1 4-d $\mathbb{Z}_2$ gauge theory on the 16^4 lattice

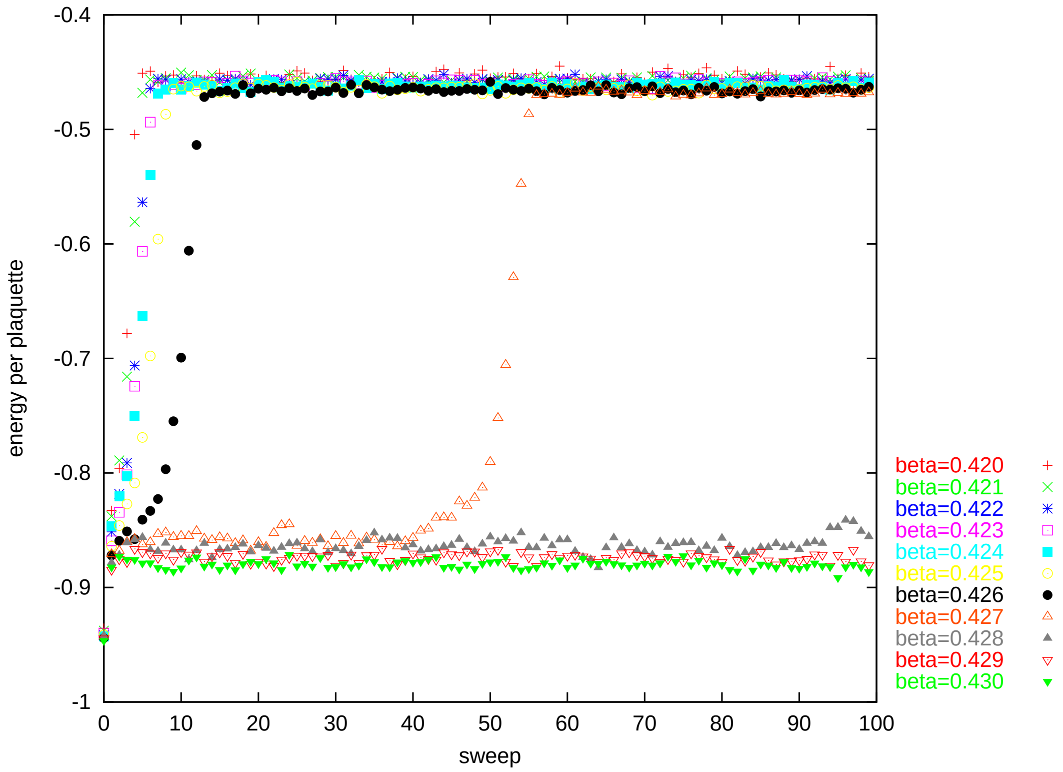
The Lagrangian is

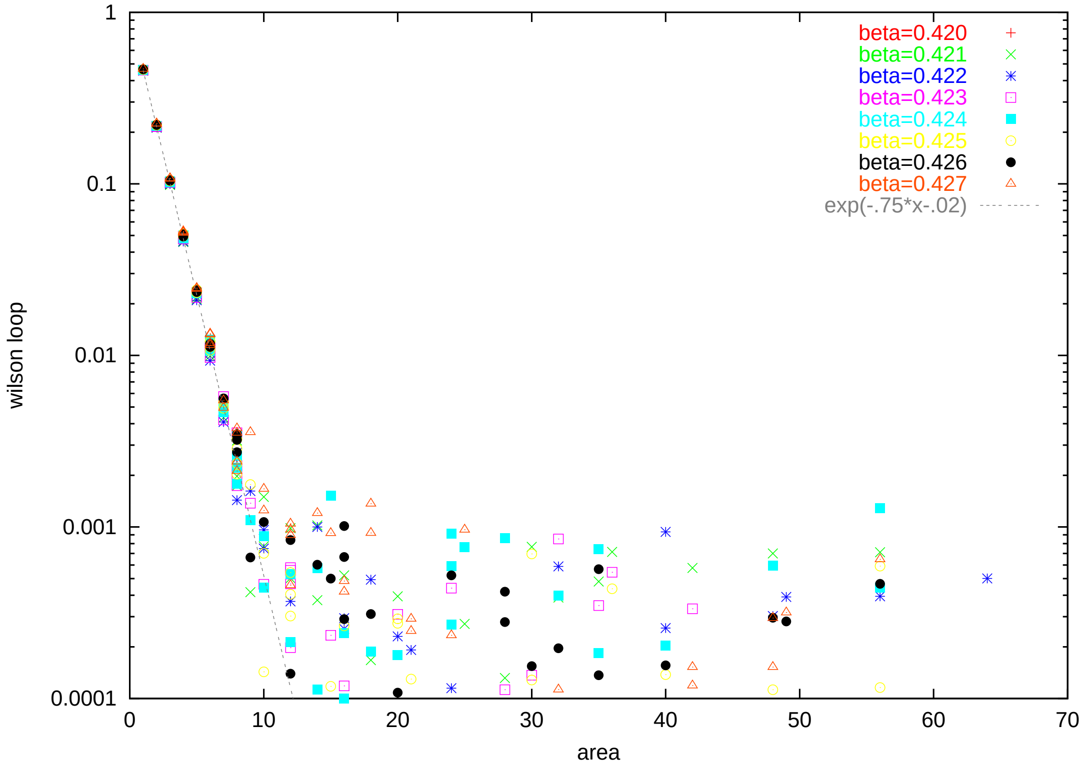
$$\mathcal{L} = -\sum_{\text{plaquettes}} \sigma\sigma\sigma\sigma.$$

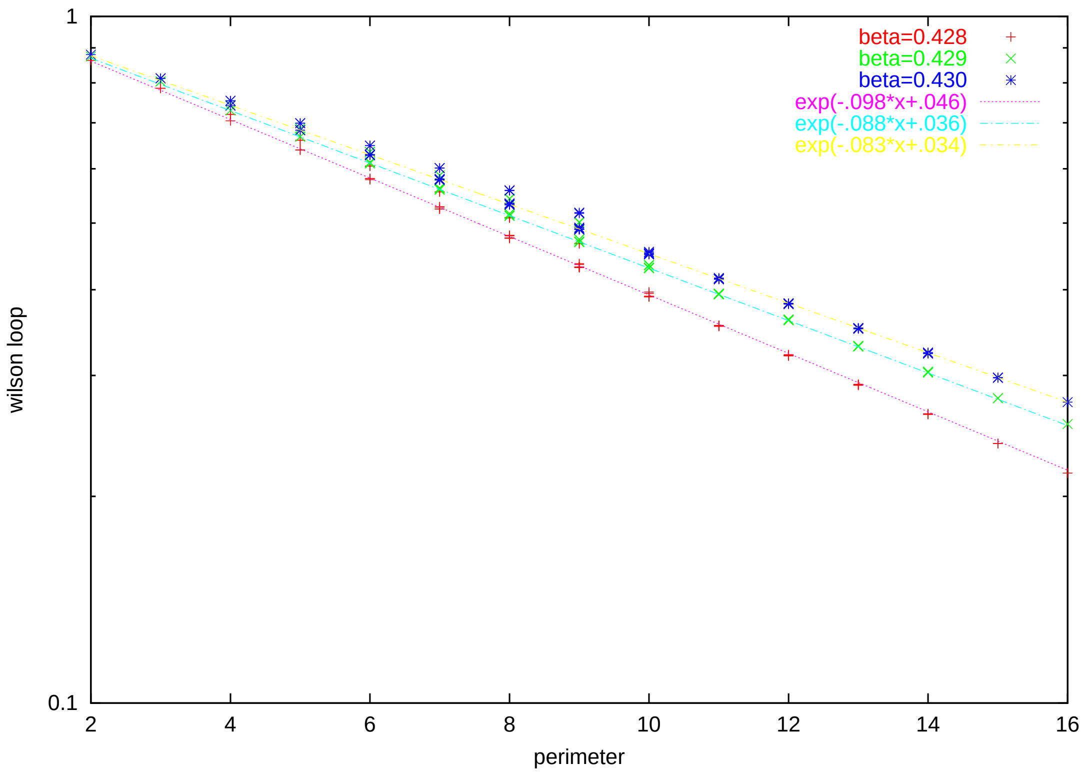
and the partition function is

$$Z = \sum_{\text{config}} e^{-\beta\mathcal{L}}.$$

We calculated by the Metropolis method two observables: the energy per plaquette and the wilson loop.







Loops with either side= 1 display slightly different scalings in the deconfined regime.

2 4-d S_3 gauge theory on the 16^4 lattice

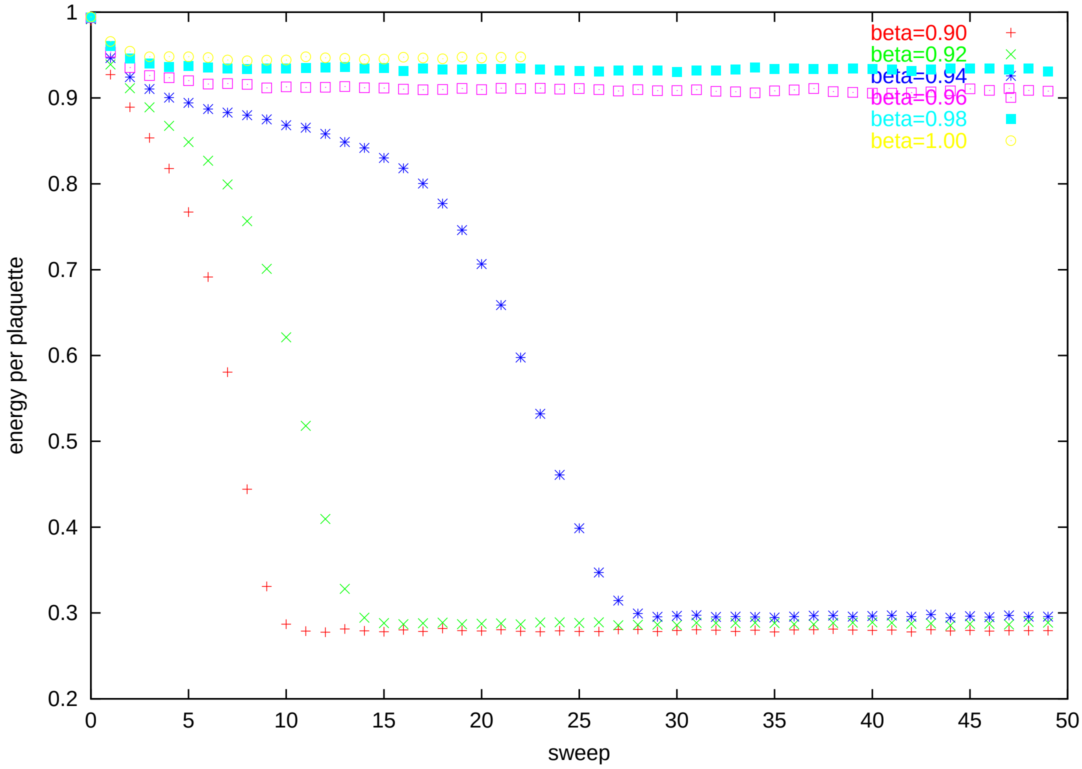
Next, we take the group to be the symmetric group of order 3, with six elements. The Lagrangian is

$$\mathcal{L} = -\sum_{\text{plaquettes}} \text{tr}(UUUU)/2.$$

where U is in the 2-d representation, and the partition function is

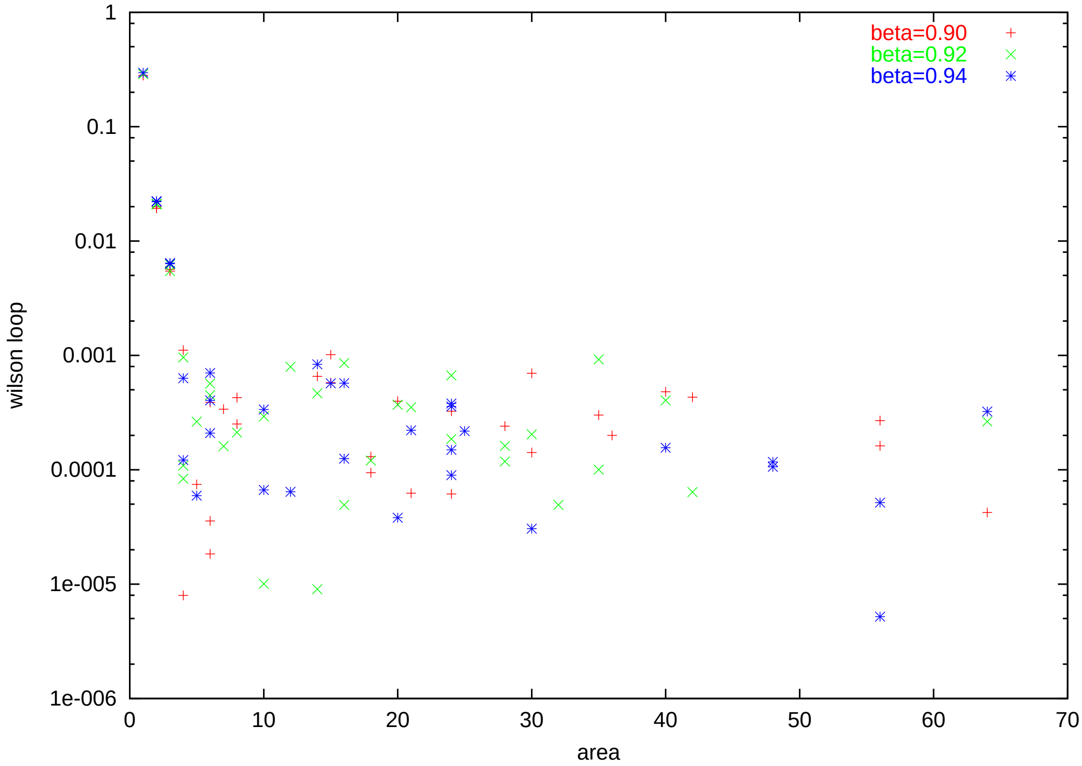
$$Z = \sum_{\text{config}} e^{-\beta\mathcal{L}}.$$

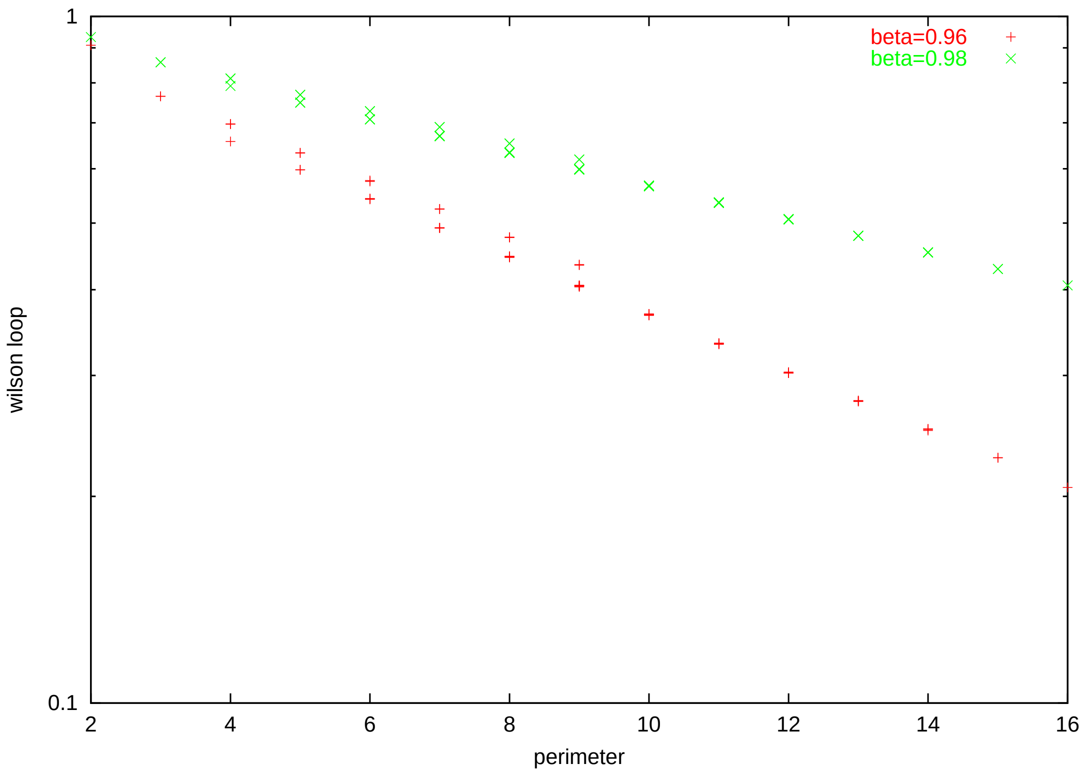
Surprisingly this system, even though non-abelian, shows 1-st order confinement–deconfinement transition.



Please see how tedious it would be to program the plaquette energy... not just tiresome; it may cause trivial bugs, hard to remove. We used a Perl script to generate this part of the program.

```
inline double plaquettes(int i,int j,int k,int l,int s){
switch(s){
case 0: return (
+character[multiply[multiply[multiply[spin[i][j][k][l][0]][spin[f[i]][j][k][l][1]]][invert[spin[i][f[j]][k][l][0]]]][invert[spin[i][j][k][l][1]]]]
+character[multiply[multiply[multiply[spin[i][b[j]][k][l][0]][spin[f[i]][b[j]][k][l][1]]][invert[spin[i][j][k][l][0]]]][invert[spin[i][b[j]][k][l][1]]]]
+character[multiply[multiply[multiply[spin[i][j][k][l][0]][spin[f[i]][j][k][l][2]]][invert[spin[i][j][f[k]][l][0]]]][invert[spin[i][j][k][l][2]]]]
+character[multiply[multiply[multiply[spin[i][j][b[k]][l][0]][spin[f[i]][j][b[k]][l][2]]][invert[spin[i][j][k][l][0]]]][invert[spin[i][j][b[k]][l][2]]]]
+character[multiply[multiply[multiply[spin[i][j][k][l][0]][spin[f[i]][j][k][l][3]]][invert[spin[i][j][k][f[l]][0]]]][invert[spin[i][j][k][l][3]]]]
+character[multiply[multiply[multiply[spin[i][j][k][b[l]][0]][spin[f[i]][j][k][b[l]][3]]][invert[spin[i][j][k][l][0]]]][invert[spin[i][j][k][b[l]][3]]]]);
case 1: return (
+character[multiply[multiply[multiply[spin[i][j][k][l][1]][spin[i][f[j]][k][l][0]]][invert[spin[f[i]][j][k][l][1]]]][invert[spin[i][j][k][l][0]]]]
+character[multiply[multiply[multiply[spin[b[i]][j][k][l][1]][spin[b[i]][f[j]][k][l][0]]][invert[spin[i][j][k][l][1]]]][invert[spin[b[i]][j][k][l][0]]]]
+character[multiply[multiply[multiply[spin[i][j][k][l][1]][spin[i][f[j]][k][l][2]]][invert[spin[i][j][f[k]][l][1]]]][invert[spin[i][j][k][l][2]]]]
+character[multiply[multiply[multiply[spin[i][j][b[k]][l][1]][spin[i][f[j]][b[k]][l][2]]][invert[spin[i][j][k][l][1]]]][invert[spin[i][j][b[k]][l][2]]]]
+character[multiply[multiply[multiply[spin[i][j][k][l][1]][spin[i][f[j]][k][l][3]]][invert[spin[i][j][k][f[l]][1]]]][invert[spin[i][j][k][l][3]]]]
+character[multiply[multiply[multiply[spin[i][j][k][b[l]][1]][spin[i][f[j]][k][b[l]][3]]][invert[spin[i][j][k][l][1]]]][invert[spin[i][j][k][b[l]][3]]]]);
case 2: return (
+character[multiply[multiply[multiply[spin[i][j][k][l][2]][spin[i][j][f[k]][l][0]]][invert[spin[f[i]][j][k][l][2]]]][invert[spin[i][j][k][l][0]]]]
+character[multiply[multiply[multiply[spin[b[i]][j][k][l][2]][spin[b[i]][j][f[k]][l][0]]][invert[spin[i][j][k][l][2]]]][invert[spin[b[i]][j][k][l][0]]]]
+character[multiply[multiply[multiply[spin[i][j][k][l][2]][spin[i][j][f[k]][l][1]]][invert[spin[i][f[j]][k][l][2]]]][invert[spin[i][j][k][l][1]]]]
+character[multiply[multiply[multiply[spin[i][b[j]][k][l][2]][spin[i][b[j]][f[k]][l][1]]][invert[spin[i][j][k][l][2]]]][invert[spin[i][b[j]][k][l][1]]]]
+character[multiply[multiply[multiply[spin[i][j][k][l][2]][spin[i][j][f[k]][l][3]]][invert[spin[i][j][k][f[l]][2]]]][invert[spin[i][j][k][l][3]]]]
+character[multiply[multiply[multiply[spin[i][j][k][b[l]][2]][spin[i][j][f[k]][b[l]][3]]][invert[spin[i][j][k][l][2]]]][invert[spin[i][j][k][b[l]][3]]]]);
case 3: return (
+character[multiply[multiply[multiply[spin[i][j][k][l][3]][spin[i][j][k][f[l]][0]]][invert[spin[f[i]][j][k][l][3]]]][invert[spin[i][j][k][l][0]]]]
+character[multiply[multiply[multiply[spin[b[i]][j][k][l][3]][spin[b[i]][j][k][f[l]][0]]][invert[spin[i][j][k][l][3]]]][invert[spin[b[i]][j][k][l][0]]]]
+character[multiply[multiply[multiply[spin[i][j][k][l][3]][spin[i][j][k][f[l]][1]]][invert[spin[i][f[j]][k][l][3]]]][invert[spin[i][j][k][l][1]]]]
+character[multiply[multiply[multiply[spin[i][b[j]][k][l][3]][spin[i][b[j]][k][f[l]][1]]][invert[spin[i][j][k][l][3]]]][invert[spin[i][b[j]][k][l][1]]]]
+character[multiply[multiply[multiply[spin[i][j][k][l][3]][spin[i][j][k][f[l]][2]]][invert[spin[i][j][f[k]][l][3]]]][invert[spin[i][j][k][l][2]]]]
+character[multiply[multiply[multiply[spin[i][j][b[k]][l][3]][spin[i][j][b[k]][f[l]][2]]][invert[spin[i][j][k][l][3]]]][invert[spin[i][j][b[k]][l][2]]]]);
}
}
```





As before, loops of size $1 \times N$ show somewhat stronger correlation.