

Model Isinga 2D

Michał Łukomski

Styczeń 2021

Symulacje zostały przeprowadzone na sieciach o rozmiarze L^2 dla $L \in \{6, 20, 70\}$. Temperaturę wyrażono w jednostkach zredukowanych $T^* = \frac{k_B}{J}T$.

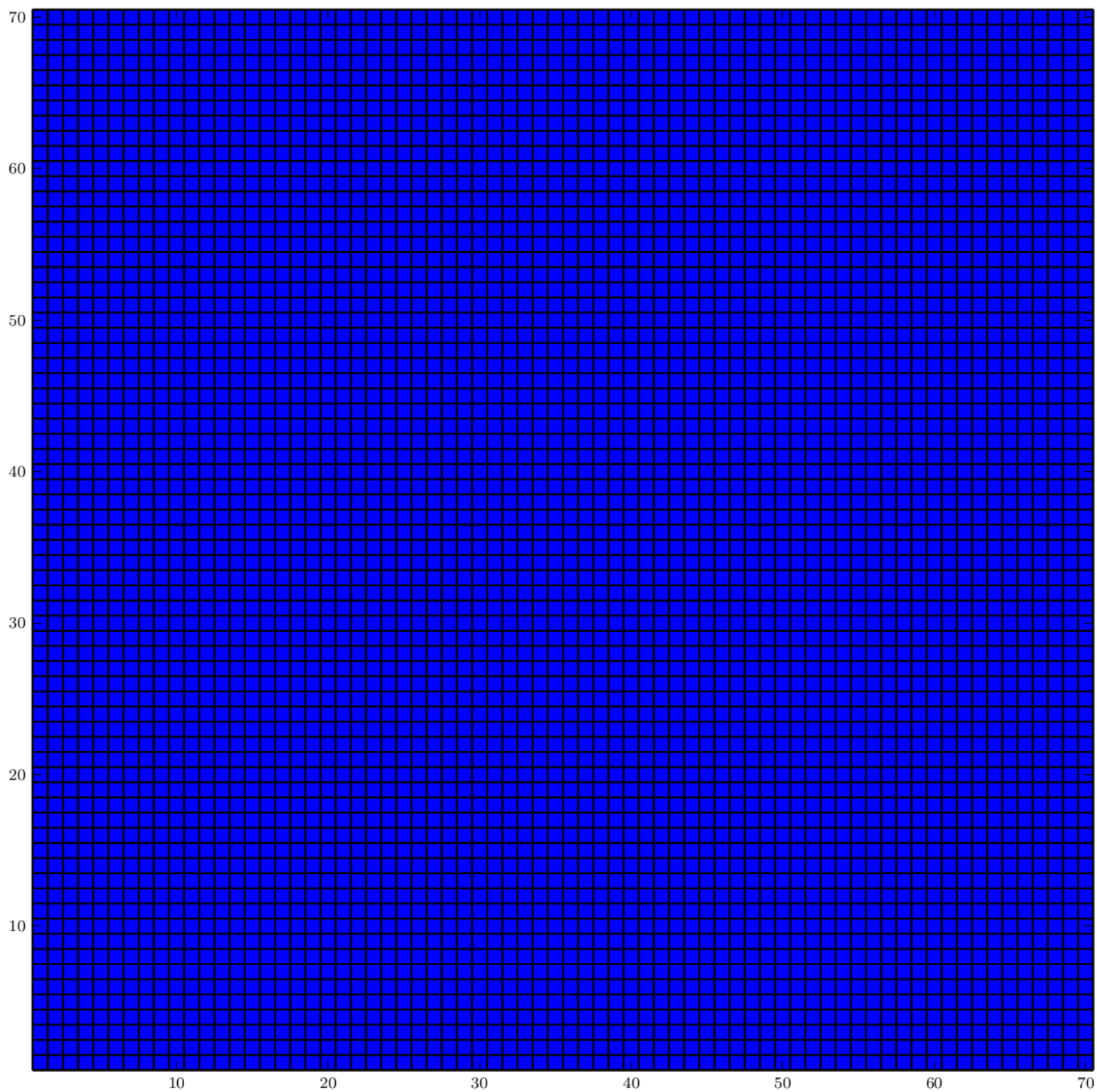
Symulacje startowano z konfiguracjami losowymi - każdemu węzłowi przypisywano stan $+1$ lub -1 z równym prawdopodobieństwem $1/2$. Sieci zostały poddane ewolucji przez $30000MCS$, ($1MCS$ = iteracyjne przejście po wszystkich węzłach układu) a następnie przez kolejne $400000MCS$, podczas których magnetyzacja była próbkowana co $100MCS$. Otrzymano więc próbki o rozmiarach 4000, które wykorzystano do obliczeń średniej magnetyzacji, podatności magnetycznej oraz pojemności cieplnej układu. Wyniki przedstawiono na wykresach.

Zakresy temperatur używane w symulacjach to

- $[0.5; 1.75]$ ze skokiem $\Delta T^* = 0.05$
- $[1.8; 2.7]$ ze skokiem $\Delta T^* = 0.005$
- $[2.75; 3.5]$ ze skokiem $\Delta T^* = 0.05$

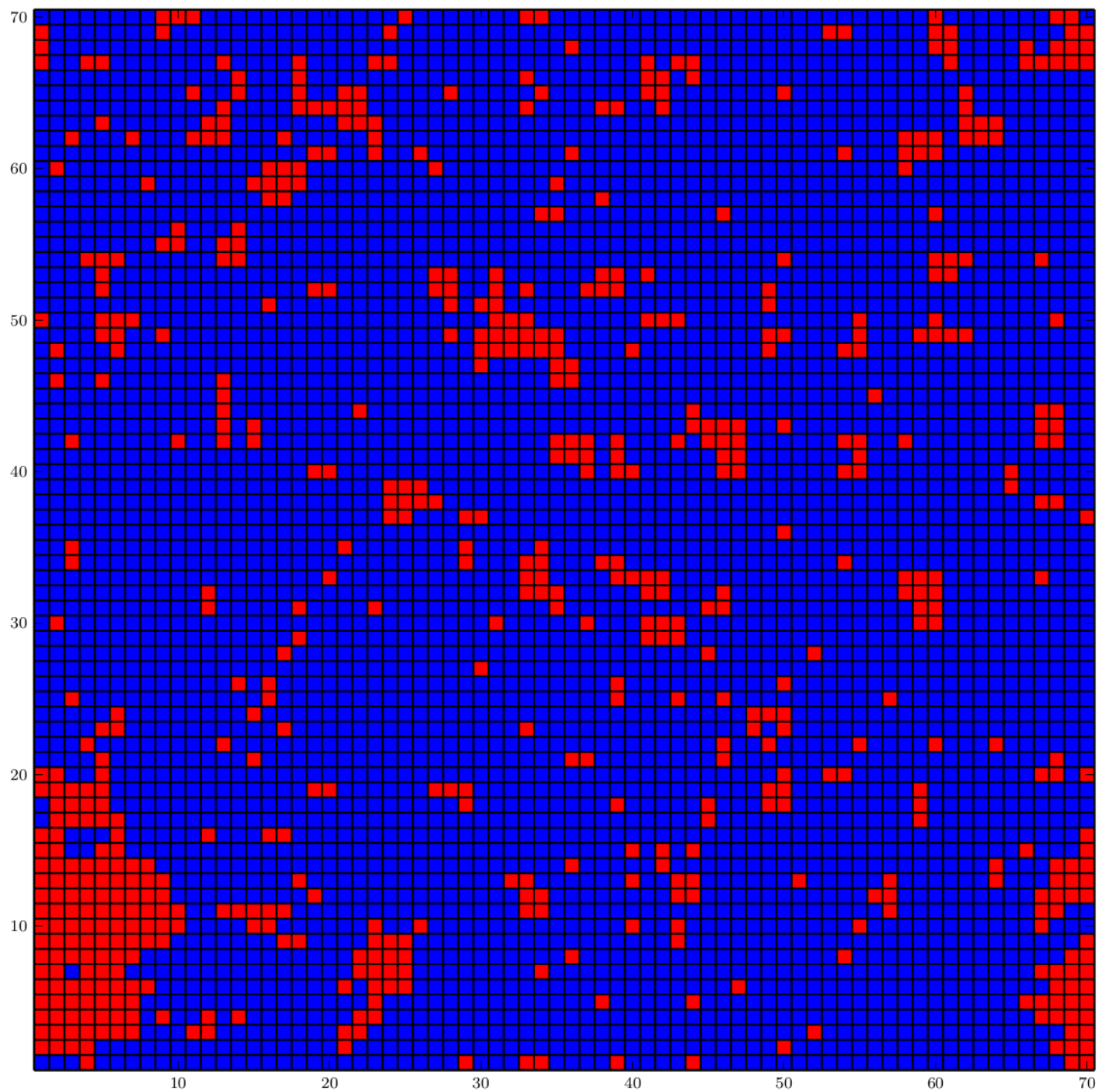
Poniżej przedstawiono zaobserwowane układy dla temperatur $T^* = 0.05, T^* = 2.269 \approx T_C^*, T^* = 8.0$. Kolor niebieski oznacza węzeł w stanie -1 , a czerwony w stanie $+1$.

$$L=70, T^*=0.05$$



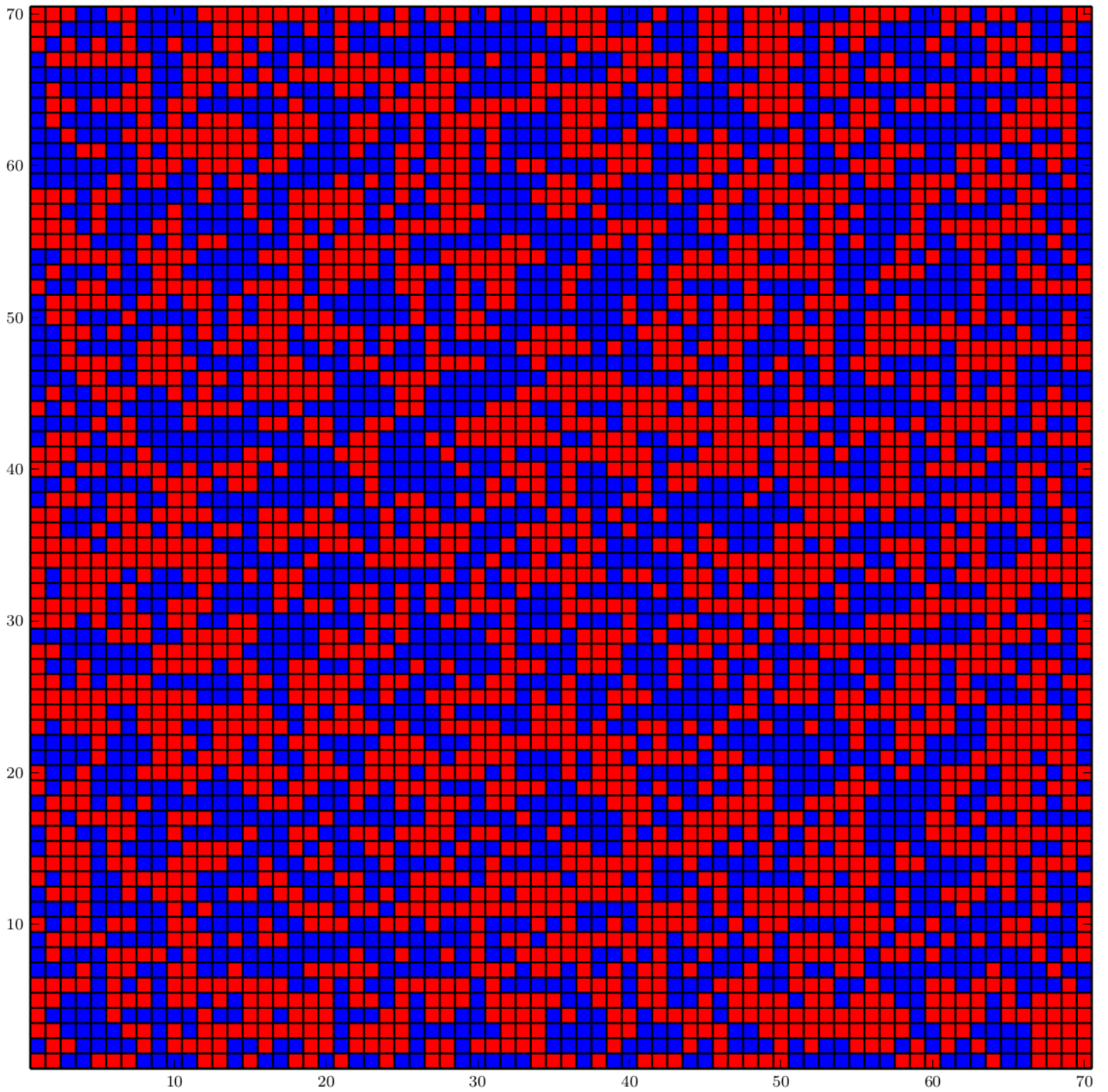
Rysunek 1: Konfiguracja dla $L = 70$ i $T^* = 0.050$ po $430000 MCS$. Kolor niebieski to stan -1, a czerwony +1

$$L=70, T^*=2.269$$

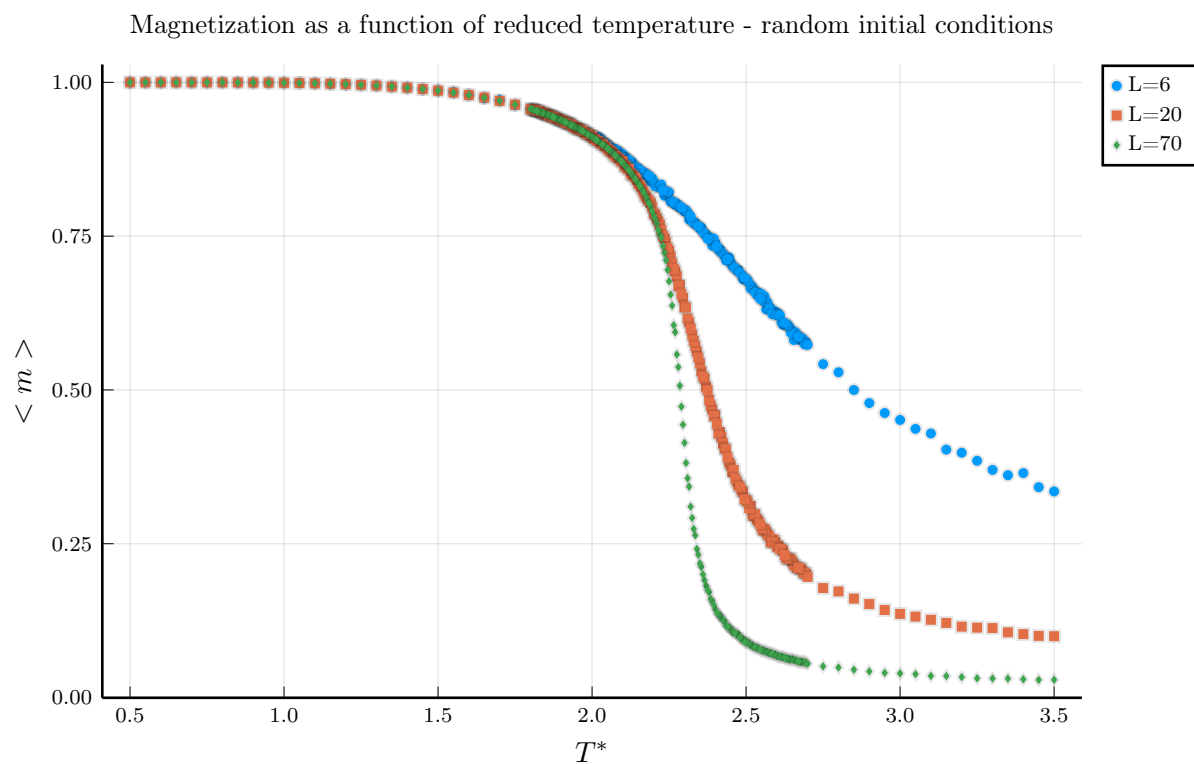


Rysunek 2: Konfiguracja dla $L = 70$ i $T^* = 2.269$ po $430000 MCS$. Kolor niebieski to stan -1, a czerwony +1

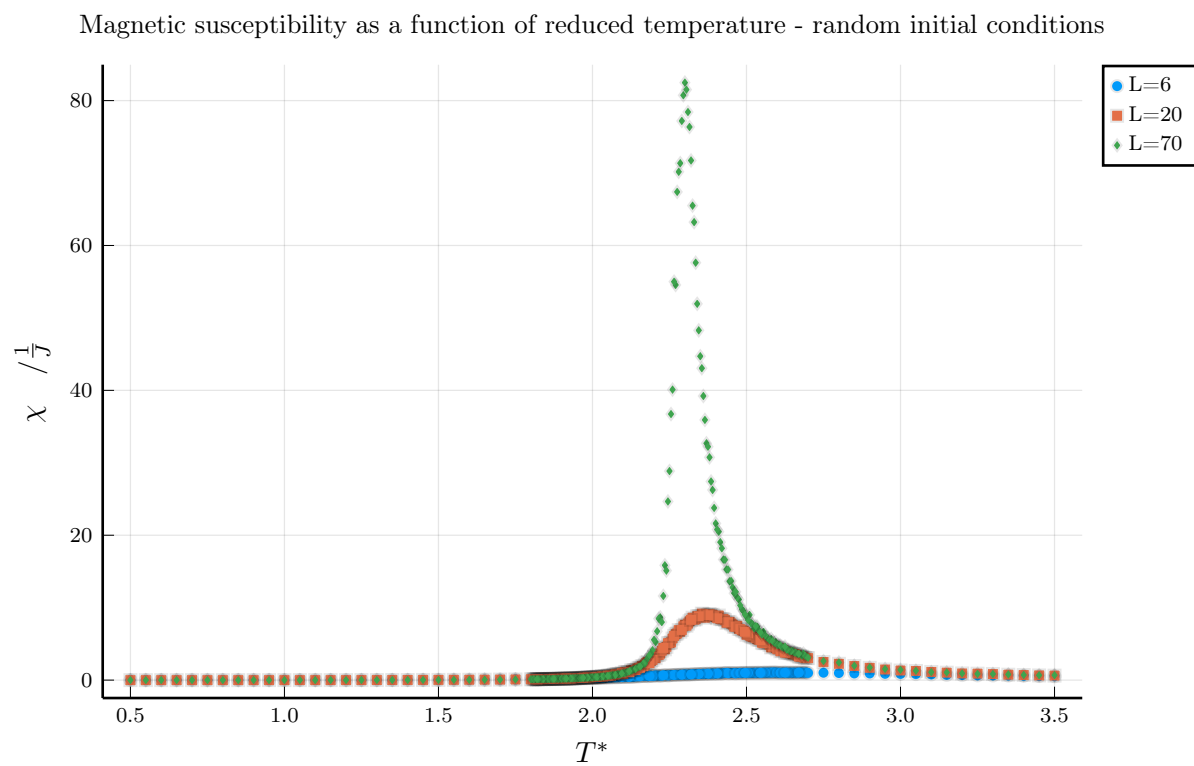
$$L=70, T^*=8.0$$



Rysunek 3: Konfiguracja dla $L = 70$ i $T^* = 8.000$ po $430000 MCS$. Kolor niebieski to stan -1 , a czerwony $+1$

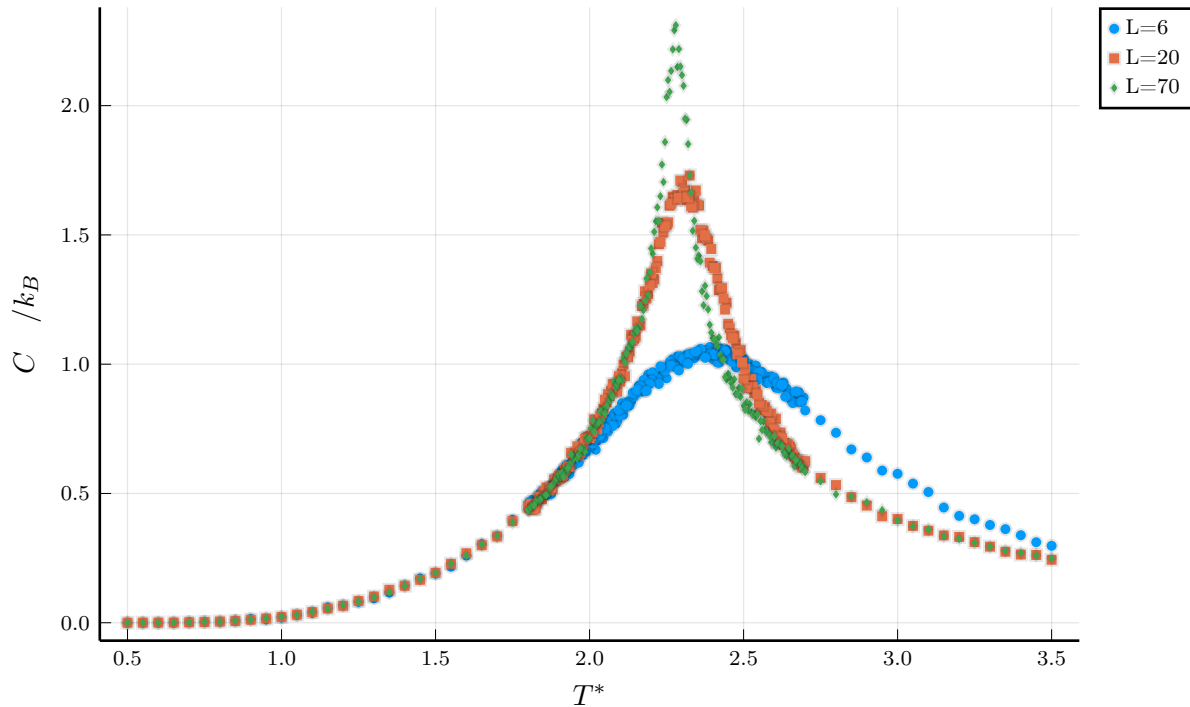


Rysunek 4: Średnia magnetyzacja układu w funkcji zredukowanej temperatury



Rysunek 5: Podatność magnetyczna układu w funkcji zredukowanej temperatury

Heat capacity as a function of reduced temperature - random initial conditions



Rysunek 6: Pojemność cieplna układu w funkcji zredukowanej temperatury

Kod programu (Julia)

```

1 using Statistics
2 using Plots
3 using LaTeXStrings
4 using JLD2
5
6 pgfplotsx()
7 function plot_lattice(lattice::Array{Int8}, T::Number, fn::String = "lattice.tex")
8     lat = heatmap(lattice, aspect_ratio=1.0, size=(800,800), grid=false, framestyle=:box, colormap_name =
9         ↪ "hot", legend=false) #blue=-1, red=1
10    l::Int = size(lattice)[1]::Int
11    hline!(lat, 0.5:1:l+0.5, line=:black)
12    vline!(lat, 0.5:1:l+0.5, line=:black)
13    title!(lat, "L=$l, T=$T")
14    savefig(lat, fn)
15    plot(lat)
16 end
17 function ising(L::Int, T::Number; MCS=230_000::Int, to_plot::Bool=false)
18     #initialize random lattice, and indexes
19     lattice = rand{Int8}[-1,1], L, L)
20     #lattice = ones{Int8}(L,L)
21     next_I = collect(2:L+1)
22     prev_I = collect(0:L-1)
23     next_I[L]=1
24     prev_I[1] = L
25
26     # compute boltzmann factor as Dict

```

```

27 boltzmann_factor = Dict{Int,Float64}( $\Delta E \Rightarrow \exp(-\Delta E / T)$  for  $\Delta E = -8:4:8$ )
28
29 #initial arrays (returned)
30 magnetizations = zeros(0)
31 energies = zeros(0)
32
33 for k=1 : MCS
34     for i=1:L, j=1:L
35         #metropolis algorithm
36          $\Delta E::Int = 2 * lattice[i,j] * (lattice[next\_I[i],j] + lattice[prev\_I[i],j] +$ 
37              $\hookrightarrow lattice[i,next\_I[j]] + lattice[i,prev\_I[j]])$ 
38         w::\Delta E])
39         if rand()  $\leq$  w
40             lattice[i,j] *= -1
41         end
42     end
43
44     if (30_000 < k && k%100 == 0)
45         m_k::\hookrightarrow lattice[i,next\_I[j]] + lattice[i,prev\_I[j]])
51         end
52         push!(energies, E)
53     end
54 end
55
56 if (to_plot)
57     plot_lattice(lattice, T, "lattice_L$(L)_T$(trunc(Int,T)).png")
58 end
59 return magnetizations, energies
60 end
61
62 function run_MC(Ls::Array{Int}, Ts::Array{Float64})
63     result = Dict{Int, Dict{Char, Dict{Float64,Float64}}} (1 => Dict{c => Dict{ } for c in ['m',' $\chi$ ','C']})
64      $\hookrightarrow$  for l in Ls
65         Threads.@threads for (L,T) in collect(Iterators.product(Ls,Ts))
66             magnetizations, energies = ising(L,T,to_plot=false,MCS=430_000)
67
68             result[L]['m'][T] = mean(magnetizations)
69              $\hookrightarrow$  # mean magnetization  $\langle m \rangle$ 
70             result[L][' $\chi$ '][T] = L^2 / T * var(magnetizations, mean=result[L]['m'][T], corrected=false)
71              $\hookrightarrow$  # magnetic susceptibility  $\chi$ 
72             result[L]['C'][T] = 1.0 / L^2 / T^2 * var(energies, corrected=false)
73              $\hookrightarrow$  # heat capacity C
74         end
75     end
76 return result
77 end
78
79 function run_MC_one_thread(Ls::Array{Int},Ts::Array{Float64})
80     for (L,T) in collect(Iterators.product(Ls,Ts))
81         @time ising(L, T, MCS=430_000, to_plot=true)
82     end
83 end
84
85 #= main =#
86 Ls = [6,20,70]

```

```

80 Ts = cat(0.5:0.05:1.75, 1.8:0.005:2.7, 2.75:0.05:3.5, dims=1)
81 result_ret = @time run_MC(Ls,Ts)
82 @save "result_ret_Ls_Ts_430.jld2" result_ret Ls Ts
83
84 #examples
85 Ls = [70]
86 Ts = [0.05, 2.269, 8.00]
87 @time run_MC_one_thread(Ls,Ts)
88
89
90 #= Plots =#
91 result = result_ret
92 plot()
93 for L in Ls
94     plot!(result[L]['m'], markershape=:auto, label="L=$L", markersize=3, markerstrokealpha=0.1)
95 end
96 xlabel!(L"T~*")
97 ylabel!(L"<m>")
98 title!("Magnetization as a function of reduced temperature - random initial conditions",
    ↪ titlefontsize=10)
99 #savefig("magnetization.tex")
100
101 plot()
102 for L in Ls
103     plot!(result[L]['χ'], markershape=:auto, label="L=$L", markersize=3, markerstrokealpha=0.1)
104 end
105 xlabel!(L"T~*")
106 ylabel!(L"\chi")
107 title!("Magnetic susceptibility as a function of reduced temperature - random initial conditions",
    ↪ titlefontsize=10)
108 #savefig("magnetic_susceptitibility.tex")
109
110 plot()
111 for L in Ls
112     plot!(result[L]['C'], markershape=:auto, label="L=$L", markersize=3, markerstrokealpha=0.1)
113 end
114 xlabel!(L"T~*")
115 ylabel!(L"C \quad / k_B")
116 title!("Heat capacity as a function of reduced temperature - random initial conditions",
    ↪ titlefontsize=10)
117 #savefig("heat_capacity.tex")

```
