HRI and the SFS

Parameters

Formulae

Get Ne

In[$\circ$]:=
```
U = L u
```
Out[$\circ$]= $L\,u$

In[$\circ$]:=
```
Vm = U s^2
```
Out[$\circ$]= $L\,s^2\,u$

In[$\circ$]:=
```
γ0 = 2 NN s
```
Out[$\circ$]= $2\,NN\,s$

In[$\circ$]:=
```
α0 = 2 NN U
```
Out[$\circ$]= $2\,L\,NN\,u$

In[$\circ$]:=
```
γ = 2 B γ0
```
Out[$\circ$]= $4\,B\,NN\,s$

Gamma0 here too?

In[$\circ$]:=
$$Vg = \frac{U\,s\,(1 - Exp[-\gamma 0])}{1 + \kappa\,Exp[-\gamma 0]}$$

Out[$\circ$]=
$$\frac{\left(1 - e^{-2\,NN\,s}\right) L\,s\,u}{1 + e^{-2\,NN\,s}\,\kappa}$$

Eq 3

In[∘]:=
```
eq3 = (Vg^3 / Vm^2 /. NN → (B NN)) + Log[B]
```

Out[∘]=
$$\frac{\left(1-e^{-2\,B\,NN\,s}\right)^3\,L\,u}{s\left(1+e^{-2\,B\,NN\,s}\,\kappa\right)^3}+\mathrm{Log}[B]$$

In[∘]:=
$$\mathrm{getPbar}[kk\_,\,ss\_,\,NN\_] := \frac{1}{1+kk\,\mathrm{Exp}[-2\,NN\,ss]}$$

In[∘]:=
$$\mathrm{getGammaHat}[kk\_,\,pBar\_] := \mathrm{Log}\left[\frac{kk\,pBar}{1-pBar}\right]$$

Expected p bar

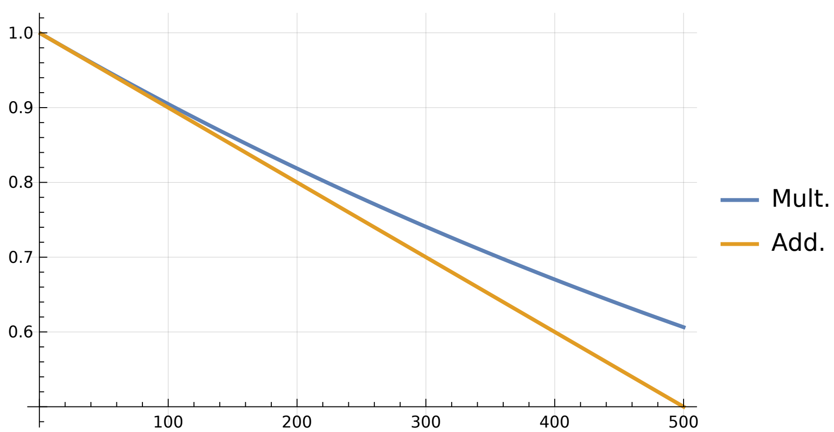
In[∘]:=
```
getPbar[1, 10^-4, 1000] // N
```

Out[∘]= 0.549834

In[∘]:=
```
getGammaHat[1, #] & /@ {0.529, 0.533, 0.537}
```

Out[∘]= {0.11613, 0.132192, 0.148271}

In[∘]:=
```
Plot[{.999^x, 1 - (1 - 0.999) x}, {x, 1, 500},
 GridLines → Automatic, PlotLegends → {"Mult.", "Add."}]
```

Out[∘]=


In[∘]:=
```
getGammaHat[1, 1 - Log[0.955] / Log[0.9999] / 1000]
```

Out[∘]= 0.158667

In[∘]:=
```
getGammaHat[1, 0.536]
```

Out[∘]= 0.14425

In[∘]:=
```
0.9999^550
```

Out[∘]= 0.946483

In[∘]:=
```
params2 = {NN → 1000, s → 10^-3, κ → 1, L → 2500, u → 10^-5}
```

Out[∘]= $\left\{NN \to 1000,\ s \to \frac{1}{1000},\ \kappa \to 1,\ L \to 2500,\ u \to \frac{1}{100\,000}\right\}$

In[∘]:=
```
eq302 = eq3 /. params2
```

Out[∘]= $$\frac{25\left(1 - e^{-2\,B}\right)^3}{\left(1 + e^{-2\,B}\right)^3} + \text{Log}[B]$$

In[∘]:=
```
bfun02[a_] := eq302 /. B → a
```

In[∘]:=
```
Clear[newtonRoot]
```

In[∘]:=
```
newtonRoot[fun : _Symbol | _Function, init_Real, tol_ : 1.*^-100] :=
 Module[{funp}, funp = Derivative[1][fun];
  NestWhile[# - fun[#] / funp[#] &, init, Abs[Subtract[##]] ≥ tol &, 2]
 ]
```

In[∘]:=
```
bsel02 = newtonRoot[bfun02[#] &, 0.5]
```

Out[∘]= 0.359356

In[∘]:=
```
pbarExp02 = getPbar[1, 10^-3, 1000 bsel02]
```

Out[∘]= 0.672323

In[∘]:=
```
getGammaHat[1, pbarExp02]
```

Out[∘]= 0.718712

Get Ne trajectory for neutral sites

In[∘]:=
```
Qt = (Vg / Vm (1 - 1 (1 - Vm / Vg)^(τ+1)))
```

Out[∘]= $$\frac{\left(1 - e^{-2\,NN\,s}\right)\left(1 - \left(1 - \frac{s\left(1+e^{-2\,NN\,s}\,\kappa\right)}{1-e^{-2\,NN\,s}}\right)^{1+\tau}\right)}{s\left(1 + e^{-2\,NN\,s}\,\kappa\right)}$$

In[∘]:=
```
params2
```

Out[∘]= $\left\{NN \to 1000,\ s \to \frac{1}{1000},\ \kappa \to 1,\ L \to 2500,\ u \to \frac{1}{100\,000}\right\}$

In[•]:=
```
Nprime[B_, params_] :=
                              NN
  ─────────────────────────────────────────────── /. params
  Exp[(Vg /. NN → (NN B)) (Qt /. NN → (NN B))^2]
```

In[•]:=
```
np02 = Nprime[bsel02, params2]
```

Out[•]= $1000\, e^{-1.02344\,(1-0.997098^{1+\tau})^2}$

In[•]:=
```
np02 /. τ → # & /@ {0, 500, 1000, 1500, 2000}
```

Out[•]= {999.991, 547.861, 400.585, 368.803, 361.555}

Ne plots

Expressions from Polanski and Kimmel. 2003

The calculations below follow the approach of Polanski, A., and Kimmel, M. (2003). New Explicit Expressions for Relative Frequencies of Single-Nucleotide Polymorphisms With Application to Statistical Inference on Population Growth.” *Genetics* 165:427–436. The equation numbers used by Polanski and Kimmel (2003) are indicated here and in the exponential growth section.

Equation 6 (coefficient used in equations 9 and 10):

In[•]:=
```
Akjn[k_,j_,n_]:= Product[Binomial[l,2],{l,Select[Range[k,n],#≠j&]}] / Product[Binomial[l,2]-Binomial[j,2],{l,Select[Range[k,n],#≠j&]}]
```

Equation 9 (coefficient used in equation 8):

In[•]:=
```
vnj[n_, j_] := Sum[j (j - 1) Akjn[k, j, n] / (k - 1), {k, 2, j}]
```

Equation 10 (coefficient used in equation 8):

In[•]:=
```
wnbj[n_, b_, j_] := Sum[j (j - 1) Binomial[n - k, b - 1] ((n - b - 1)! (b - 1)!) / (n - 1)! Akjn[k, j, n], {k, 2, j}]
```

Exponential example

Doing it on HRI

Smooth

Piecewise approx

*Get extrema from Brian's message
* get inverse with respect to τ
* make τ intervals so that the Δy are equal

In[∘]:=
```
kk = 1
```
Out[∘]=
```
1
```

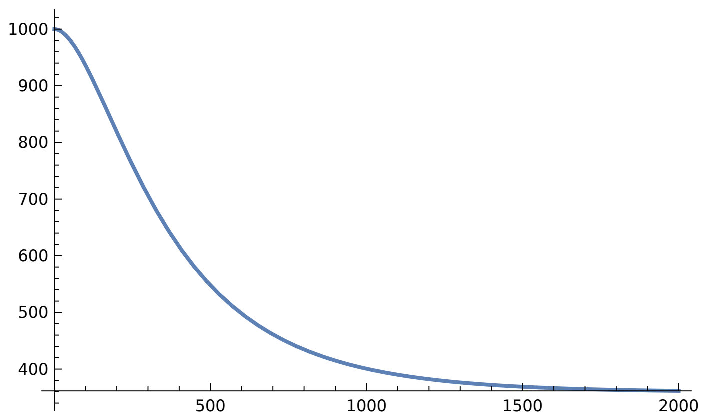
In[∘]:=
```
nPrimeAndLimits[ee_] :=
  {ee, ee /. τ → 0, ee /. τ → ∞}
```

Should s be pos or neg? Either give reasonable trajectories.

In[∘]:=
```
rAndL = nPrimeAndLimits[np02];
```

In[∘]:=
```
rAndL
```
Out[∘]= $\left\{1000\, e^{-1.02344\,(1-0.997098^{1+\tau})^2},\ 999.991,\ 359.356\right\}$

In[∘]:=
```
{x → #, y → rAndL[[1]] /. τ → #} & /@ Range[0, 5000, 500] // N
```
Out[∘]=
```
{{x → 0., y → 999.991}, {x → 500., y → 547.861},
 {x → 1000., y → 400.585}, {x → 1500., y → 368.803}, {x → 2000., y → 361.555},
 {x → 2500., y → 359.87}, {x → 3000., y → 359.476}, {x → 3500., y → 359.384},
 {x → 4000., y → 359.363}, {x → 4500., y → 359.358}, {x → 5000., y → 359.357}}
```

In[∘]:=
```
Plot[rAndL[[1]], {τ, 0, 2000}]
```
Out[∘]=


In[∘]:=
```
getInv[ral_] := Solve[{ral[[1]] == y}, τ]
```

In[∘]:=
```
rAndL
```
Out[∘]=
```
rAndL
```

In[∘]:=
```
inv = getInv[rAndL];
```

Solve: Inverse functions are being used by Solve, so some solutions may not be found; use Reduce for complete solution information.

In[∘]:=
```
inv /. y → 660 // N
```

Out[∘]= {{τ → 347.912}, {τ → -170.656}}

In[∘]:=
```
inv
```

Out[∘]= $\left\{\left\{\tau \to -344.147\,\mathrm{Log}\left[5.03916\times10^{-17}\left(1.99023\times10^{16}-1.\sqrt{3.96103\times10^{32}-3.87031\times10^{32}\,\mathrm{Log}[0.002782754254592482\,y]}\right)\right]\right\},\left\{\tau \to -344.147\,\mathrm{Log}\left[5.03915672988227\times10^{-17}\left(1.99023371966983\times10^{16}+\sqrt{3.9610302589108\times10^{32}-3.8703057397061\times10^{32}\,\mathrm{Log}[0.002782754254592482\,y]}\right)\right]\right\}\right\}$

In[∘]:=
```
inv2 = inv[[1, 1, 2]]
```

Out[∘]= $-344.147\,\mathrm{Log}\left[5.03916\times10^{-17}\left(1.99023\times10^{16}-1.\sqrt{3.96103\times10^{32}-3.87031\times10^{32}\,\mathrm{Log}[0.002782754254592482\,y]}\right)\right]$

In[∘]:=
```
inv2 /. y → 660 // N
```

Out[∘]= 347.912

Breaks, even x steps

In[∘]:=
```
getXs[ral_, n_, inv_] := Module[{fiveP = (ral[[2]] - ral[[3]]) / 20 + ral[[3]]},
  Subdivide[0, inv /. y → fiveP, n]
 ]
```

In[∘]:=
```
xs = getXs[rAndL, 10, inv2];
```

In[∘]:=
```
xs // N
```

Out[∘]= {0., 108.491, 216.981, 325.472, 433.962, 542.453, 650.944, 759.434, 867.925, 976.415, 1084.91}

In[∘]:=
```
{1, 2, 3, 4, 5} + 1 / 2
```

Out[∘]= $\left\{\frac{3}{2}, \frac{5}{2}, \frac{7}{2}, \frac{9}{2}, \frac{11}{2}\right\}$

In[∘]:=
```
getYs[xs_, ral_] := Module[{dd = xs[[2]] - xs[[1]]},
  Join[{ral[[2]]}, ral[[1]] /. τ → # & /@ ((xs // Rest // Most) + dd / 2), {ral[[3]]}]
 ]
```

In[∘]:=
```
ys = getYs[xs, rAndL];
```

In[∘]:=
```
ys // N
```

Out[∘]=
```
{999.991, 863.558, 736.538, 632.335, 554.858,
 499.205, 459.62, 431.468, 411.38, 396.988, 359.356}
```

In[∘]:=
```
xy1 = {xs, ys} // N
```

Out[∘]=
```
{{0., 108.491, 216.981, 325.472, 433.962, 542.453, 650.944,
  759.434, 867.925, 976.415, 1084.91}, {999.991, 863.558, 736.538,
  632.335, 554.858, 499.205, 459.62, 431.468, 411.38, 396.988, 359.356}}
```

Breaks, even y steps

In[∘]:=
```
getEquiDistYs[ral_, n_, inv_] := Module[{
    ys = Join[{ral[[2]]}, ral[[2]] - Range[1, n - 1] (ral[[2]] - ral[[3]]) / n, {ral[[3]]}]},
   {Join[{0}, inv /. y → # & /@ (Most[ys] - ((ral[[2]] - ral[[3]]) / n / 2))], ys}
  ]
```

In[∘]:=
```
xy2 = getEquiDistYs[rAndL, 10, inv2] // N
```

Out[∘]=
```
{{0., 66.6165, 128.805, 182.327, 236.226, 294.656, 361.782,
  443.819, 552.733, 719.106, 1084.91}, {999.991, 935.928, 871.864,
  807.801, 743.737, 679.674, 615.61, 551.547, 487.483, 423.42, 359.356}}
```

Return to function

In[∘]:=
```
Nprime01p = D[rAndL[[1]], τ]
```

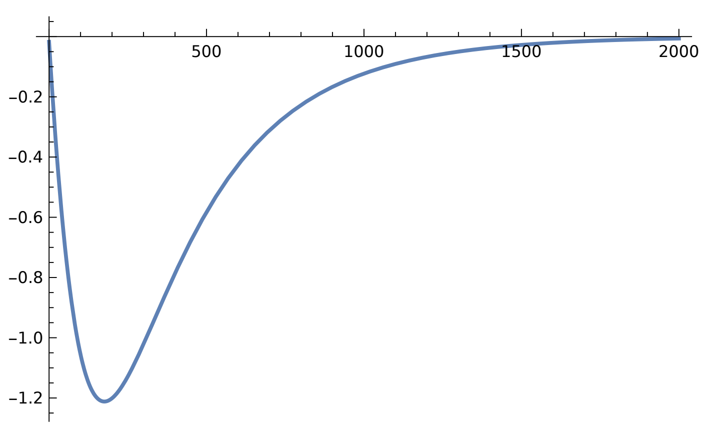
Out[∘]= $-5.9477 \times 0.997098^{1+\tau}\left(1-0.997098^{1+\tau}\right) e^{-1.02344\left(1-0.997098^{1+\tau}\right)^2}$

In[∘]:=
```
Nprime01p /. τ → -10 // N
```

Out[∘]=
```
0.161653
```

In[∘]:=
```
Plot[Nprime01p /. τ → t, {t, 0, 2000}]
```

Out[∘]=



Assuming Nprime is monotonously decreasing, take values as x=0 and limit for x→∞

In[∘]:=
```
makePiecewiseTraj[xys_] :=
 Module[{pRules0 = {n, x < a} /. a → #1 /. n → #2 & @@@ Transpose[{Rest[xys[[1]]], Most[xys[[2]]]}]},
  Piecewise[Join[pRules0, {{Last[xys[[2]]], x > Last[xys[[1]]]}}], {{0, True}}] // N]
```

In[∘]:=
```
pwNe1 = makePiecewiseTraj[xy1]
```

Out[∘]=

999.991	x < 108.491
863.558	x < 216.981
736.538	x < 325.472
632.335	x < 433.962
554.858	x < 542.453
499.205	x < 650.944
459.62	x < 759.434
431.468	x < 867.925
411.38	x < 976.415
396.988	x < 1084.91
359.356	x > 1084.91
0.	True

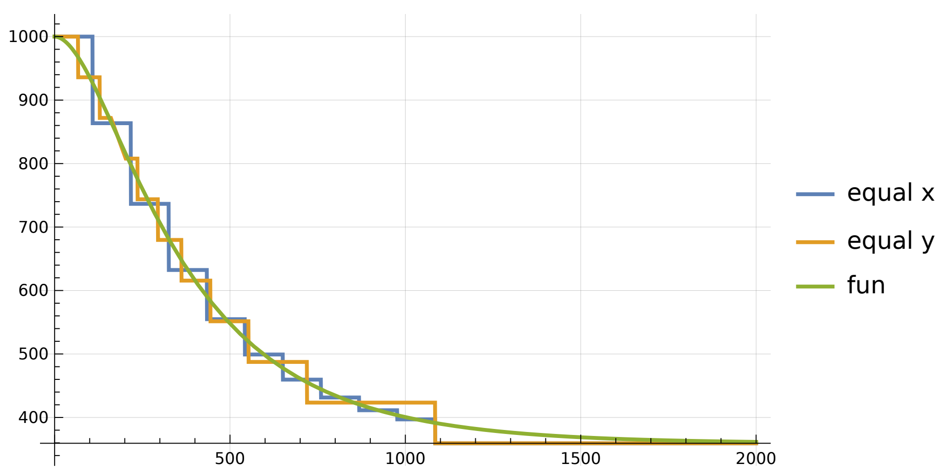
In[∘]:=
```
pwNe2 = makePiecewiseTraj[xy2]
```

Out[∘]=

999.991	x < 66.6165
935.928	x < 128.805
871.864	x < 182.327
807.801	x < 236.226
743.737	x < 294.656
679.674	x < 361.782
615.61	x < 443.819
551.547	x < 552.733
487.483	x < 719.106
423.42	x < 1084.91
359.356	x > 1084.91
0.	True

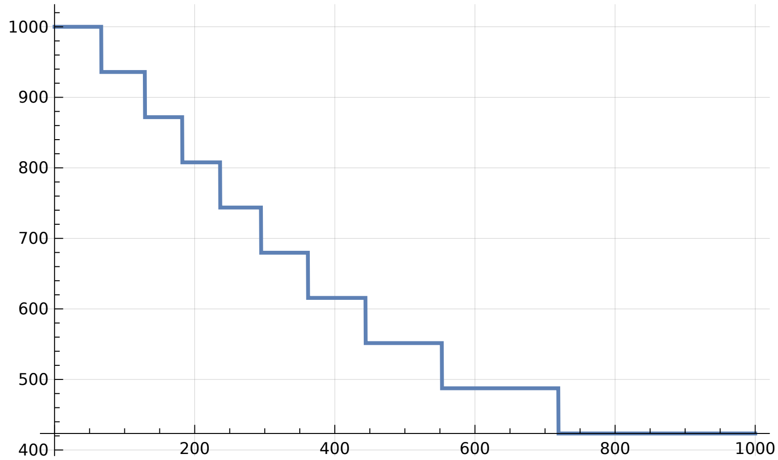
In[∘]:=
```
Plot[{pwNe1 /. x → t, pwNe2 /. x → t, rAndL[[1]] /. τ → t}, {t, 0, 2000},
 GridLines → Automatic, ImageSize → 400, PlotLegends → {"equal x", "equal y", "fun"}]
```

Out[∘]=



In[∘]:=
```
Plot[pwNe2 /. x → t, {t, 0, 1000}, GridLines → Automatic, ImageSize → 400]
```

Out[∘]=



In[∘]:=
```
pwNe = pwNe2
```

Out[∘]=

999.991	x < 66.6165
935.928	x < 128.805
871.864	x < 182.327
807.801	x < 236.226
743.737	x < 294.656
679.674	x < 361.782
615.61	x < 443.819
551.547	x < 552.733
487.483	x < 719.106
423.42	x < 1084.91
359.356	x > 1084.91
0.	True

In[∘]:=
```
pwNe /. x → 1000 // N
```

Out[∘]= 423.42

In[∘]:=

$$\text{qjtHriPw01[j\_]:=}\frac{\text{Binomial[j,2]}}{\text{pwNe}}\text{Exp}\left[-\text{Integrate}\left[\frac{\text{Binomial[j,2]}}{\text{pwNe/.x}\to\sigma},\{\sigma,0,x\},\text{Assumptions}\to x\in\mathbb{R}_{>0}\right]\right]$$

In[∘]:=

$$\text{qjtHriPw01smooth[j\_]:=}\frac{\text{Binomial[j,2]}}{\text{np02}}\text{Exp}\left[-\text{Integrate}\left[\frac{\text{Binomial[j,2]}}{\text{np02/.}\tau\to\sigma},\{\sigma,0,\tau\},\text{Assumptions}\to \tau\in\mathbb{R}_{>0}\right]\right]$$

Prob of coalescence for a pair of lineages at given time:

In[∘]:=
```
qjtHriPw01[2] /. x → 600 // N // AbsoluteTiming
```

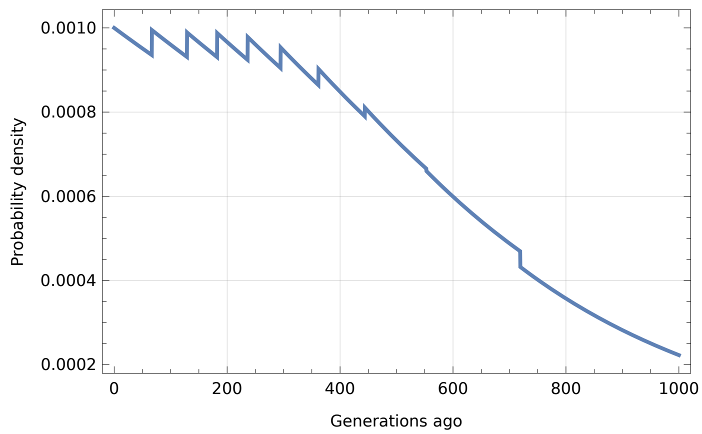
Out[∘]= {1.20703, 0.000862697}

In[∘]:=
```
qjtHriPw01smooth[2] /. τ → 600 // N // AbsoluteTiming
```

Out[∘]= {3.46896, 0.000846223}

In[∘]:=
```
Plot[qjtHriPw01[2], {x, 0, 1000}, Frame → True,
 FrameLabel → {"Generations ago", "Probability density"}, GridLines → Automatic]
```

Out[∘]=



Equation 3 (the expectation of equation 4):

In[∘]:=
```
ejHriPw[j_]:=Integrate[x qjtHriPw01[j],{x,0,∞}, Assumptions→x∈Reals]
```

In[∘]:=
```
ejHriPwSmooth[j_] := Integrate[τ qjtHriPw01smooth[j], {τ, 0, ∞}, Assumptions → τ ∈ Reals]
```

Expected coalescence time for a pair of lineages.

2 steps 1.4s, 2 steps 3.2s, 4 steps 13s, 5 steps 62s

Is is much faster to use Integrate[] with Assumptions→x∈Reals than PiecewiseIntegrate[]: 5steps 10s, 8steps 18s,9 steps 21s

In[∘]:=
```
a01 = (ejHriPw[2] // AbsoluteTiming)
```

Out[∘]=
```
{12.9598, 599.115}
```

In[∘]:=
```
a01smooth = (ejHriPwSmooth[2] // AbsoluteTiming)
```

In[∘]:=
```
a01[[2]]
```

Out[∘]=
```
599.115
```

Equation 8 (an expression for the probability to see *b* derived alleles in a sample of *n*):

In[∘]:=
```
qnbHriPw[n_, b_] := Sum[ejHriPw[j] × wnbj[n, b, j], {j, 2, n}] / Sum[ejHriPw[j] × vnj[n, j], {j, 2, n}]
```

In[∘]:=
```
AbsoluteTiming[n12b1 = qnbHriPw[12, 1]]
```

Out[∘]=
```
{220.314, 0.413224}
```

Equal x steps

Out[∘]=
```
{253.907, 0.340639}
```

2 steps 30s, 3 steps 70s, 4 steps 263s, 5 steps 1334s (22min)
With Assumptions x ∈R:
5steps 206s, 6 steps 270, 7 steps 355, 8 steps 394s, 9 steps 435s

Duration similar irrespective of b.

In[∘]:=
```
n12All = AbsoluteTiming[n12b1 = qnbHriPw[12, #]] & /@ Range[1, 11]
```

Out[∘]=
```
{{212.535, 0.413224}, {177.512, 0.177479}, {176.77, 0.105818}, {175.299, 0.0728697},
 {174.444, 0.0545042}, {173.993, 0.0430248}, {173.701, 0.035276},
 {173.393, 0.0297467}, {173.1, 0.0256311}, {175.457, 0.0224646}, {174.973, 0.0199623}}
```

In[∘]:=
```
n10All = AbsoluteTiming[n10b1 = qnbHriPw[10, #]] & /@ Range[1, 9]
```

Out[∘]=
```
{{174.405, 0.435723}, {145.477, 0.184826}, {144.681, 0.109549},
 {144.006, 0.0752579}, {142.8, 0.0562673}, {142.455, 0.0444489},
 {141.802, 0.0364934}, {141.776, 0.0308254}, {141.513, 0.0266096}}
```

In[∘]:=
```
Export["sfs10.txt", n10All]
```

Out[∘]=
```
sfs10.txt
```

In[∘]:=
```
n20All = AbsoluteTiming[n20b1 = qnbHriPw[20, #]] & /@ Range[1, 19]
```

Out[∘]=
```
{{335.366, 0.360495}, {301.575, 0.160202}, {300.78, 0.0973517}, {300.003, 0.0677397},
 {301.037, 0.0509267}, {301.384, 0.0402702}, {300.364, 0.0330018}, {300.533, 0.0277762},
 {298.902, 0.0238663}, {298.986, 0.0208479}, {300.504, 0.0184579}, {300.564, 0.0165256},
 {298.428, 0.0149356}, {298.215, 0.0136076}, {298.062, 0.0124839}, {298.067, 0.0115224},
 {297.542, 0.0106914}, {297.707, 0.00996697}, {297.977, 0.00933037}}
```

In[∘]:=
```
Export["sfs20.txt", n20All]
```

Out[∘]=
```
sfs20.txt
```

In[∘]:=
```
n40All = AbsoluteTiming[n40b1 = qnbHriPw[40, #]] & /@ Range[1, 39]
```

Out[∘]=
```
{{690.693, 0.304833}, {623.161, 0.141069}, {622.787, 0.0880718}, {622.684, 0.0624239},
 {622.817, 0.0475338}, {623.176, 0.0379199}, {623.621, 0.0312601}, {622.813, 0.0264088},
 {622.994, 0.0227385}, {622.536, 0.0198786}, {622.167, 0.0175965}, {620.71, 0.0157396},
 {620.431, 0.0142037}, {620.423, 0.0129155}, {621.37, 0.0118219}, {620.64, 0.0108838},
 {620.362, 0.0100715}, {620.451, 0.00936248}, {620.599, 0.00873897}, {620.424, 0.00818704},
 {621.924, 0.00769554}, {620.63, 0.00725549}, {620.187, 0.00685953}, {620.273, 0.00650162},
 {620.931, 0.00617676}, {620.491, 0.00588075}, {620.321, 0.00561005}, {620.584, 0.00536169},
 {620.562, 0.00513311}, {620.536, 0.00492213}, {620.482, 0.00472686}, {620.47, 0.00454568},
 {620.657, 0.00437717}, {620.099, 0.00422009}, {620.359, 0.00407336},
 {620.797, 0.00393602}, {620.367, 0.00380723}, {618.06, 0.00368623}, {618.464, 0.00357237}}
```

In[∘]:=
```
Export["sfs40.txt", n40All]
```

Out[∘]=
```
sfs40.txt
```

In[∘]:= n80All = AbsoluteTiming[n80b1 = qnbHriPw[80, #]] & /@ Range[1, 79]

Out[∘]= {{1441.72, 0.261063}, {1321.75, 0.124573}, {1318.68, 0.0796208}, {1319.44, 0.0574755}, {1317.49, 0.0444041}, {1319.97, 0.0358379}, {1320.13, 0.0298246}, {1315.44, 0.0253922}, {1317.17, 0.0220031}, {1315.77, 0.0193367}, {1315.44, 0.0171904}, {1316.51, 0.0154299}, {1316.1, 0.013963}, {1315.06, 0.0127244}, {1313.84, 0.0116664}, {1312.99, 0.0107537}, {1313.22, 0.00995939}, {1312.68, 0.00926276}, {1313.66, 0.00864755}, {1312.47, 0.00810088}, {1312.83, 0.00761238}, {1313.48, 0.00717364}, {1313.07, 0.00677776}, {1316.55, 0.00641904}, {1313.55, 0.00609271}, {1313.1, 0.00579477}, {1313.52, 0.00552185}, {1313.02, 0.00527107}, {1313.29, 0.00503996}, {1316.72, 0.0048264}, {1315.21, 0.00462857}, {1314.11, 0.00444486}, {1313.49, 0.00427389}, {1313.45, 0.00411443}, {1312.96, 0.00396542}, {1312.44, 0.00382591}, {1312.89, 0.00369506}, {1312.47, 0.00357212}, {1313.2, 0.00345644}, {1312.89, 0.00334742}, {1313.04, 0.00324452}, {1314.33, 0.00314726}, {1313.1, 0.00305522}, {1311.98, 0.00296801}, {1312.64, 0.00288527}, {1312.77, 0.00280667}, {1312.49, 0.00273194}, {1312.92, 0.00266081}, {1314.3, 0.00259303}, {1313.9, 0.00252838}, {1312.79, 0.00246666}, {1312.69, 0.00240768}, {1313.16, 0.00235127}, {1312.77, 0.00229727}, {1312.56, 0.00224555}, {1312.97, 0.00219595}, {1312.77, 0.00214837}, {1308.22, 0.00210268}, {1302.04, 0.00205879}, {1305.87, 0.00201658}, {1312.32, 0.00197597}, {1312.35, 0.00193687}, {1312.18, 0.0018992}, {1312.51, 0.00186289}, {1312.51, 0.00182787}, {1312.9, 0.00179407}, {1312.46, 0.00176144}, {1313.84, 0.00172991}, {1314.26, 0.00169944}, {1313.64, 0.00166997}, {1313.44, 0.00164145}, {1312.77, 0.00161385}, {1311.87, 0.00158711}, {1312.56, 0.00156121}, {1313.5, 0.0015361}, {1312.8, 0.00151175}, {1311.83, 0.00148813}, {1312.78, 0.0014652}, {1311.33, 0.00144294}}

In[∘]:= Export["sfs80.txt", n80All]

Out[∘]= sfs80.txt

In[∘]:= sfs01 = #[[2]] & /@ n12All

Out[∘]= {0.413224, 0.177479, 0.105818, 0.0728697, 0.0545042, 0.0430248, 0.035276, 0.0297467, 0.0256311, 0.0224646, 0.0199623}

In[∘]:= #[[2]] & /@ n12All // Total

Out[∘]= 1.

Expected π is ejHriPw[2] × 2 × μ

In[∘]:= a01[[2]]

Out[∘]= 599.115

In[∘]:= piUncond[T_, u_, κ_] := 2 T 2 u u κ / (u + u κ)

In[∘]:= piUncond01 = piUncond[a01[[2]], 10^{-5}, 1]

Out[∘]= 0.0119823

In[∘]:= (1 + kk) 10^{-5}

Out[∘]= $\frac{1 + kk}{100\,000}$

In[∘]:=
```
condPi[sfs_] := Module[{n = Length[sfs] + 1},
  Sum[2 i (n - i) sfs[[i]], {i, 1, n - 1}] / n^2
 ]
```

$$\sum_{i=1}^{n-1} 2\, i\, (n-i)\, sfs[[i]] \Big/ n^2$$

In[∘]:=
```
condPi01 = condPi[sfs01]
```
Out[∘]=
```
0.281777
```

In[∘]:=
```
an[n_] := Sum[1 / i, {i, 1, n - 1}]
```

$$an[n\_] := \sum_{i=1}^{n-1} 1 / i$$

In[∘]:=
```
an[12]
```
Out[∘]= $\frac{83711}{27720}$

In[∘]:=
```
an[Length[sfs01] + 1]
```
Out[∘]= $\frac{83711}{27720}$

$\Delta\theta$

In[∘]:=
```
1 - condPi[sfs01] × an[Length[sfs01] + 1]
```
Out[∘]=
```
0.149067
```

In[∘]:=
```
thetaw01 = piUncond01 / condPi01 / an[12]
```
Out[∘]=
```
0.0140814
```

In[∘]:=
```
1 - piUncond01 / thetaw01
```
Out[∘]=
```
0.149067
```

In[∘]:=
```
Length[sfs01]
```
Out[∘]=
```
11
```