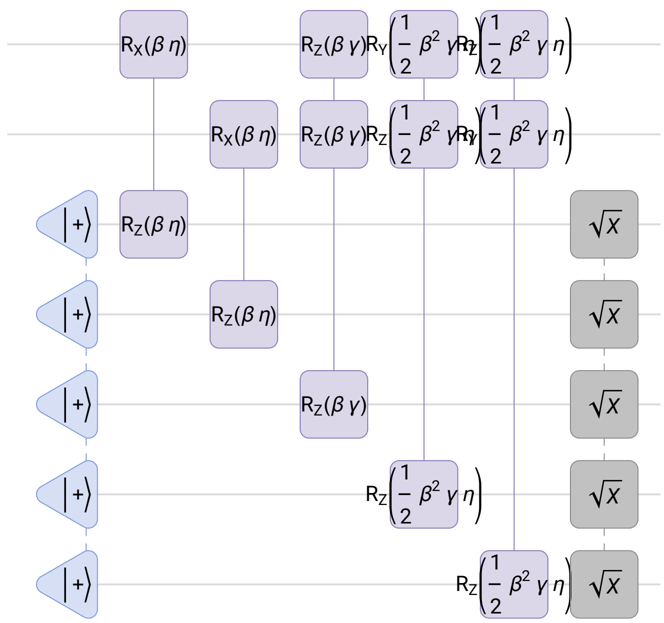
```
In[∘]:= Needs["Wolfram`QuantumFramework`"]

In[∘]:= sx := QuantumOperator["SX", "Label" → "√X"]
        inp := QuantumState["UniformMixture"[2]]

In[∘]:= example := QuantumCircuitOperator[
          {"+" → {3, 4, 5, 6, 7}, QuantumOperator[{"R", β*η, "XZ"}, {1, 3}],
           QuantumOperator[{"R", β*η, "XZ"}, {2, 4}],
           QuantumOperator[{"R", β*ϒ, "ZZZ"}, {1, 2, 5}],
           QuantumOperator[{"R", β^2 * (η*ϒ)/2, "YZZ"}, {1, 2, 6}],
           QuantumOperator[{"R", β^2 * (η*ϒ)/2, "ZYZ"}, {1, 2, 7}], sx → {3, 4, 5, 6, 7}},
          "WireLabels" → {""}, "MeasurementWireLabel" → "", "Parameters" → {η, ϒ, β}]
        example["Diagram"]
Out[∘]=
```




```
ops = {QuantumOperator[{"R", β*η, "XZ"}, {1, 3}], QuantumOperator[
    {"R", β*η, "XZ"}, {2, 4}], QuantumOperator[{"R", β*ϒ, "ZZZ"}, {1, 2, 5}],
   QuantumOperator[{"R", β^2 * (η*ϒ)/2, "YZZ"}, {1, 2, 6}],
   QuantumOperator[{"R", β^2 * (η*ϒ)/2, "ZYZ"}, {1, 2, 7}]};

In[∘]:= cz := QuantumCircuitOperator[{"+" → {3, 4, 5, 6, 7}, ops,
            ops, ops, ops, ops, sx → {3, 4, 5, 6, 7}}, "Parameters" → {η, ϒ, β}]
        cx := QuantumCircuitOperator[{"+" → {3, 4, 5, 6, 7}, ops, ops, ops,
            ops, ops, "H" → {1, 2}, sx → {3, 4, 5, 6, 7}}, "Parameters" → {η, ϒ, β}]

In[∘]:= Γ = 0.785398163;
        Λ = 0.785398163;
```

```
pos = Range[1, 128];
spins = 1 - 2 * IntegerDigits[pos - 1, 2, 7];
Fz[res_] := Sum[
  Γ * res[[pos]] * spins[[pos]][[1]] * spins[[pos]][[2]] * spins[[pos]][[5]] / shots, {pos, 1, 128}]
Fx[res_] :=
 Sum[Λ * res[[pos]] * (spins[[pos]][[3]] * spins[[pos]][[1]] + spins[[pos]][[4]] * spins[[pos]][[2]]) /
   shots, {pos, 1, 128}]

Hz = Table[cz[<|η → 2 Λ, γ → 2 Γ, β → -i|>][inp], {i, 0.01, 0.15, 0.01}];

Hx = Table[cx[<|η → 2 Λ, γ → 2 Γ, β → -i|>][inp], {i, 0.01, 0.15, 0.01}];
```

In[∘]:=
```
shots := 10 000
resz = Values[QuantumMeasurementSimulation[#,
      QuantumMeasurementOperator /@ {"ZZZZZZZ"}, shots]][[1]] & /@ Hz;
resx = Values[QuantumMeasurementSimulation[#,
      QuantumMeasurementOperator /@ {"ZZZZZZZ"}, shots]][[1]] & /@ Hx;
```

In[∘]:=
```
approx = Prepend[Most[Map[Fz, resz] + Map[Fx, resx]], 0.]
```

Out[∘]=
```
{0., -0.175615, -0.369294, -0.546794, -0.702303, -0.861111, -0.999498, -1.14621,
 -1.25884, -1.37209, -1.44388, -1.51362, -1.5298, -1.55996, -1.55226}
```

In[∘]:=
```
X = {{0, 1}, {1, 0}};
Z = {{1, 0}, {0, -1}};
I2 = {{1, 0}, {0, 1}};

H = Γ KroneckerProduct[Z, Z] +
   Λ (KroneckerProduct[X, I2] + KroneckerProduct[I2, X]);

exact = Most[Table[Z = Tr[MatrixExp[-β H]];
     energy = Tr[H.MatrixExp[-β H]] / Z;
     {β, N[energy]}, {β, Range[0, 0.15, 0.01] * 2 * 5}][[All, 2]]];
```

In[∘]:=
```
gs = Min[Eigenvalues[H]]
```

Out[∘]=
```
-1.7562
```

```
In[•]:= p1 = Table[{(i - 1) * 10^-1, approx[[i]]}, {i, 2, Length[approx]}];
p2 = Table[{(i - 1) * 10^-1, exact[[i]]}, {i, 2, Length[exact]}];
fit =
  FindFit[p1, yInf + (y0 - yInf) Exp[-γ x], {{yInf, -1.6}, {y0, 0}, {γ, 0.5}}, x];

f1[x_] = (yInf + (y0 - yInf) Exp[-γ x]) /. fit;

f2 = Interpolation[p2, InterpolationOrder → 2];

xmin = Min[Min[p1[[All, 1]]], Min[p2[[All, 1]]]];
xmax = Max[Max[p1[[All, 1]]], Max[p2[[All, 1]]]];

ymin = Min[Min[p1[[All, 2]]], Min[p2[[All, 2]]]];
ymax = Max[Max[p1[[All, 2]]], Max[p2[[All, 2]]]];

marginFactor = 0.15;
margin = marginFactor * Abs[gs] + 10 ^ -1;

ylo = Min[ymin, gs - margin];
yhi = ymax;

Plot[{f1[x], f2[x], gs}, {x, xmin, xmax},
 PlotRange → {{xmin, xmax}, {ylo, yhi}}, PlotRangePadding → Scaled[.01],
 PlotStyle → {{Directive[Blue, Dashed, Opacity[0.5]]},
   {Directive[Red, Dashed, Opacity[0.5]]},
   {Directive[GrayLevel[0.25], Dashed, Opacity[0.4], AbsoluteThickness[2]]} },
 PlotLegends → Placed[{"approx", "exact", "ground"}, {0.8, 0.65}],
 AxesLabel → {"β", "\!\(\*SubscriptBox[\(⟨H⟩\), \(β\)]\)"},
 GridLines → {Automatic, None}, LabelStyle → Directive[FontSize → 15],
 Epilog → {{Blue, Opacity[0.5], PointSize[0.015], Point[p1]},
   {Red, Opacity[0.5], PointSize[0.015], Point[p2]}}]
```

Out[ ]=