Variational Monte Carlo method

Clear all symbols from previous evaluations to avoid problems

In[65]:=
```
Clear["Global`*"]
```

1D harmonic oscillator

Trial function which is exact for $\alpha = 0.5$:

In[66]:=
```
Clear[α];
ψ[x_] := Exp[-α * x * x];
exactEnergy = Assuming[α > 0,
  Integrate[ψ[x] * (-D[ψ[x], x, x] + x^2 ψ[x]) / 2, {x, -Infinity, Infinity}] /
   Integrate[ψ[x]^2, {x, -Infinity, Infinity}]]
```

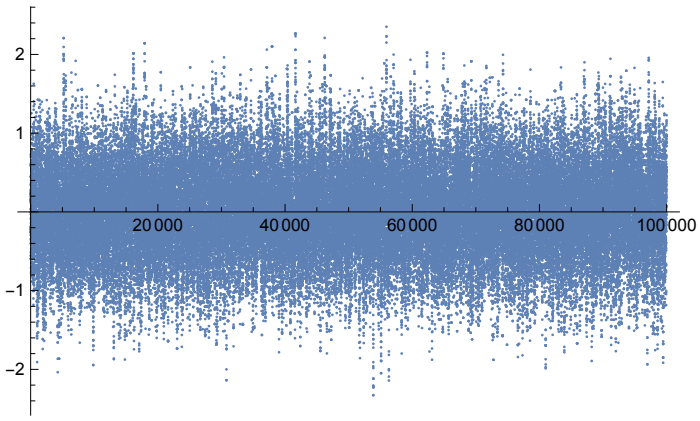
Out[68]= $\frac{1 + 4\alpha^2}{8\alpha}$

In[69]:=
```
n = 100000;
a = -8.0; b = 8.0;
Do[
  α = 0.2 + 0.05 * i;
  ρ[x_] := ψ[x]^2;
  x0 = 0.5; δ = 0.5;
  points = MyRandomMetropolis1D[n, ρ, a, b, x0, δ, 1];
  ϵ[x_] := (α + x * x * (0.5 - 2.0 * α * α));
  {energy, error} = MyMCIntegration[points, ϵ];
  Print["Alpha = ", α, ", energy = ", energy,
   ", exact energy = ", exactEnergy, ", probable error = ", error],
  {i, 0, 11}
 ];
```

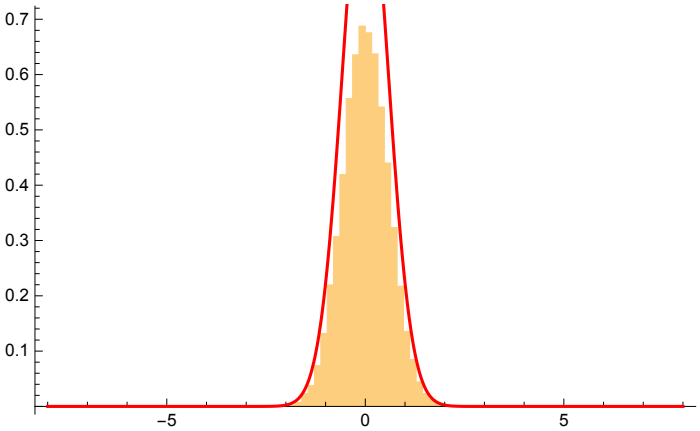
```
Alpha = 0.2, energy = 0.7256289487, exact energy = 0.725, probable error = 0.00237847454

Alpha = 0.25, energy = 0.6111412671, exact energy = 0.625, probable error = 0.001571257532

Alpha = 0.3, energy = 0.5661793475
 , exact energy = 0.5666666667, probable error = 0.001205489716

Alpha = 0.35, energy = 0.5357004511
 , exact energy = 0.5321428571, probable error = 0.0008447179975

Alpha = 0.4, energy = 0.5117683384, exact energy = 0.5125, probable error = 0.0005010676344

Alpha = 0.45, energy = 0.5025310255
 , exact energy = 0.5027777778, probable error = 0.0002330726097

Alpha = 0.5, energy = 0.5, exact energy = 0.5, probable error = 0.

Alpha = 0.55, energy = 0.5019462794
 , exact energy = 0.5022727273, probable error = 0.0002123680068

Alpha = 0.6, energy = 0.5088437079
 , exact energy = 0.5083333333, probable error = 0.0004123001684

Alpha = 0.65, energy = 0.517692127
 , exact energy = 0.5173076923, probable error = 0.0005892616549

Alpha = 0.7, energy = 0.5279182723
 , exact energy = 0.5285714286, probable error = 0.000759257753

Alpha = 0.75, energy = 0.5433784074
 , exact energy = 0.5416666667, probable error = 0.0009271982272
```

In[72]:=
```
ListPlot[points[[1 ;; n]]]
Histogram[points[[1 ;; n]], {a, b, (b - a) / 100}, "PDF",
 Epilog → First@Plot[ρ[x], {x, a, b}, PlotRange → All, PlotStyle → Red]]
```

Out[72]=



Out[73]=



Hydrogen atom

Trial function which is exact for α = 1.0:

In[74]:=
```
Clear[α];
ψ[x_, y_, z_, α_] := Exp[-α * Sqrt[x^2 + y^2 + z^2]];
ϵloc[x_, y_, z_, α_] := Evaluate[
    (-1/2 (D[ψ[x, y, z, α], x, x] + D[ψ[x, y, z, α], y, y] + D[ψ[x, y, z, α], z, z]) -
        ψ[x, y, z, α] / Sqrt[x^2 + y^2 + z^2]) / ψ[x, y, z, α] // Simplify];
ϵloc[
 x,
 y,
 z,
 α]
```

Out[77]= $$-\frac{2 - 2\,\alpha + \sqrt{x^2 + y^2 + z^2}\;\alpha^2}{2\sqrt{x^2 + y^2 + z^2}}$$

In[78]:=
```
n = 100000;
dim = 3;
xmax = 8.0; xmin = -xmax;
a = {xmin, xmin, xmin}; b = {xmax, xmax, xmax};
Do[
  αα = 0.8 + 0.1 * i;
  ρ[x_] := ψ[x[[1]], x[[2]], x[[3]], αα]^2;
  (* Print[{N[ρ[a]],N[ρ[b]]}]; *)
  x0 = b / 10.0; δ = b / 10;
  points = MyRandomMetropolisND[n, dim, ρ, a, b, x0, δ, 1];
  ϵ[x_] := ϵloc[x[[1]], x[[2]], x[[3]], αα];
  {energy, error} = MyMCIntegration[points, ϵ];
  Print["Alpha = ", αα, ", energy = ", energy, ", probable error = ", error],
  {i, 0, 4}
 ];
```

```
Alpha = 0.8, energy = -0.480038945, probable error = 0.000468713139

Alpha = 0.9, energy = -0.4941179581, probable error = 0.000291374451

Alpha = 1., energy = -0.5, probable error = 0.

Alpha = 1.1, energy = -0.4952097905, probable error = 0.0003565619738

Alpha = 1.2, energy = -0.4815355579, probable error = 0.0006956720153
```

Trial function consisting of one Gaussian function:

In[83]:=
```
Clear[α];
ψ[x_, y_, z_, α_] := Exp[-α * (x^2 + y^2 + z^2)];
ϵloc[x_, y_, z_, α_] := Evaluate[
    (-1/2 (D[ψ[x, y, z, α], x, x] + D[ψ[x, y, z, α], y, y] + D[ψ[x, y, z, α], z, z]) -
        ψ[x, y, z, α] / Sqrt[x^2 + y^2 + z^2]) / ψ[x, y, z, α] // Simplify];
ϵloc[
 x,
 y,
 z,
 α]
```

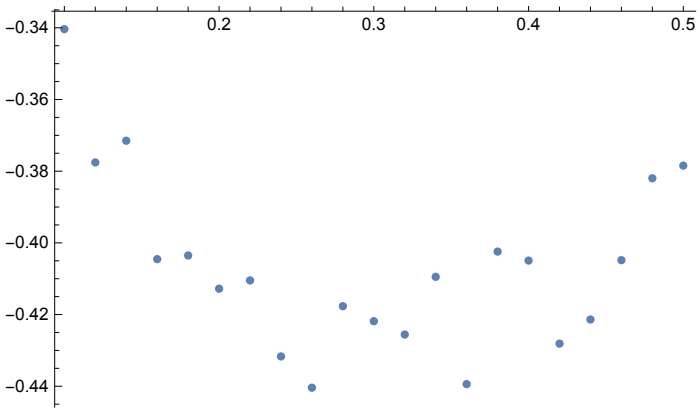
Out[86]= $-\frac{1}{\sqrt{x^2+y^2+z^2}}+3\,\alpha-2\,\left(x^2+y^2+z^2\right)\,\alpha^2$

In[87]:=
```
n = 10000;
dim = 3;
xmax = 8.0; xmin = -xmax;
a = {xmin, xmin, xmin}; b = {xmax, xmax, xmax};
nalpha = 21;
alpha = ConstantArray[0.0, nalpha];
energy = ConstantArray[0.0, nalpha];
Do[
  alpha[[i]] = 0.1 + 0.02 * (i - 1);
  ρ[x_] := ψ[x[[1]], x[[2]], x[[3]], alpha[[i]]]^2;
  (* Print[{N[ρ[a]],N[ρ[b]]}]; *)
  x0 = b / 10.0; δ = b / 10;
  points = MyRandomMetropolisND[n, dim, ρ, a, b, x0, δ, 1];
  ϵ[x_] := ϵloc[x[[1]], x[[2]], x[[3]], alpha[[i]]];
  {energy[[i]], error} = MyMCIntegration[points, ϵ];
  Print["Alpha = ", alpha[[i]],
   ", energy = ", energy[[i]], ", probable error = ", error],
  {i, 1, nalpha}
 ];
```

```
Alpha = 0.1, energy = -0.3403679963, probable error = 0.002810984699
Alpha = 0.12, energy = -0.3775572575, probable error = 0.003361324811
Alpha = 0.14, energy = -0.3715083291, probable error = 0.003402584724
Alpha = 0.16, energy = -0.4045365708, probable error = 0.003895232562
Alpha = 0.18, energy = -0.4035146429, probable error = 0.004041721821
Alpha = 0.2, energy = -0.4127865401, probable error = 0.004428343077
Alpha = 0.22, energy = -0.4104797925, probable error = 0.004052632722
Alpha = 0.24, energy = -0.4316810108, probable error = 0.004751976369
Alpha = 0.26, energy = -0.4404050864, probable error = 0.004792898538
Alpha = 0.28, energy = -0.4176633914, probable error = 0.005056820062
Alpha = 0.3, energy = -0.4218427564, probable error = 0.006522465869
Alpha = 0.32, energy = -0.4255784641, probable error = 0.006262036174
Alpha = 0.34, energy = -0.4094661506, probable error = 0.006726670033
Alpha = 0.36, energy = -0.4393933803, probable error = 0.006680262939
Alpha = 0.38, energy = -0.4024271117, probable error = 0.006130119641
Alpha = 0.4, energy = -0.4049304744, probable error = 0.005770464752
Alpha = 0.42, energy = -0.4281079796, probable error = 0.006822933002
Alpha = 0.44, energy = -0.4213651884, probable error = 0.00660637154
Alpha = 0.46, energy = -0.4048260779, probable error = 0.006675575244
Alpha = 0.48, energy = -0.3819763973, probable error = 0.005684973334
Alpha = 0.5, energy = -0.3784460294, probable error = 0.006308525427
```

In[95]:=
```
plot1 = ListPlot[Transpose[{alpha, energy}]]
```

Out[95]=

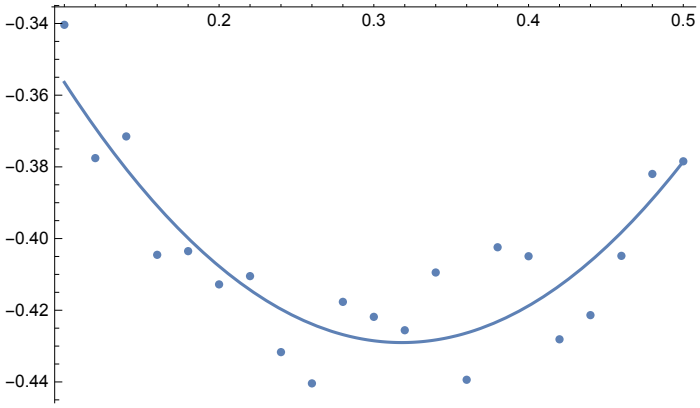


Finding minimum using the least-square method:

In[96]:=
```
fToFit[α_, par_] := par[[1]] + par[[2]] * (α - par[[3]]) ^2;
f[par_] := Module[
   {i, fx, squares},
   squares = 0.0;
   Do[
    fx = fToFit[alpha[[i]], par] - energy[[i]];
    squares += fx * fx,
    {i, 1, nalpha}
   ];
   Return[squares]
  ];
{minpar, value, niter} =
 MyMinimization[{-0.42, 1.0, 0.3}, {0.01, 0.01, 0.01}, f, 10^(-6)]
plot2 = Plot[fToFit[x, minpar], {x, 0.1, 0.5}];
Show[plot1, plot2]
```

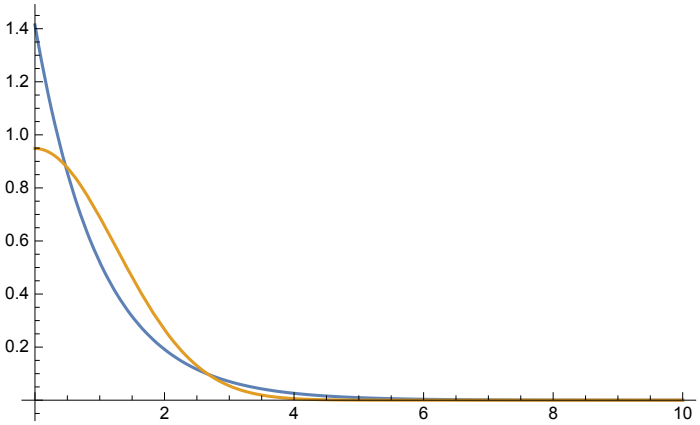
Out[98]= {{-0.4290154322, 1.52612252, 0.3180906114}, 0.002896405792, 110}

Out[100]=



In[101]:=
```
alphaSTO = 1;
alphaGTO = minpar[[3]];
norm1 = 1 / Sqrt[Integrate[Exp[-2 * alphaSTO * r], {r, 0, Infinity}]];
norm2 = 1 / Sqrt[Integrate[Exp[-2 * alphaGTO * r^2], {r, 0, Infinity}]];
Plot[{Exp[-alphaSTO * r] * norm1, Exp[-alphaGTO * r^2] * norm2},
 {r, 0, 10}, PlotRange → All]
```

Out[105]=



Trial function consisting of two Gaussian function with minimization:

In[106]:=
```
ψ[x_, y_, z_, α1_, α2_, c2_] :=
  Exp[-α1 * (x^2 + y^2 + z^2)] + c2 * Exp[-α2 * (x^2 + y^2 + z^2)];
ϵloc[x_, y_, z_, α1_, α2_, c2_] :=
  Evaluate[(-1/2 (D[ψ[x, y, z, α1, α2, c2], x, x] + D[ψ[x, y, z, α1, α2, c2], y, y] +
          D[ψ[x, y, z, α1, α2, c2], z, z]) - ψ[x, y, z, α1, α2, c2] /
        Sqrt[x^2 + y^2 + z^2]) / ψ[x, y, z, α1, α2, c2] // Simplify];
ϵloc[
 x,
 y,
 z,
 α1,
 α2,
 c2]
```

Out[108]=
$$\left(e^{(x^2+y^2+z^2)\,\alpha 2}\left(-1+3\sqrt{x^2+y^2+z^2}\;\alpha 1-2\left(x^2+y^2+z^2\right)^{3/2}\alpha 1^2\right)-c2\,e^{(x^2+y^2+z^2)\,\alpha 1}\left(1-3\sqrt{x^2+y^2+z^2}\;\alpha 2+2\left(x^2+y^2+z^2\right)^{3/2}\alpha 2^2\right)\right)\Big/\left(\left(c2\,e^{(x^2+y^2+z^2)\,\alpha 1}+e^{(x^2+y^2+z^2)\,\alpha 2}\right)\sqrt{x^2+y^2+z^2}\right)$$

In[109]:=
```
(* par1 = α1, par2 = α2, par3 = c2 *)
minFunc[par_] :=
 Module[{n, dim, xmin, xmax, a, b, ρ, x0, δ, points, ϵ, energy},
   n = 100000;
   dim = 3;
   xmax = 8.0; xmin = -xmax;
   a = {xmin, xmin, xmin}; b = {xmax, xmax, xmax};
   ρ[x_] := ψ[x[[1]], x[[2]], x[[3]], par[[1]], par[[2]], par[[3]]]^2;
   x0 = b / 10.0; δ = b / 10;
   points = MyRandomMetropolisND[n, dim, ρ, a, b, x0, δ, 1];
   ϵ[x_] := ϵloc[x[[1]], x[[2]], x[[3]], par[[1]], par[[2]], par[[3]]];
   energy = MyMCIntegration[points, ϵ];
   Print["Parameters = ", par,
    ", energy = ", energy[[1]], ", probable error = ", energy[[2]]];
   Return[energy[[1]]];
]];
```

In[110]:=
```
minFunc[{0.25, 1.50, 1.00}]
```

```
Parameters = {0.25, 1.5, 1.}, energy = -0.4825805424, probable error = 0.00153249269
```

Out[110]= -0.4825805424

In[111]:=
```
MyMinimization[{0.5, 2.00, 0.50}, {0.1, 0.1, 0.1}, minFunc, 0.01]
```

```
Parameters = {0.5, 2., 0.5}, energy = -0.3914571247, probable error = 0.002097209318

Parameters = {0.6, 2., 0.5}, energy = -0.3223982913, probable error = 0.002253844974

Parameters = {0.5, 2.1, 0.5}, energy = -0.3970669436, probable error = 0.002713159791

Parameters = {0.5, 2., 0.6}, energy = -0.3782106067, probable error = 0.001954082281

Parameters = {0.4, 2.066666667, 0.5666666667}
 , energy = -0.4403719609, probable error = 0.001708102234

Parameters = {0.3, 2.1, 0.6}, energy = -0.4727498678, probable error = 0.00135896728

Parameters = {0.3666666667, 2.133333333, 0.4666666667}
 , energy = -0.4487178323, probable error = 0.001410254622

Parameters = {0.2777777778, 2.222222222, 0.5444444444}
 , energy = -0.47425355, probable error = 0.001524287003

Parameters = {0.1666666667, 2.333333333, 0.5666666667}
 , energy = -0.4448478831, probable error = 0.001100719089

Parameters = {0.1296296296, 2.203703704, 0.5740740741}
 , energy = -0.4116060676, probable error = 0.001329473338

Parameters = {0.2222222222, 2.177777778, 0.5555555556}
 , energy = -0.4680257743, probable error = 0.001293225053

Parameters = {0.1666666667, 2.2, 0.6666666667}
 , energy = -0.4475693009, probable error = 0.001204498327

Parameters = {0.3166666667, 2.15, 0.5166666667}
 , energy = -0.4689739614, probable error = 0.001516926321

Parameters = {0.3740740741, 2.137037037, 0.5518518519}
 , energy = -0.4568058624, probable error = 0.001780125518

Parameters = {0.2601851852, 2.167592593, 0.5546296296}
 , energy = -0.4678675902, probable error = 0.001186801827

Parameters = {0.2888888889, 2.161111111, 0.5722222222}
 , energy = -0.4734061713, probable error = 0.001451330678

Parameters = {0.25, 2.2, 0.55}, energy = -0.4745611954, probable error = 0.001883421025

Parameters = {0.2972222222, 2.186111111, 0.5305555556}
 , energy = -0.4629388471, probable error = 0.001289137033

Parameters = {0.2472222222, 2.202777778, 0.5805555556}
 , energy = -0.4739685946, probable error = 0.001561185199
```

Out[111]= {{0.25, 2.2, 0.55}, -0.4745611954, 15}

In[112]:=

```
f1[r_] := Exp[-25/100 r^2] + Exp[-150/100 r^2];
f2[r_] := Exp[-0.25 r^2] + 0.55 Exp[-2.2 r^2];
norm = 1/Sqrt[Integrate[Exp[-2 r], {r, 0, Infinity}]];
norm1 = 1/Sqrt[Integrate[f1[r]^2, {r, 0, Infinity}]];
norm2 = 1/Sqrt[Integrate[f2[r]^2, {r, 0, Infinity}]];
Plot[{Exp[-r] * norm, f1[r] * norm1, f2[r] * norm2}, {r, 0, 4}, PlotRange → All]
```

Out[116]=