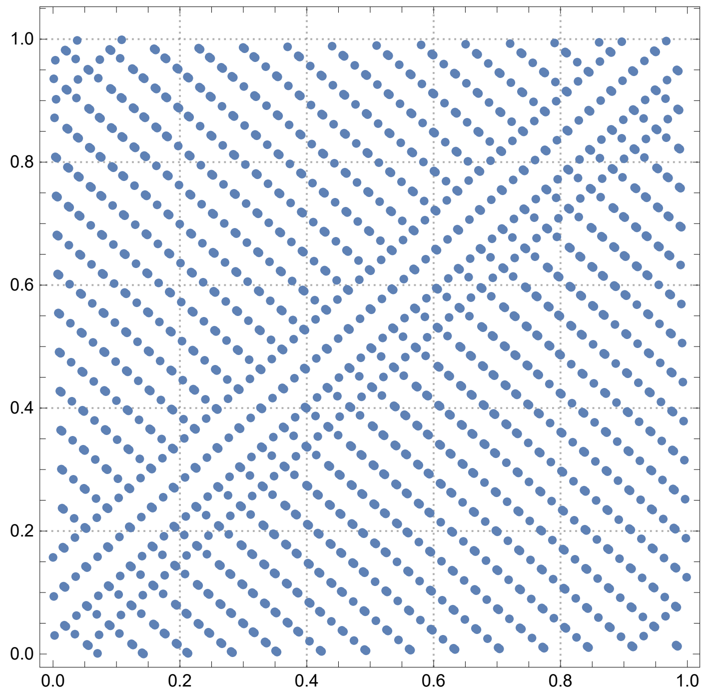
In[]:=
```
ListPlot[h2, AspectRatio → 1, PlotStyle → PointSize[Medium], PlotTheme → "Detailed"]
```

Out[]=



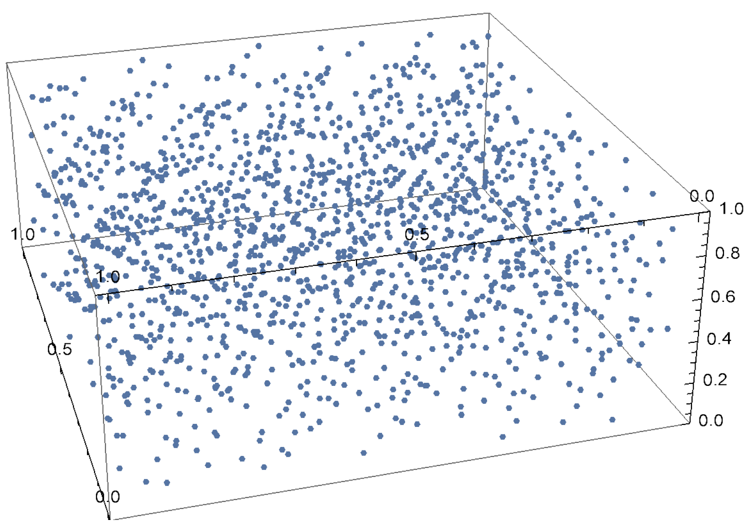
In[∘]:= `ListPlot[h2example, AspectRatio → 1, PlotStyle → PointSize[Medium], PlotTheme → "Detailed"]`

Out[∘]=



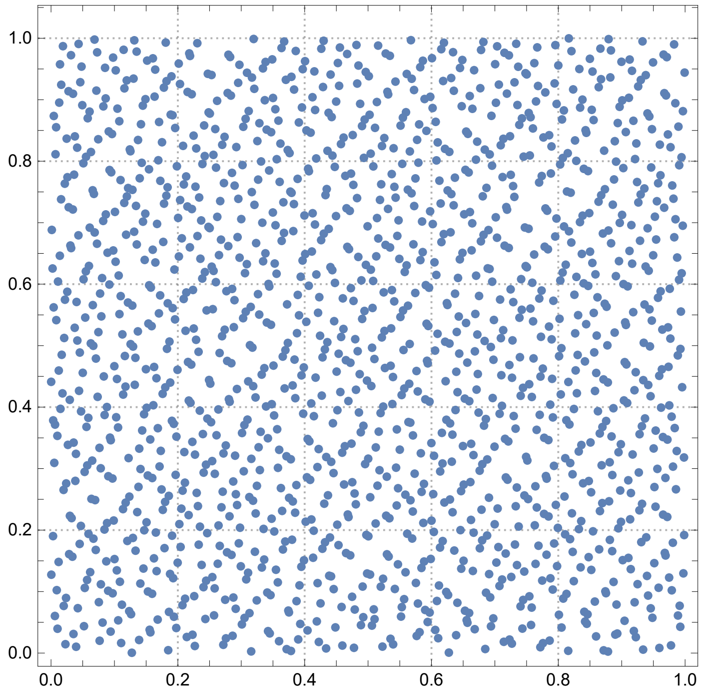
In[∘]:= `ListPointPlot3D[h3]`

Out[∘]=

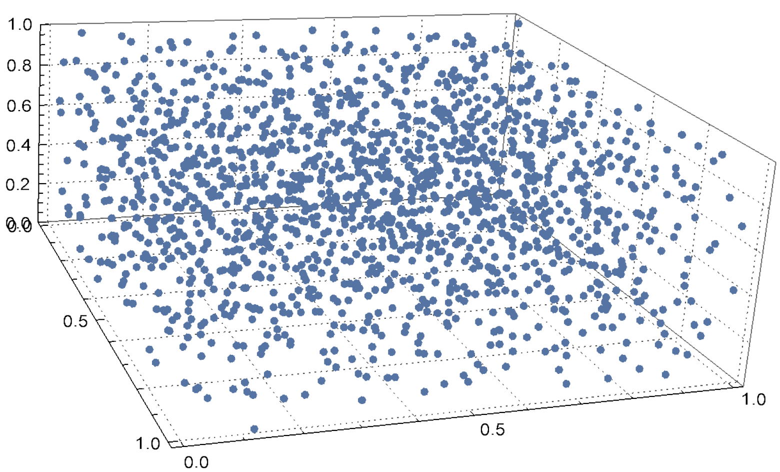


7.1.2 Niederreiter sequence

In[•]:=
```
ListPlot[
 BlockRandom[SeedRandom[Method → {"MKL", Method → {"Niederreiter", "Dimension" → 2}}];
  RandomReal[1, {1500, 2}]], AspectRatio → 1,
 PlotStyle → PointSize[Medium], PlotTheme → "Detailed"]
```

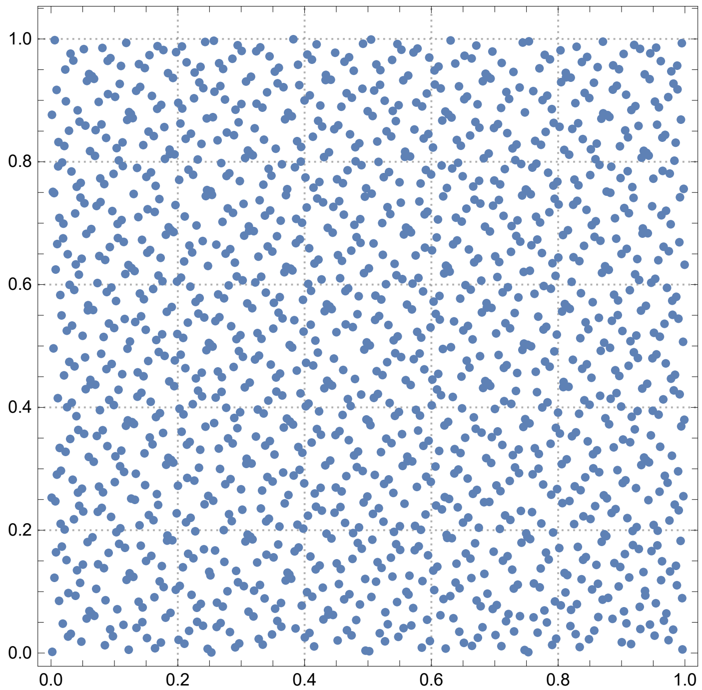
Out[•]=


In[•]:=
```
ListPointPlot3D[
 BlockRandom[SeedRandom[Method → {"MKL", Method → {"Niederreiter", "Dimension" → 3}}];
  RandomReal[1, {1500, 3}]], PlotStyle → PointSize[Medium], PlotTheme → "Detailed"]
```

Out[•]=


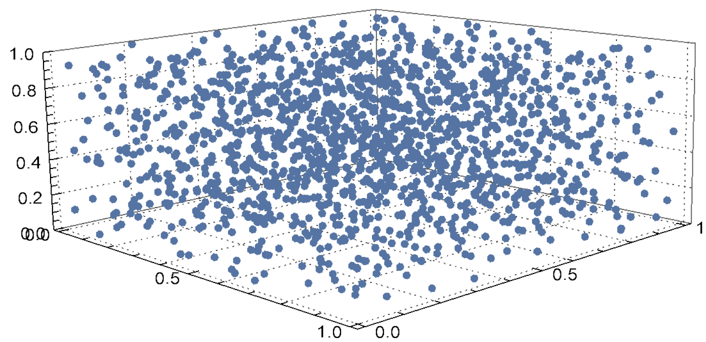
In[•]:=
```
ListPlot[BlockRandom[SeedRandom[Method → {"MKL", Method → {"Sobol", "Dimension" → 2}}];
  RandomReal[1, {1500, 2}]], AspectRatio → 1,
 PlotStyle → PointSize[Medium], PlotTheme → "Detailed"]
```

Out[•]=



In[•]:=
```
ListPointPlot3D[
 BlockRandom[SeedRandom[Method → {"MKL", Method → {"Sobol", "Dimension" → 3}}];
  RandomReal[1, {1500, 3}]], AspectRatio → 1,
 PlotStyle → PointSize[Medium], PlotTheme → "Detailed"]
```

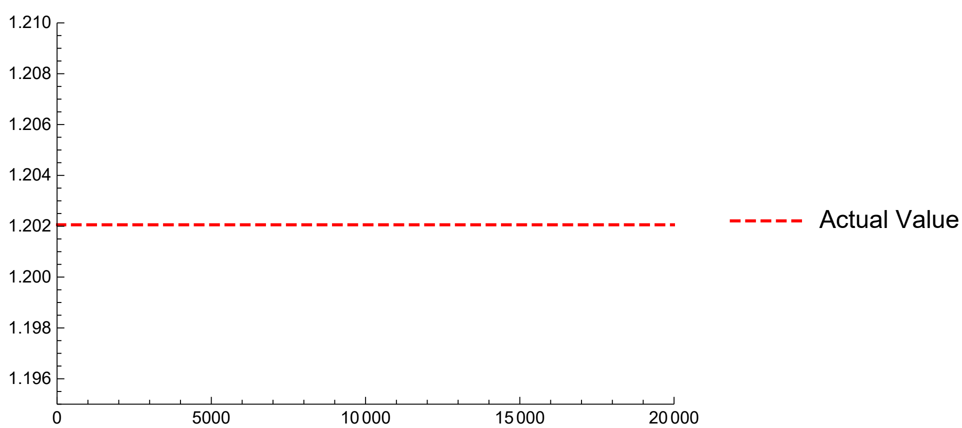
Out[ ]=



7.2 Example of Quasi Monte Carlo integration (Estimating Apery's constant)

```
real = Plot[y = Zeta[3], {x, 0, 20000}, PlotRange → {{0, 20000}, {1.195, 1.210}},
  PlotStyle → {Red, Dashed}, PlotLegends → {"Actual Value"}]
```

Out[ ]=



In[ ]:=
```
ClearAll[int, x, y, l , m, n, a, b, c]
```

$$\text{int}[x\_, y\_, z\_] := \frac{1}{1 - x\,y\,z}$$

In[•]:=
```
quasinaivemc = ParallelTable[
    SeedRandom[123, Method → {"MKL", Method → {"Niederreiter", "Dimension" → 3}}];
    1/(1 points) Total[int @@@ (RandomReal[{0, 1}, {1 points, 3}])], {points, 1, 2 × 10^4}]; //
  AbsoluteTiming
```

Out[•]=
```
{221.307, Null}
```

In[•]:=
```
naiveapery =
   ParallelTable[SeedRandom[123, Method → {"MKL", Method → {"MersenneTwister"}}];
    1/points Total[int @@@ (RandomReal[{0, 1}, {1 points, 3}])], {points, 1, 2 × 10^4}]; //
  AbsoluteTiming
```

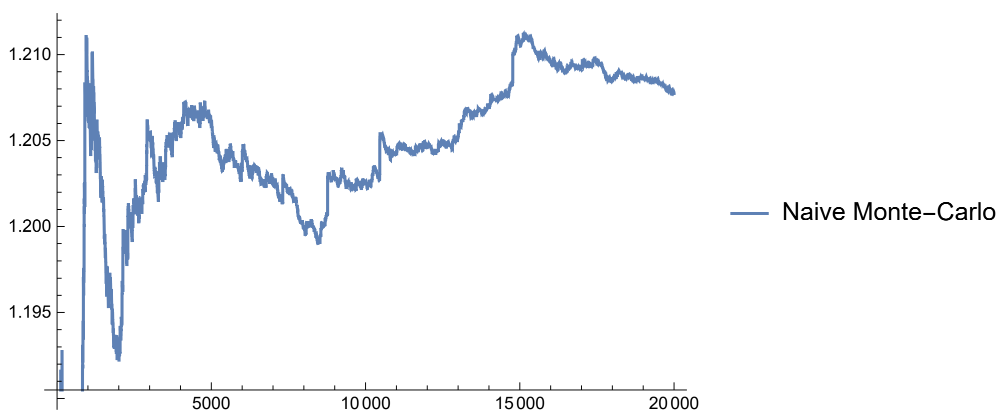
Out[•]=
```
{255.472, Null}
```

In[•]:=
```
psobol = ListLinePlot[sobolquasinaivemc,
   PlotLegends → {"Quasi Monte-Carlo\n (Sobol)"}, PlotStyle → , PlotRange → Full]
```

Out[•]=
```
ListLinePlot[sobolquasinaivemc, PlotLegends → {Quasi Monte-Carlo
 (Sobol)}, PlotStyle → , PlotRange → Full]
```

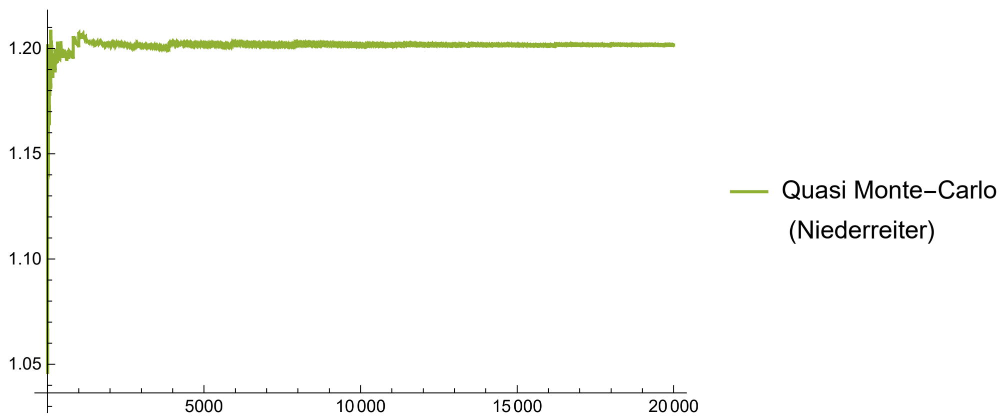
In[•]:=
```
pnaiveapery = ListLinePlot[naiveapery, PlotLegends → {"Naive Monte-Carlo"}]
```

Out[•]=



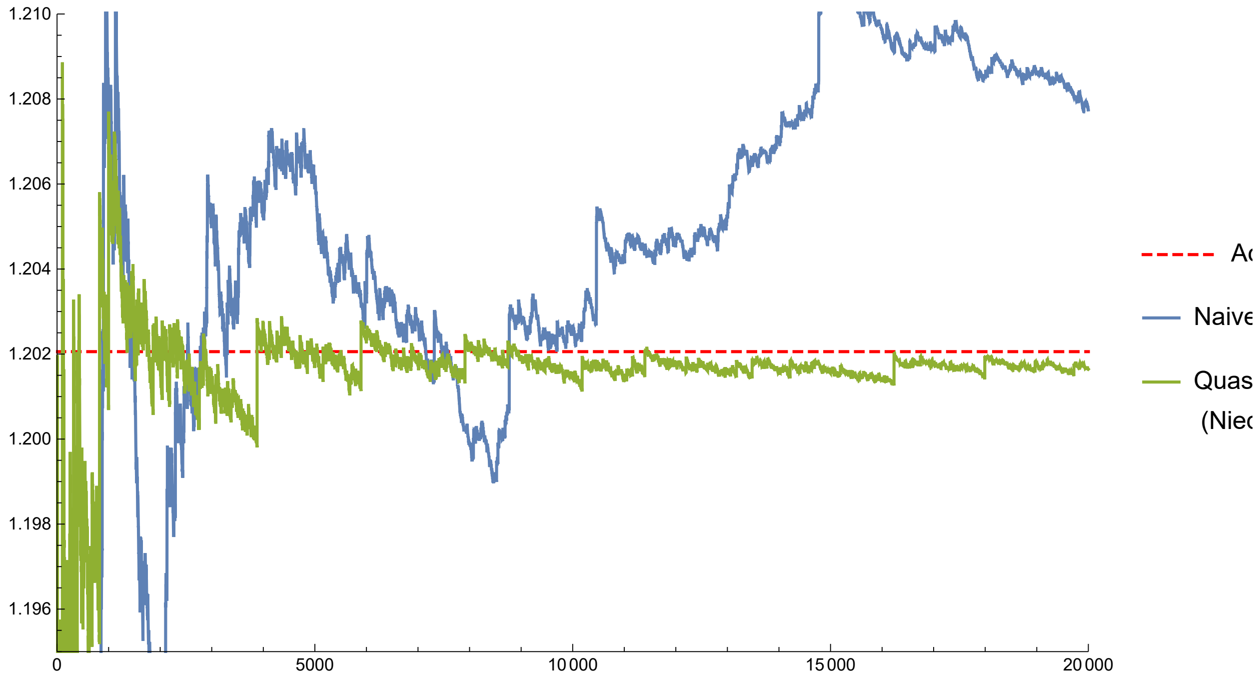
In[]:=
```
quasinaivecv = ListLinePlot[quasinaivemc,
  PlotLegends → {"Quasi Monte-Carlo\n (Niederreiter)"}, PlotStyle → , PlotRange → Full]
```

Out[]=




```
Show[real, pnaiveapery, quasinaivecv, ImageSize → Large]
```

Out[]=



In[]:=
```
quasifinal =
  ParallelTable[SeedRandom[Method → {"MKL", Method → {"Niederreiter", "Dimension" → 3}}];
   1/10^6 Total[int @@@ (RandomReal[{0, 1}, {10^6, 3}])], {10^2}]; // AbsoluteTiming
```

Out[]= {111.401, Null}

In[]:=
```
aperyquasimu = Mean[quasifinal]
aperyquasisd = StandardDeviation[quasifinal]
```

Out[]= 1.20206

Out[]= 0.0000309507

In[•]:=
```
NumberForm[aperyquasimu, 15]
ScientificForm[aperyquasisd]
```
$$\texttt{ScientificForm}\left[100\,\frac{\texttt{aperyquasimu} - \texttt{Zeta[3]}}{\texttt{Zeta[3]}}\right]$$

Out[•]//NumberForm=

1.20205508359899

Out[•]//ScientificForm=

3.09507×10^{-5}

Out[•]//ScientificForm=

-1.51371×10^{-4}