

# Diffusion quantum Monte Carlo method

Clear all symbols from previous evaluations to avoid problems

```
In[265]:= Clear["Global`*"]
```

---

## 1D harmonic oscillator

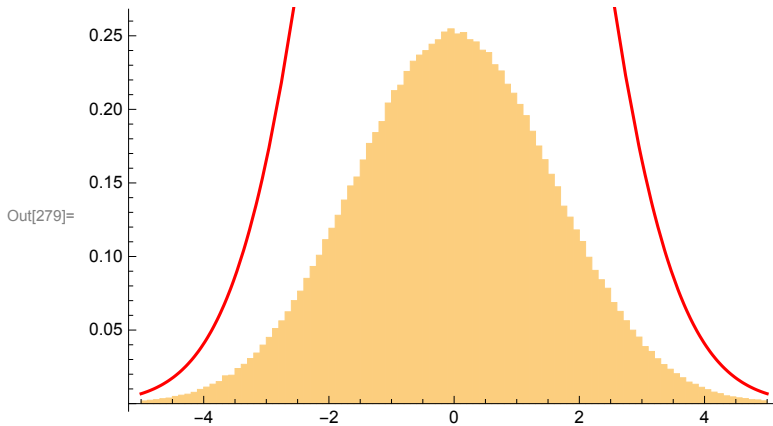
Starting wave function, local energy, diffusion coefficient

```
In[266]:=  $\phi_0[x_, \alpha_] := \text{Exp}[-\alpha * x * x];$   
exactEnergy[ $\alpha_$ ] =  
  Assuming[ $\alpha > 0$ , Integrate[ $\phi_0[x, \alpha] * (-D[\phi_0[x, \alpha], x, x] + x^2 \phi_0[x, \alpha]) / 2$ ,  
    {x, -Infinity, Infinity}] / Integrate[ $\phi_0[x, \alpha]^2$ , {x, -Infinity, Infinity}]]];  
locEn[x_,  $\alpha_$ ] = Simplify[ $1/2 (-D[\phi_0[x, \alpha], x, x] + x^2 \phi_0[x, \alpha]) / \phi_0[x, \alpha]$ ];  
Dif[x_,  $\alpha_$ ] = D[ $\phi_0[x, \alpha]$ , x] /  $\phi_0[x, \alpha]$ ;  
Print[" $\phi_0(x, \alpha) =$ ",  $\phi_0[x, \alpha]$ , " $, \epsilon(x) =$ ", locEn[x,  $\alpha$ ], " $, D(x) =$ ", Dif[x,  $\alpha$ ]]  
  
 $\phi_0(x, \alpha) = e^{-x^2 \alpha}, \epsilon(x) = \alpha + x^2 \left( \frac{1}{2} - 2\alpha^2 \right), D(x) = -2x\alpha$ 
```

Get initial n points with distribution given by  $|\phi_0(x)|^2$

```
In[271]:=  $\alpha_{ini} = 0.1; (* \text{exact for } \alpha = 1/2 *)$   
nini = 1000000;  
a = -5.0; b = 5.0;  
 $\rho[x_] := \phi_0[x, \alpha_{ini}]^2;$   
x0 = 0.0;  $\delta = 4.0;$   
(* use in Metropolis each 10th point of random walk to avoid correlation*)  
points = MyRandomMetropolis1D[nini,  $\rho$ , a, b, x0,  $\delta$ , 10];  
 $\epsilon[x_] := \text{locEn}[x, \alpha_{ini}];$   
energy = MyMCIntegration[points,  $\epsilon$ ];  
Print["Alpha = ",  $\alpha_{ini}$ , " $, \text{mean energy} =$ ",  
  energy[[1]], " $, \text{exact energy} =$ ", exactEnergy[ $\alpha_{ini}$ ]]  
Alpha = 0.1, mean energy = 1.282350347, exact energy = 1.3
```

```
In[279]:= Histogram[points[[1 ;; nini]], {a, b, (b - a) / 100}, "PDF",
  Epilog -> First@Plot[ $\rho[x]$ , {x, a, b}, PlotRange -> All, PlotStyle -> Red]
```



Time evolution of random points:

```
In[280]:= nt = 200; (* number of iterations *)
dt = 0.01; (* time step *)
En = energy[[1]];

xx = ConstantArray[0.0, 2 * nini];
xx[[1 ;; nini]] = points;
nn = nini;
meanEnergy = ConstantArray[0.0, nt];
Do[
  j = 1;
  While[j ≤ nn,
    xx[[j]] = xx[[j]] + Dif[xx[[j]],  $\alpha$ ini] * dt + MyRandomNormal[{0.0, Sqrt[dt]}];
    expEn = Exp[-(locEn[xx[[j]],  $\alpha$ ini] - En) * dt];
    If[
      expEn < 1.0,
      (* then *)
      If[expEn < RandomReal[], (* delete the point (replace with the last one) *)
        xx[[j]] = xx[[nn]];
        nn--;
        j--;
      ],
      (* else *)
      If[expEn - 1.0 > RandomReal[], (* add another point *)
        xx[[nn + 1]] = xx[[j]];
        nn++;
      ]
    ];
    j++;
  ];
  {En, error} = MyMCIntegration[xx[[1 ;; nn]],  $\epsilon$ ];
  meanEnergy[[i]] = En;
  If[Mod[i, 10] == 0,
    Print["Iteration ", i, ", En = ", En, ", number of points: ", nn]],
  {i, 1, nt}
];
```

```

Iteration 10, En = 1.067563824, number of points: 1001209
Iteration 20, En = 0.9230953388, number of points: 1001752
Iteration 30, En = 0.8219621792, number of points: 1002017
Iteration 40, En = 0.7487338195, number of points: 1002758
Iteration 50, En = 0.6952685078, number of points: 1002995
Iteration 60, En = 0.6550190245, number of points: 1003552
Iteration 70, En = 0.6233170062, number of points: 1003472
Iteration 80, En = 0.5986013788, number of points: 1003635
Iteration 90, En = 0.5796163639, number of points: 1003672
Iteration 100, En = 0.5644419674, number of points: 1003587
Iteration 110, En = 0.552181779, number of points: 1003731
Iteration 120, En = 0.5418245234, number of points: 1003904
Iteration 130, En = 0.5335669079, number of points: 1003839
Iteration 140, En = 0.5274715379, number of points: 1003791
Iteration 150, En = 0.5224408851, number of points: 1003741
Iteration 160, En = 0.5185056383, number of points: 1003759
Iteration 170, En = 0.5145690177, number of points: 1003456
Iteration 180, En = 0.5115484951, number of points: 1003399
Iteration 190, En = 0.508756775, number of points: 1003452
Iteration 200, En = 0.5067439619, number of points: 1003302

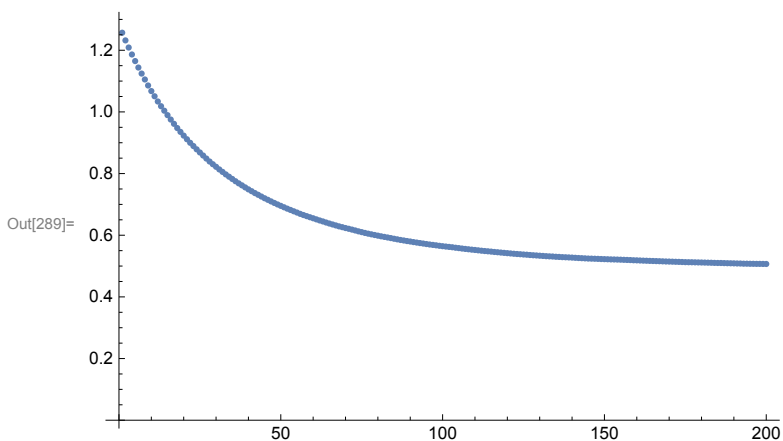
```

```

In[288]:= Print["Average energy ", Mean[meanEnergy[[nt/2 ;; nt]]]];
ListPlot[meanEnergy]

```

Average energy 0.5266031907



```

In[290]:= Histogram[{xx[[1 ;; nn]], points[[1 ;; nini]]}, {a, b, (b - a) / 100}, "PDF",
  Epilog -> First@Plot[ $\rho[x]$ , {x, a, b}, PlotRange -> All, PlotStyle -> Red]

```

