Test finite temperature lanczos code.

In [42]:
```
include("package.jl")
include("kagome_setup.jl")
include("KH_hamiltonian.jl")
include("ED_Thermal.jl")
include("Full_Orthogonalization_Lanczos.jl")
include("Kagome-FTLM.jl")
println("Finish loading functions!")
```

```
Finish loading packages!
Finish loading functions!
```

In [71]:
```
include("OFTLM.jl")
```

Out[71]: OFTLM (generic function with 1 method)

In [39]:
```
θ = 0.2
J = cospi(θ) * ones(3)
K = sinpi(θ) * ones(3)

la = Lattice(2,2)
N = sitenum(la)
dim = dimension(la)

Hk = Kitaev(K, la)
Hh = Heisenberg(J, la)
Htot = Hk + Hh;

println("Finish setting Hamiltonian!")
```

```
Finish setting Hamiltonian!
```

In [40]:
```
# Full ED
Hdense  = Array(Htot)
e,x = eigen(Hdense)
println("Finish diagonalizing K-H model!")
```

```
Finish diagonalizing K-H model!
```

In [43]:
```
V = FTLM(Htot, M = 90, R = 50)
println("Finish FTLM!")
```

```
Finish FTLM!
```

In [72]:
```
Vo = OFTLM(Htot, M = 90, R = 50)
println("Finish OFTLM!")
```

```
Finish OFTLM!
```

In [44]:
```
""" ED Thermaldynamic quanties"""
step = 100
temp = [10^x for x in LinRange(-4,2,step)]
Z = zeros(step)
E = zeros(step)
C = zeros(step)
S = zeros(step)

#println(Z)
for i = 1: step
    t = temp[i]
    Z[i] += partitian(t, e)
    E[i] += energy(t,e)
    C[i] += specific_heat(t,e)
    S[i] += E[i]/t +log(Z[i])
end

println("ed: Finish calculate Thermaldynamic quanties!")
```

```
ed: Finish calculate Thermaldynamic quanties!
```

In [45]:
```
""" Thermaldynamic quanties"""
step = 100
temp = [10^x for x in LinRange(-4,2,step)]
Zt = zeros(step)
Et = zeros(step)
Ct = zeros(step)
St = zeros(step)

for i = 1: step
    t = temp[i]
    Zt[i] = FTLM_partition(V,t)
    Et[i], Ct[i] = FTLM_EandC(V,t)
    St[i] = Et[i]/t + log(Zt[i])
end

println("ftlm: Finish calculate Thermaldynamic quanties!")
```

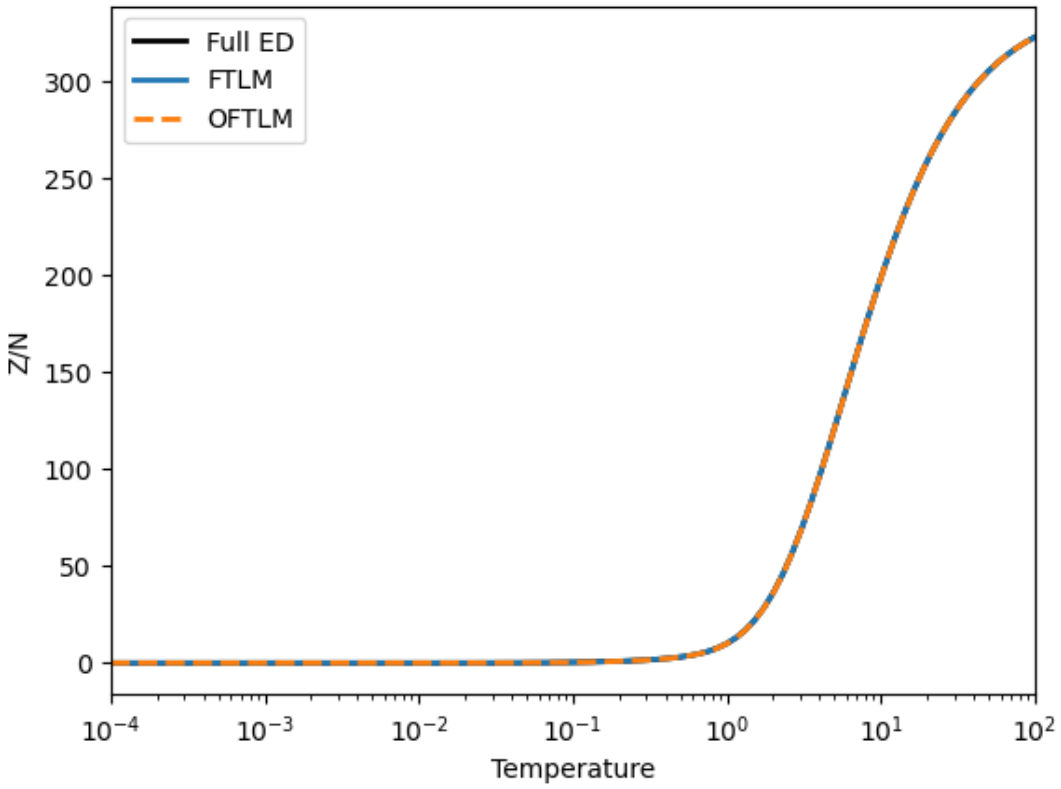
```
ftlm: Finish calculate Thermaldynamic quanties!
```

In [75]:
```
""" OFTLM Thermaldynamic quanties"""
step = 100
temp = [10^x for x in LinRange(-4,2,step)]
Zo = zeros(step)
Eo = zeros(step)
Co = zeros(step)
So = zeros(step)

for i = 1: step
    t = temp[i]
    Zo[i] = FTLM_partition(Vo,t)
    Eo[i], Co[i] = FTLM_EandC(Vo,t)
    So[i] = Eo[i]/t + log(Zo[i])
end

println("oftlm: Finish calculate Thermaldynamic quanties!")
```

```
oftlm: Finish calculate Thermaldynamic quanties!
```

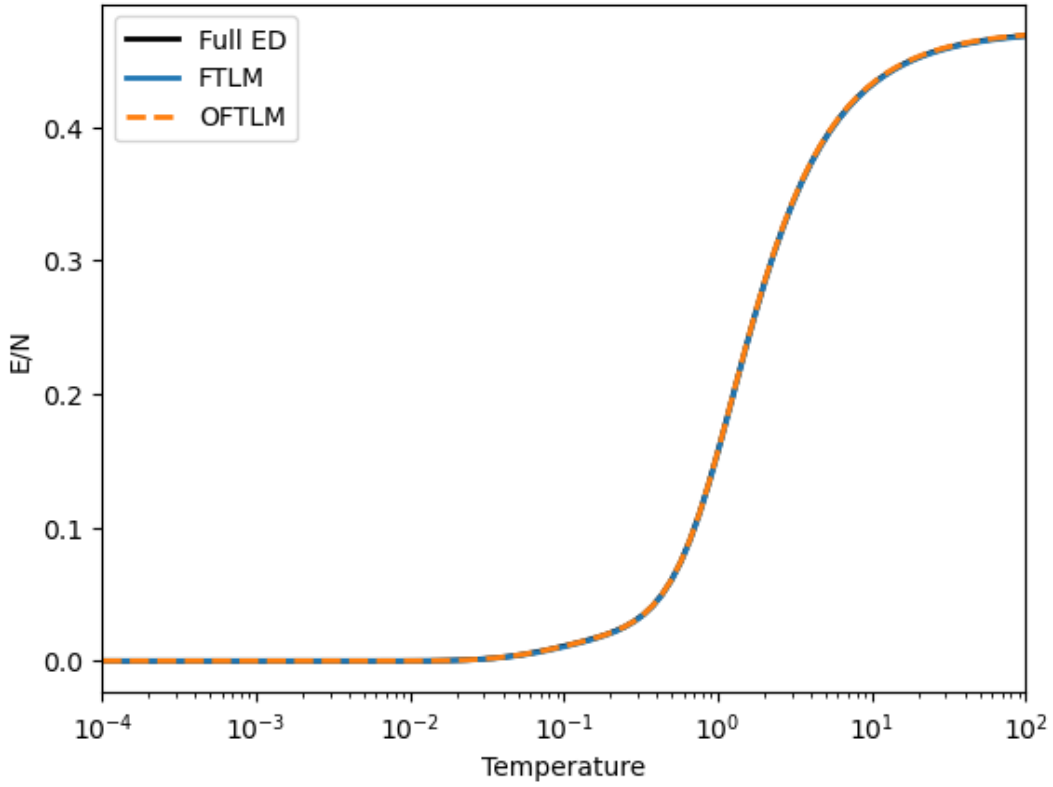
In [120]:

```
semilogx(temp, Z/N, linewidth = 2, color = "k", label = "Full ED")
semilogx(temp, Zt/N, linewidth = 2, label = "FTLM" )
semilogx(temp, Zo/N, linewidth = 2, "--",label = "OFTLM")
legend()
ylabel("Z/N")
xlabel("Temperature")
xlim(10^-4, 10^2);
```



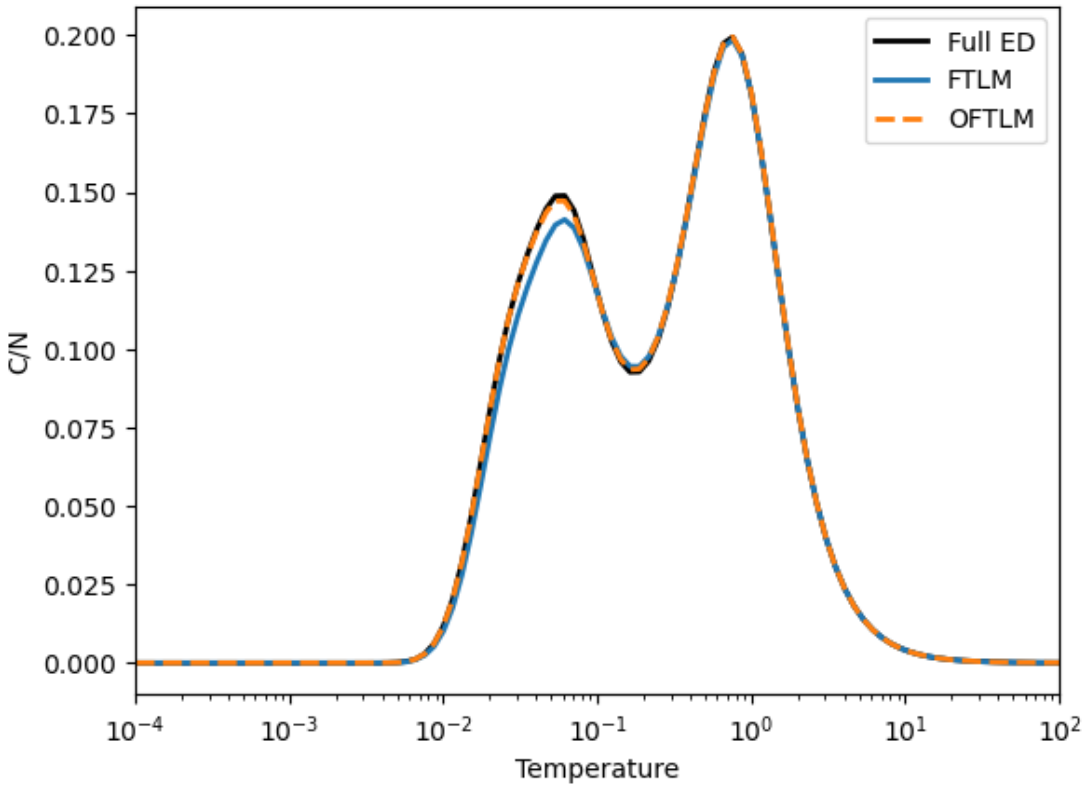
In [121]:

```
semilogx(temp, E/N, linewidth = 2, color = "k", label = "Full ED")
semilogx(temp, Et/N, linewidth = 2, label = "FTLM" )
semilogx(temp, Eo/N, linewidth = 2, "--", label = "OFTLM")
legend()
ylabel("E/N")
xlabel("Temperature")
xlim(10^-4, 10^2);
```

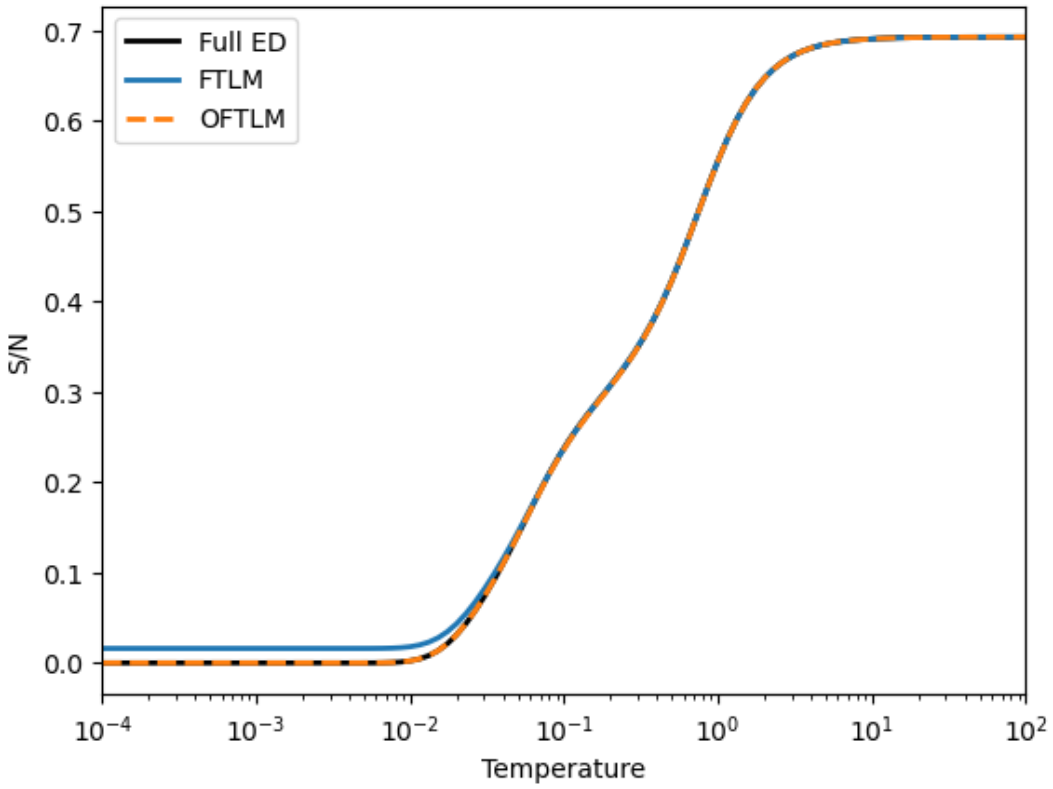


In [122]:

```
semilogx(temp, C/N, linewidth = 2, color = "k", label = "Full ED")
semilogx(temp, Ct/N, linewidth = 2, label = "FTLM" )
semilogx(temp, Co/N, linewidth = 2, "--", label = "OFTLM")
legend()
ylabel("C/N")
xlabel("Temperature")
xlim(10^-4, 10^2);
```



In [123]:
```
semilogx(temp, S/N, linewidth = 2, color = "k", label = "Full ED")
semilogx(temp, St/N, linewidth = 2, label = "FTLM" )
semilogx(temp, So/N, linewidth = 2, "--", label = "OFTLM")
legend()
ylabel("S/N")
xlabel("Temperature")
xlim(10^-4, 10^2);
```



In [ ]:
```
1
```