Mathematica solution to assignment

Problem 1

This function takes a Monte Carlo step in x according to the Metropolis algorithm with weight x. Current value of x is given as an argument, and the new value of x (which may equal the old one) is returned.

In[1]:=
```
MCstep[w_, x_] := Block[{xnew = x + RandomReal[{-stepsize, +stepsize}]},
  If[RandomReal[] ≤ w[xnew] / w[x], Sow[1];
   xnew, x] (*Sow is used to track acceptance rates*)
 ]
```

This function computes a single block average for the integral. It takes n MC weighted steps and averages f/w, which should give the integral of f.

In[2]:=
```
MCBlockAverage[w_, f_, n_, xInit_] :=
 Module[{vals = {}, x = xInit},
  Do[
   x = MCstep[w, x];
   AppendTo[vals, f[x] / w[x]],
   {i, n}];
  Mean[vals](*gives the average of the values in vals*)
 ]
```

This computes the integral. It calls the block average routine nB times, and uses the variance of the nB averages to estimate the uncertainty.

In[3]:=
```
MCIntegrate[w_, f_, n_, nB_, xInit_] :=
 Module[{vals = {}, avg, err},
  Do[AppendTo[vals, MCBlockAverage[w, f, n, xInit]], {i, nB}];
  avg = Mean[vals];
  err = Sqrt[Variance[vals] / nB];
  {avg, err}
 ]
```

Here's the integrand.

In[4]:=
```
targetFn[x_] := 3 x^2
```

Evaluate with uniform weight. Output is average and uncertainty. We use 100 blocks of 10,000 steps each.

In[15]:=
```
w1[x_] := If[0 ≤ x ≤ 1, 1, 0]
nSteps = 10 000; nBlocks = 100; stepsize = 1;
result = Reap[MCIntegrate[w1, targetFn, nSteps, nBlocks, 0.5]];
Print["Average, error: ", result[[1]]]
Print["Accept fraction: ", N[Length[result[[2, 1]]] / nSteps / nBlocks]]
```

```
Average, error: {1.00078, 0.0016644}
```

```
Accept fraction: 0.499784
```

Evaluate with linear weight.

In[10]:=
```
w1[x_] := If[0 ≤ x ≤ 1, 2 x, 0]
nSteps = 10 000; nBlocks = 100; stepsize = .6;
result = Reap[MCIntegrate[w1, targetFn, nSteps, nBlocks, 0.5]];
Print["Average, error: ", result[[1]]]
Print["Accept fraction: ", N[Length[result[[2, 1]]] / nSteps / nBlocks]]
```

```
Average, error: {1.00156, 0.000829958}
```

```
Accept fraction: 0.519369
```

Evaluate with cubic weight.

In[25]:=
```
w1[x_] := If[0 ≤ x ≤ 1, 4 x^3, 0]
nSteps = 10 000; nBlocks = 100; stepsize = 0.3;
result = Reap[MCIntegrate[w1, targetFn, nSteps, nBlocks, 0.5]];
Print["Average, error: ", result[[1]]]
Print["Accept fraction: ", N[Length[result[[2, 1]]] / nSteps / nBlocks]]
```

```
Average, error: {0.999864, 0.000935474}
```

```
Accept fraction: 0.554637
```

Problem 2

We do this for the NPT case

We'll use a fixed value of temperature (300 K) and vary pressure

In[30]:=
```
kT = 82.06 (*cm^3-atm/mol-K) *) 300 (*K*)
  10^24 (*A^3/cm^3*) / 6.022*^23 (*mol/molecules*)
```

Out[30]=
```
40 880.1
```

Weight for the isothermal-isobaric simulation

```
nSteps = 1000; nBlocks = 100; Nmol = 100 000;
wNPT[V_] := If[V > 0, V^Nmol Exp[-P V / kT], 0]
targetNPT[V_] := V wNPT[V]
vIG[P_] := 82.06 × 300 / 1000 / P (* ideal-gas volume, in liters *)

(* function to compute the NPT Monte Carlo average volume *)
vNPT[P_] := Block[{vInit = Nmol kT / P},
  Print[P];
  stepsize = vInit / 50;
  result = Reap[MCIntegrate[wNPT, targetNPT, nSteps, nBlocks, vInit]];
  vmolar[P] =
   result[[1]] / Nmol (*A^3/molecules*) 6.022*^23 (*molecules/mol*) / 10^24
     (*cm^3/A^3*) / 1000 (*l/cm^3*); (*l/mol*)
  Print["Average, error: ", vmolar[P], " ", vIG[P]];
  Print["Accept fraction: ", N[Length[result[[2, 1]]] / nSteps / nBlocks]];
  vmolar[P]
 ]
```

In[240]:=
```
nptResults = Table[{P, vNPT[P]}, {P, {1., 5., 10., 50., 100.}}]
```

```
1.
Average, error: {24.6187, 0.000604185} 24.618
Accept fraction: 0.25105
5.
Average, error: {4.92376, 0.000117776} 4.9236
Accept fraction: 0.25173
10.
Average, error: {2.46173, 0.0000545691} 2.4618
Accept fraction: 0.25161
50.
Average, error: {0.492372, 0.0000116027} 0.49236
Accept fraction: 0.25116
100.
Average, error: {0.246178, 5.00669 × 10^-6} 0.24618
Accept fraction: 0.2526
```

Out[240]=
$$\{\{1., \{24.6187, 0.000604185\}\}, \{5., \{4.92376, 0.000117776\}\}, \{10., \{2.46173, 0.0000545691\}\}, \{50., \{0.492372, 0.0000116027\}\}, \{100., \{0.246178, 5.00669 \times 10^{-6}\}\}\}$$

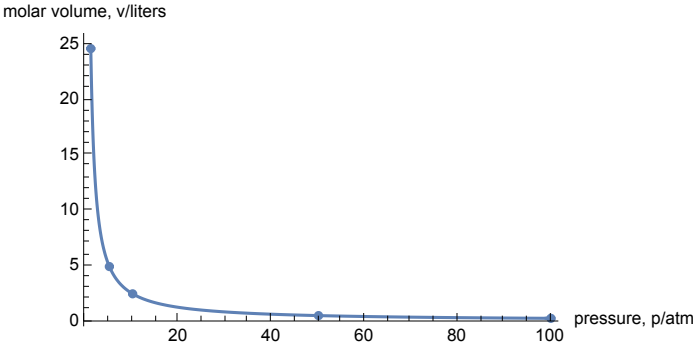
Plot molar-volume averages vs. P. Error bars are too small for the scale, so also plot difference with respect to ideal-gas value (red points)

In[283]:=
```
(*set up data arrays*)
nptResultsErr = {{#[[1]], #[[2, 1]]}, ErrorBar[#[[2, 2]]]} & /@ nptResults
(* this is needed for the error-bar plot *)
nptResultsDiff =
 {{Log[10, #[[1]]], #[[2, 1]] - 82.06 × 300 / 1000 / #[[1]]}, ErrorBar[#[[2, 2]]]} & /@
  nptResults

(* make absolute plot*)
dataPlot = ErrorListPlot[nptResultsErr, PlotRange → {{0, 102}, All}];
igPlot = Plot[vIG[p], {p, 1, 100}, PlotRange → All];
Show[dataPlot, igPlot, AxesLabel → {"pressure, p/atm", "molar volume, v/liters"}]
(* data compare to IG EOS *)

(*make difference plot*)
ErrorListPlot[nptResultsDiff, PlotRange → Automatic,
 PlotStyle → Red, AxesLabel → {"p/atm", "(v-vIG)/liters"}]
```

Out[283]=
```
{{{1., 24.6187}, ErrorBar[0.000604185]},
 {{5., 4.92376}, ErrorBar[0.000117776]}, {{10., 2.46173}, ErrorBar[0.0000545691]},
 {{50., 0.492372}, ErrorBar[0.0000116027]},
 {{100., 0.246178}, ErrorBar[5.00669 × 10^-6]}}
```

Out[284]=
```
{{{0., 0.000673688}, ErrorBar[0.000604185]},
 {{0.69897, 0.00016026}, ErrorBar[0.000117776]},
 {{1., -0.0000681255}, ErrorBar[0.0000545691]},
 {{1.69897, 0.0000121482}, ErrorBar[0.0000116027]},
 {{2., -2.49557 × 10^-6}, ErrorBar[5.00669 × 10^-6]}}
```

Out[287]=



Out[288]=