## Process simulated data for all Q values simulataneously

Clear all previously calculated variable values

```
ClearAll["Global`*"]
```

Set the grid size for the input data and calculate data

```
GridSize = L;
```

```
delta = (GridSize)/(GridSize^2);
```

Now Import the Lattice Data from the spreadsheet

```
DataGrid = ImportString["", "TSV"];
```

### Process the Perioidic Lattice data

Define a function that converts a list {1,2,3,4,5} to access the correct element in the imported data

```
CorrectRegColumn[q_]:=If[q==2, 4, If[q==3, 5, If[q==4, 6, If[q==8, 7, If[q==10, 8, 4]]]]]
```

Define the function that calculates the continuity constants as a function of midpoint number and q.

```
ContinuityConstantsReg[k_, q_]:=(DataGrid[[1]][[CorrectRegColumn[q]]]/2 + Sum[DataGrid[[i]][[CorrectRegColumn[q]]],
{i, 1, k - 1}] + DataGrid[[k]][[CorrectRegColumn[q]]]/2) * delta
```

Define a function of E done as x for simplicity and q using the Piecewise function generator natively embedded in Mathematica

```
S0[x_, q_]:=Piecewise[Table[{ContinuityConstantsReg[i, q] + DataGrid[[i]][[CorrectRegColumn[q]]](x - DataGrid[[i]][[3]]),
DataGrid[[i]][[1]]<=x<=DataGrid[[i]][[2]]}, {i, 1, Length[DataGrid]}]]
```

Plot a graph across the Energy Range {-2,-2/q} for all of the Q values being studied to ensure that continuity occurs as expected.

```
Table[Plot[S0[x, q], {x, -2, -2/q}, ImageSize->Large, AxesLabel->{E, Log[ρ[E]]}], {q, {2, 3, 4, 8, 10}}]
```

Calculate C0 for the Periodic Lattice

```
C0regular[q_]:=(2 * (GridSize^2))/(NIntegrate[S0[x, q], {x, -2, -2/q}])
```

Define the DoS for the Periodic Lattice

```
DoSregular[x_, q_]:=Exp[C0regular[q] + S0[x, q]]
```

Define the Partition Function for the Periodic Lattice

```
Zreg[beta_, q_]:=NIntegrate[Log[DoSregular[x, q]] + beta * x, {x, -2, -2/q}]
```

### Now to process the twisted Data

Define a function that converts a list {1, 2, 3, 4, 5} to access the correct element in the imported data

```
CorrectIntColumn[q_]:=If[q==2, 9, If[q==3, 10, If[q==4, 11, If[q==8, 12, If[q==10, 13, 9]]]]]
```

Define the function that calculates the continuity constants as a function of midpoint number and q.

```
ContinuityConstantsInt[k_, q_]:=(DataGrid[[1]][[CorrectIntColumn[q]]]/2 + Sum[DataGrid[[i]][[CorrectIntColumn[q]]],
{i, 1, k - 1}] + DataGrid[[k]][[CorrectIntColumn[q]]]/2) * delta
```

Define a function of E done as x for simplicity and q using the Piecewise function generator natively embedded in Mathematica

```
S0int[x_, q_]:=Piecewise[Table[{ContinuityConstantsInt[i, q] + DataGrid[[i]][[CorrectIntColumn[q]]](x - DataGrid[[i]][[3]]),
DataGrid[[i]][[1]]<=x<=DataGrid[[i]][[2]]}, {i, 1, Length[DataGrid]}]]
```

Plot a graph across the Energy Range {-2,-2/q} for all of the Q values being studied to ensure that continuity occurs as expected.

```
Table[Plot[S0int[x, q], {x, -2, -2/q}, ImageSize->Large, AxesLabel->{E, Log[ρ[E]]}], {q, {2, 3, 4, 8, 10}}]
```

Calculate C0 for the Twisted Lattice

```
C0interface[q_]:=(2 * (GridSize^2))/(NIntegrate[S0int[x, q], {x, -2, -2/q}])
```

Define the DoS for the Twisted Lattice

```
DoSinterface[x_, q_]:=C0interface[q] + Exp[S0[x, q]]
```

Define the Partition Function for the Periodic Lattice

```
Zint[beta_, q_]:=NIntegrate[Log[DoSinterface[x, q]] + beta * x, {x, -2, -2/q}]
```

**NowtocalculatetheFreeEnergyoftheInterfaceattheCriticalPoint**

atcalculateCriticalEnergyFreeInterfaceNowofPointthe$^3$to

## Calculate the Interface Free Energy from the Twisted and Periodic Partiton Functions

```
IntFreeEnergy[beta_, q_]:= - Log[Zint[beta, q]/Zreg[beta, q]] - Log[GridSize]
```

```
Table[IntFreeEnergy[Log[1 + Sqrt[q]], q], {q, {2, 3, 4, 8, 10}}]
```