Spin Ensemble

October 5, 2021

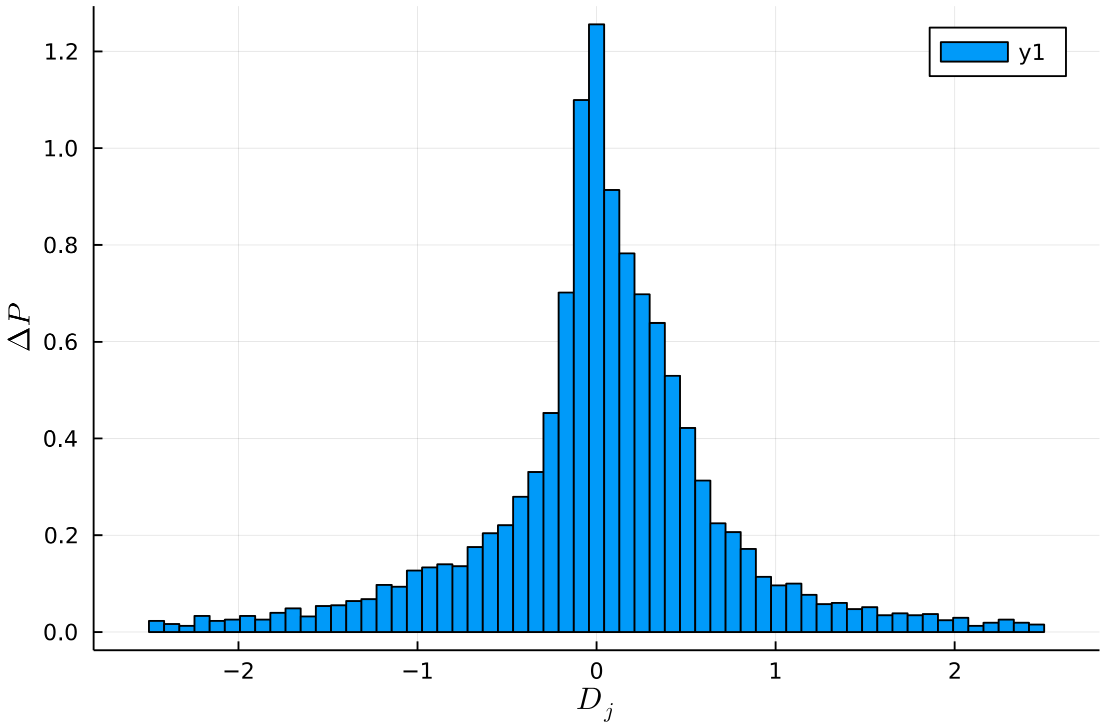
[1]:
```
using Plots
using LaTeXStrings
using Printf
using Statistics
```

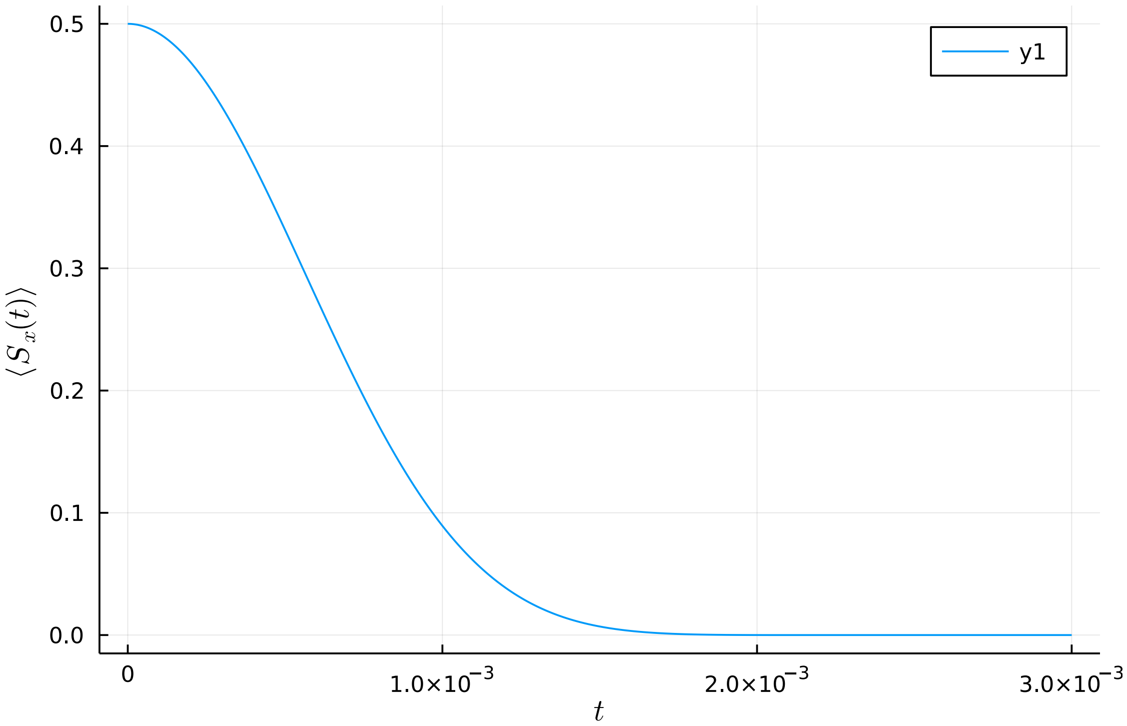
[2]:
```
include("SpinEnsemble.jl")
```

[2]:
```
transverse_threshold
```

[3]:
```
"""
main session of the workflow.
calculate the decaying curve given by a given set of parameter
"""
function main(T::Float64,N::Int, d::Int,a::Float64)
    coeff_sample=rand_dipolar_bath_coefs(N,d,a);
    bin_set=range(-20/a^d, 20/a^d, length = 60)
    fig1=histogram(coeff_sample, bins = bin_set, xlabel=L"D_j", ylabel=L"\Delta␣
 ↪P", norm=true)
    println("max: ",maximum(coeff_sample)," min: ",minimum(coeff_sample))
    time=collect(range(0,T,length=1001))
    decay_curve=ensemble_FID(time,coeff_sample)
    fig2=plot(time,decay_curve,xformatter=:scientific,xlabel=L"t",␣
 ↪ylabel=L"$\langle S_x(t) \rangle$")
    display(fig1)
    display(fig2)
end;


function main(T::Float64,f::Float64, p0::Vector{Float64},d::Int,a::Float64)
    N=floor(Int,a^d*f) # number of spins given by density
    return main(T,N,d,a)
end;
```

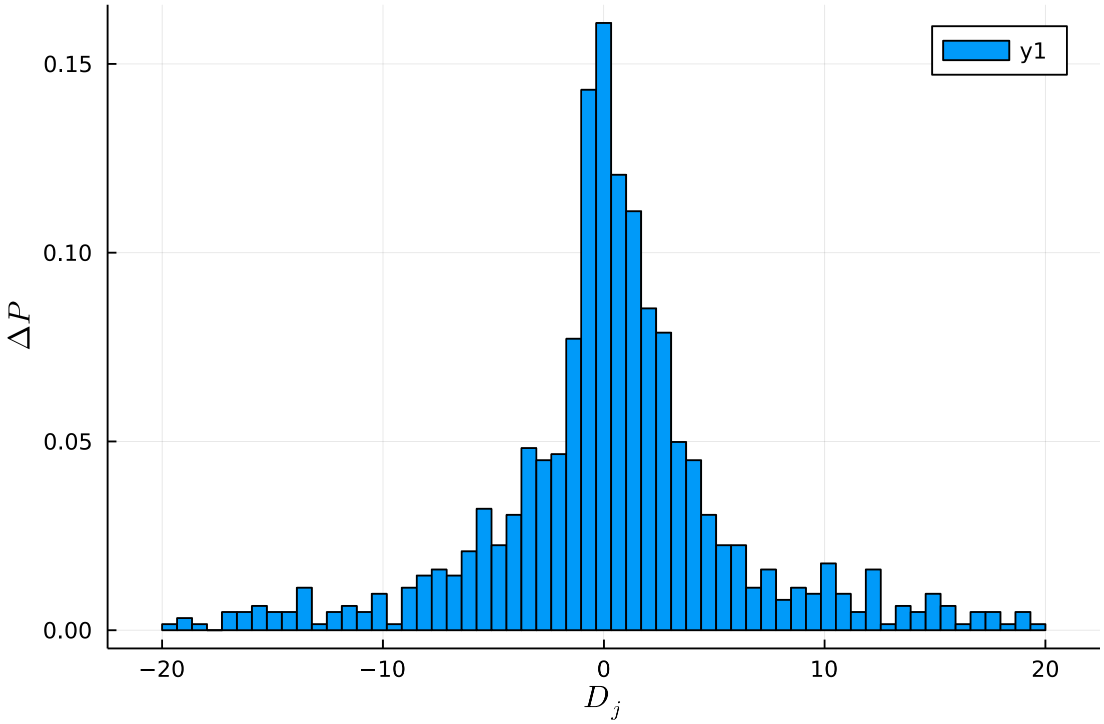
[64]:
```
main(3e-3,10^4,3,2.0)
```

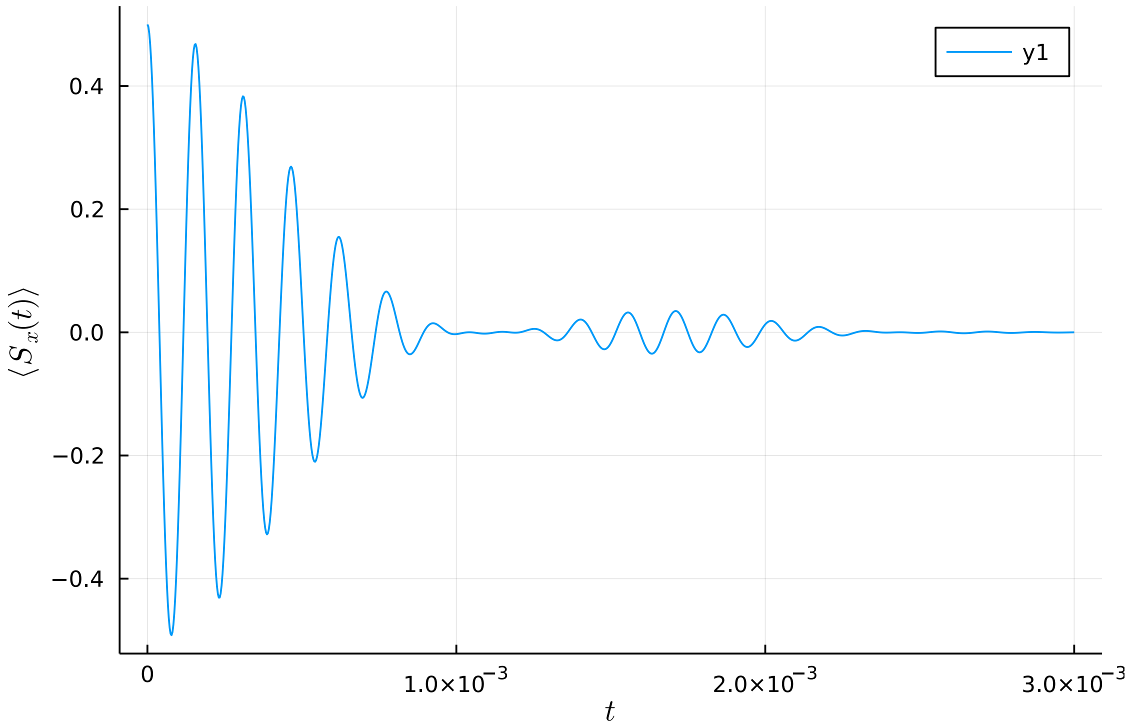

y1
1.2
1.0
0.8
0.6
0.4
0.2
0.0
ΔP
−2
−1
0
1
2
D_j


y1
0.5
0.4
0.3
0.2
0.1
0.0
$\langle S_x(t) \rangle$
0
1.0×10^{-3}
2.0×10^{-3}
3.0×10^{-3}
t

```
max: 760.2327949131752 min: -548.146043124062
```

```
[73]: main(3e-3,10^3,3,1.0)
```



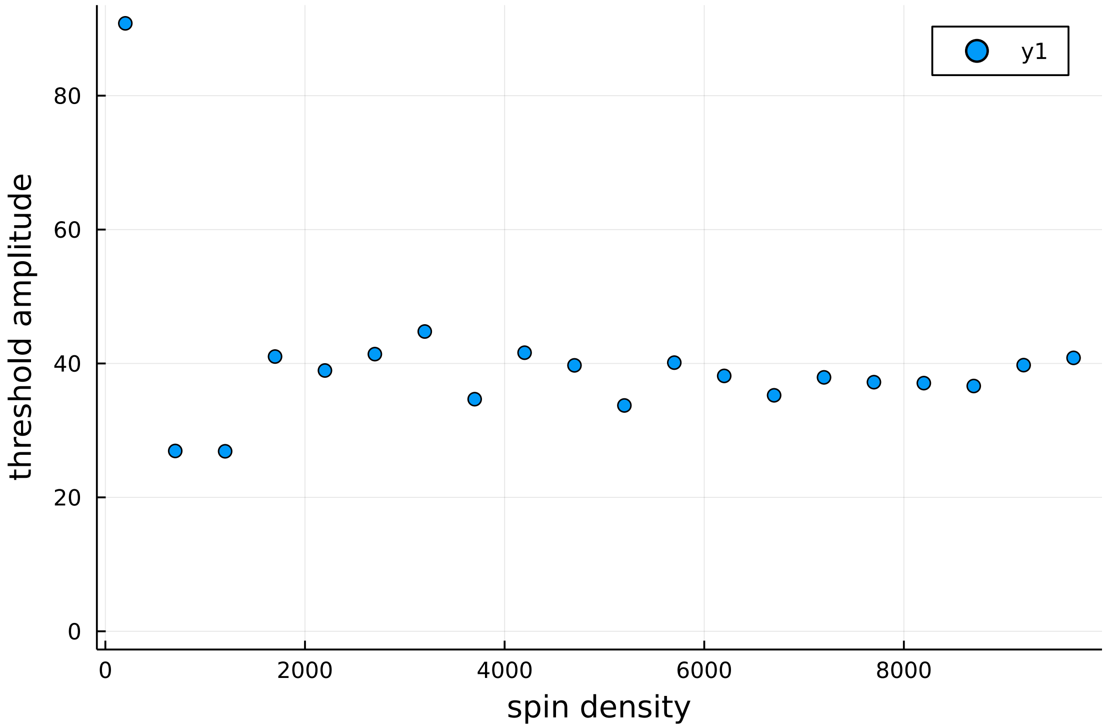


```
max: 40408.95451533921 min: -1517.5703478780365
```

[96]:

```
f=[100.0*n for n in 2:5:100]
h_s=map(x->transverse_threshold(0.95,x,2,1.0),f)
scatter(f, h_s,
yrange=[0,maximum(h_s)],
xlabel="spin density",
ylabel="threshold amplitude")
```

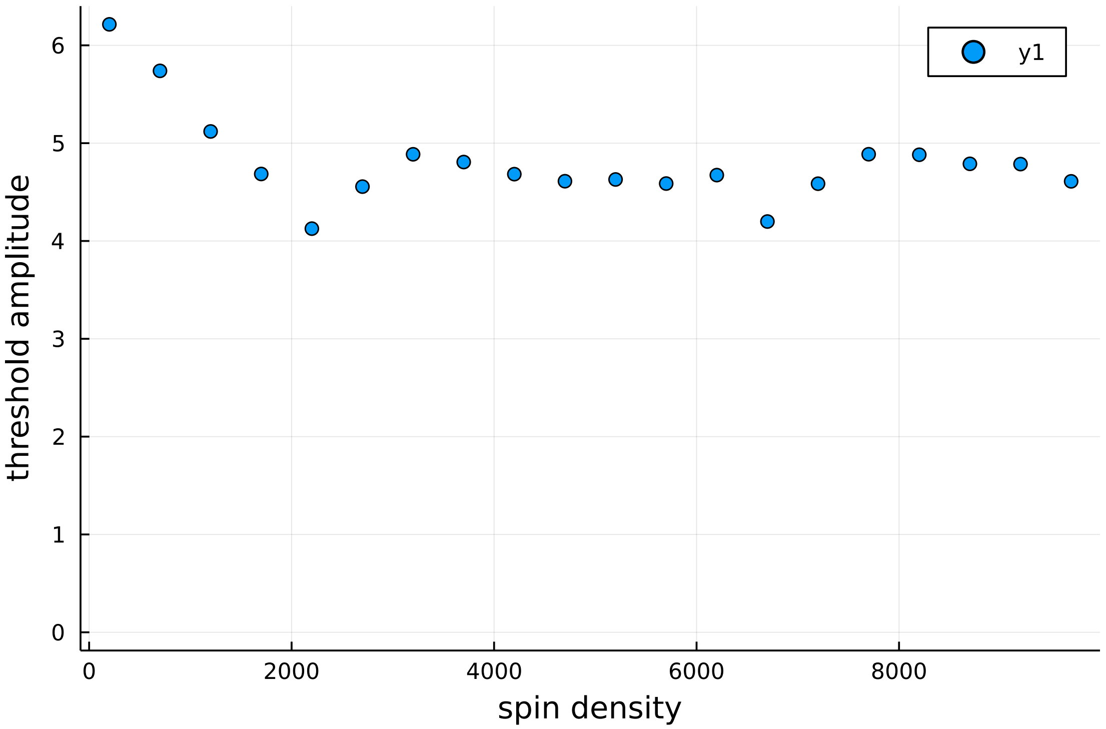
[96]:



[97]:

```
f=[100.0*n for n in 2:5:100]
h_s=map(x->transverse_threshold(0.95,x,2,2.0),f)
scatter(f, h_s,
yrange=[0,maximum(h_s)],
xlabel="spin density",
ylabel="threshold amplitude")
```

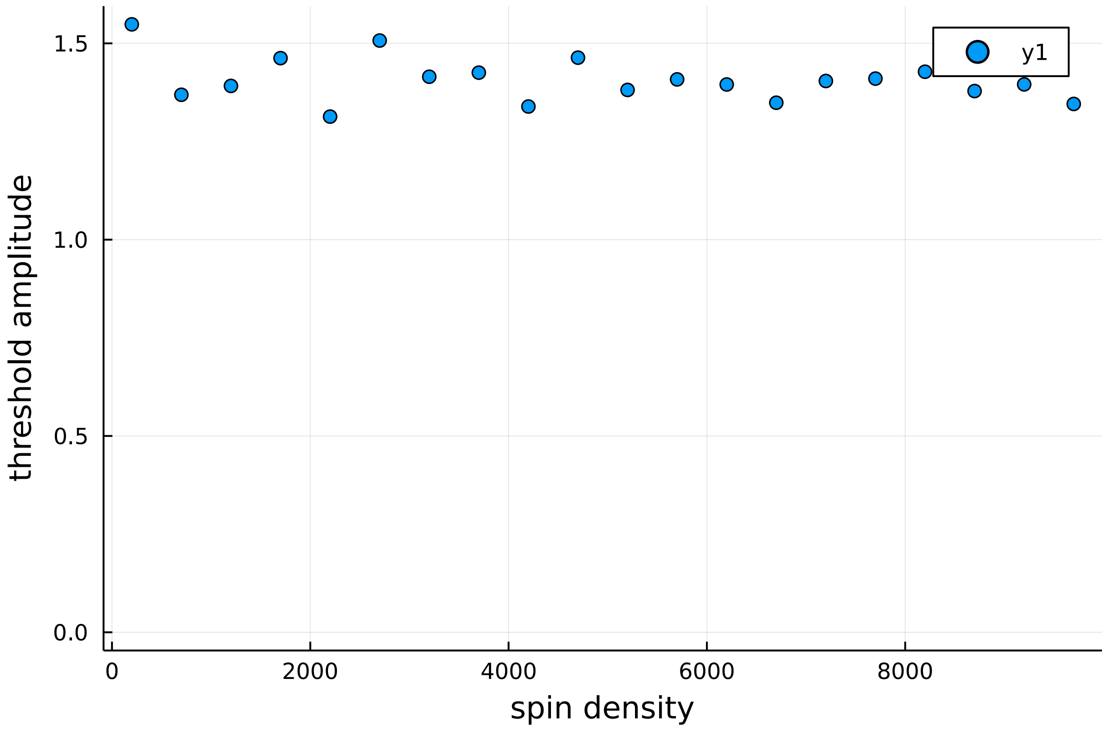
[97]:



[98]:

```
f=[100.0*n for n in 2:5:100]
h_s=map(x->transverse_threshold(0.95,x,2,3.0),f)
scatter(f, h_s,
yrange=[0,maximum(h_s)],
xlabel="spin density",
ylabel="threshold amplitude")
```

[98]:



[52]:

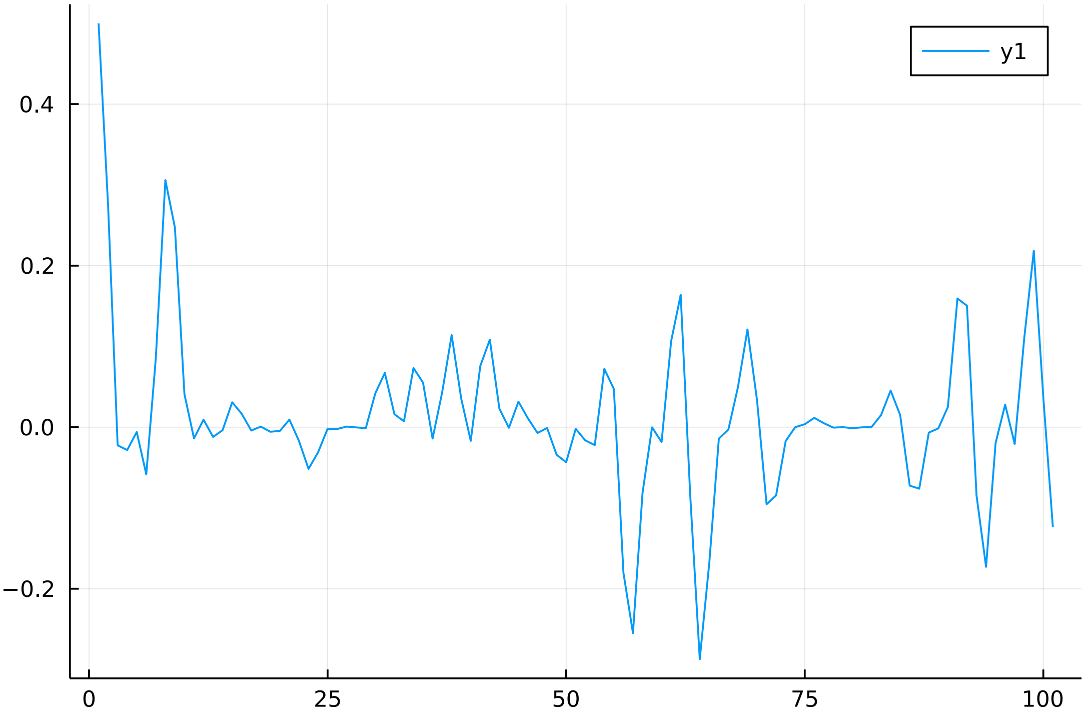
```
b=SpinBasis(1//2)
Sz=sigmaz(b)
Sx=sigmax(b)
Si=Sz*Sz

"""
contruct the Hamiltonian of the ensemble,
Args:
    D: dipolar coupling strength
    H_R: transerve magnetic field
"""
function ensemble_Hamiltonian(D::Vector{Float64},H_R::Float64)
    N=length(D)
    L=repeat([Si],N)

    H=H_R*tensor(Sx,L...)
    for i in 1:N
        L[i]=Sz
        H+=D[i]*tensor(Sz,L...)
        L[i]=Si
    end
    return H
end
```

[52]: ensemble_Hamiltonian

[73]:
```
N=5
D=rand_dipolar_bath_coefs(N,3,1.0);
H=ensemble_Hamiltonian(D,0.0);
```

[74]:
```
T=0:0.1:10
# plot the decay curve using the direct production
plot(ensemble_FID(collect(Float64,T),D))
```

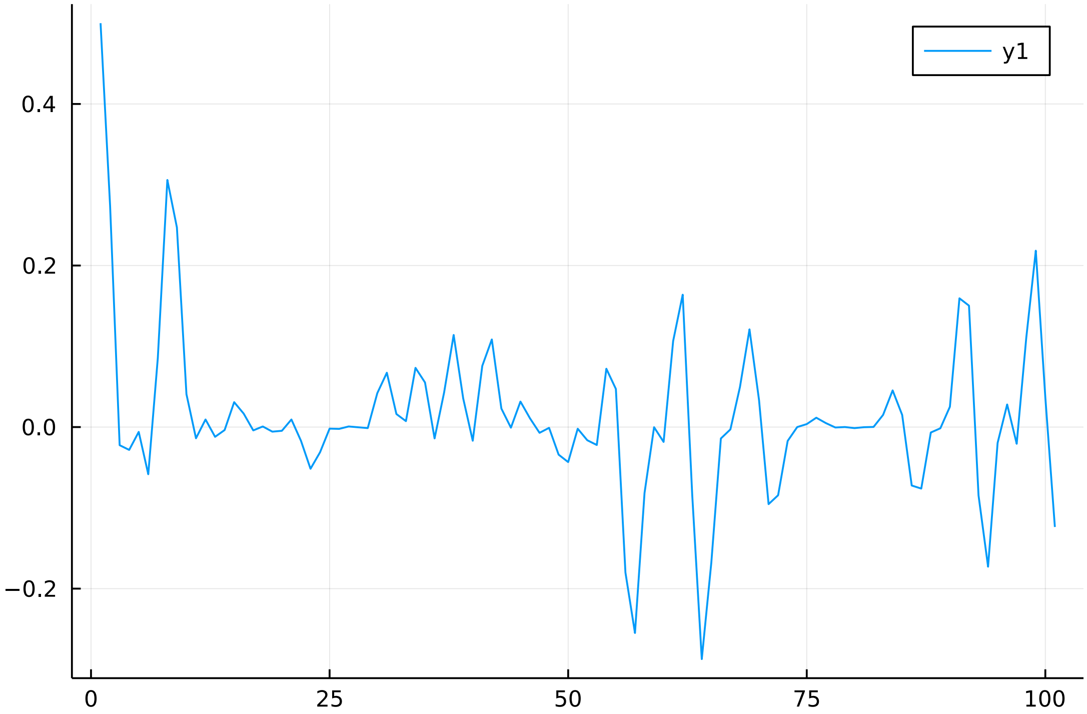
[74]:



[75]:
```
# plot the decay curve by simulating the time evolution of the composite system
Sx_0=tensor(Sx/2,repeat([Si],N)...)
psi0=tensor(Si/2+Sx/2,repeat([Si/2],N)...)
exp_func(t, psi) = expect(Sx_0, psi)
tout, exp_val = timeevolution.schroedinger(T, psi0, H; fout=exp_func);
```

[76]:
```
plot(real(exp_val))
```

[76]:


y1
0.4
0.2
0.0
−0.2
0
25
50
75
100