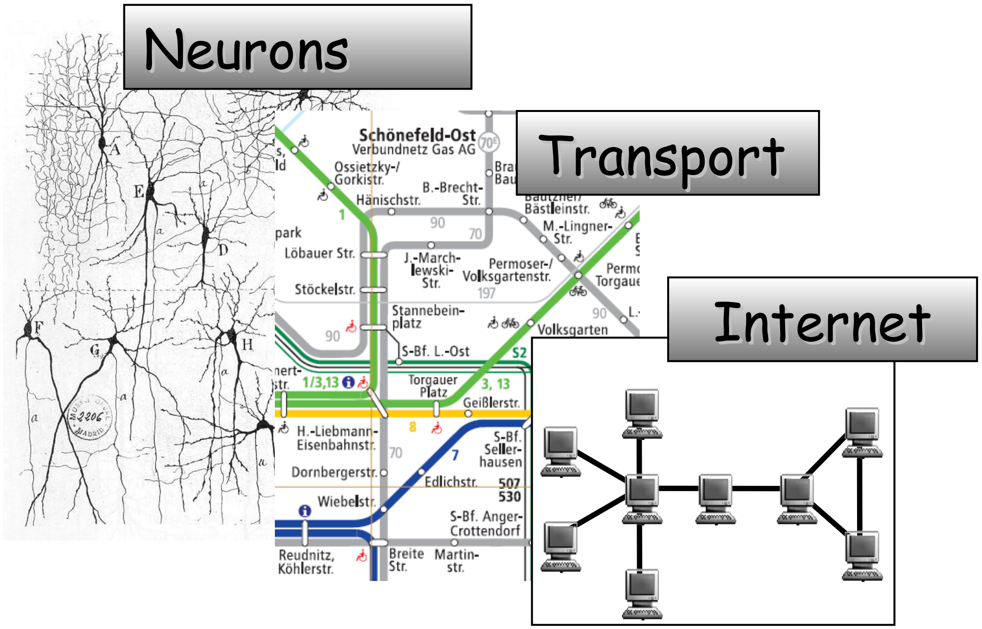
Multicanonical simulation of networks

B. Waclaw[1], L. Bogacz[2], W. Janke[1]

[1]Institut für Theoretische Physik, Universität Leipzig, Germany

[2]Department of Physics, Jagellonian University, Kraków, Poland

Random networks



different networks

modelled as random graphs

How to generate them?

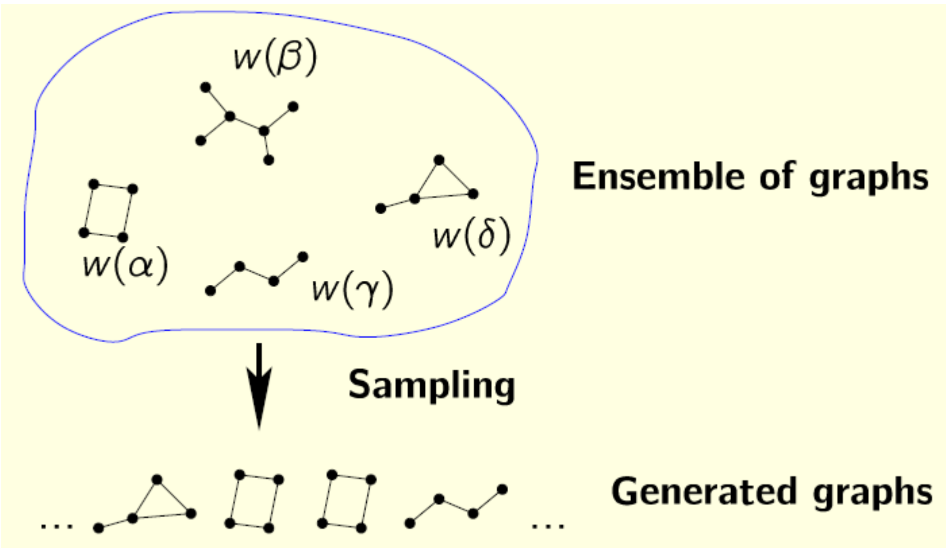
Different methods:

- "physical processes": growth, rewiring of connections, interactions between agents etc.
- "unphysical" but producing desired features much faster

our algorithm

Our method

- define a statistical ensemble
- sample graphs using a Monte Carlo procedure



and then...

- make measurements
- simulate some processes on networks
- etc...

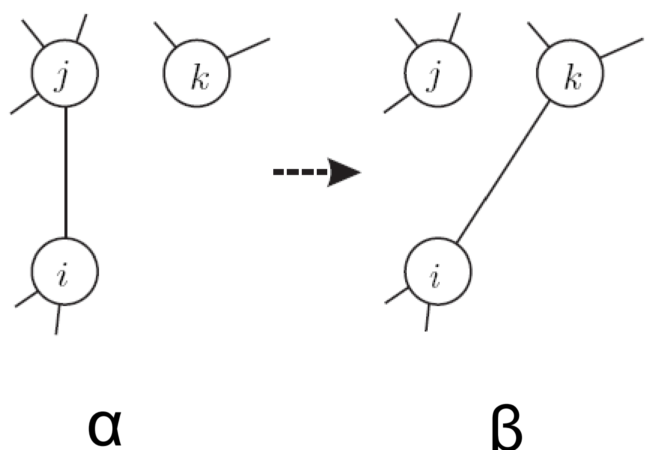
Advantages: wide spectrum, fast, advanced MC methods can be used

More details...

- At each time step a new network β is produced by a small change of the previous one α
- Metropolis algorithm: we accept the new configuration with probability

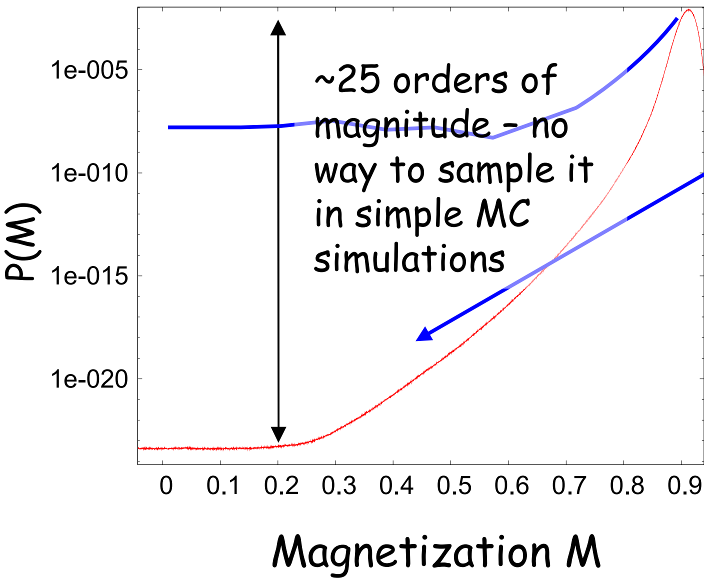
$$P_{\alpha\beta} = \min\left\{1, \frac{W_\beta}{W_\alpha}\right\}$$

- "Small changes" = rewiring of links



Multicanonical simulations

- sometimes we need to measure rare events (e.g. P(M≈0) in the Ising model)
- they are rare → small probability → poor statistics



The idea: increase probability here

$$e^{-\beta E} \rightarrow e^{-\beta E} r(M)$$

r(M) is some reweighting factor

Multicanonical simulation (MUCA)

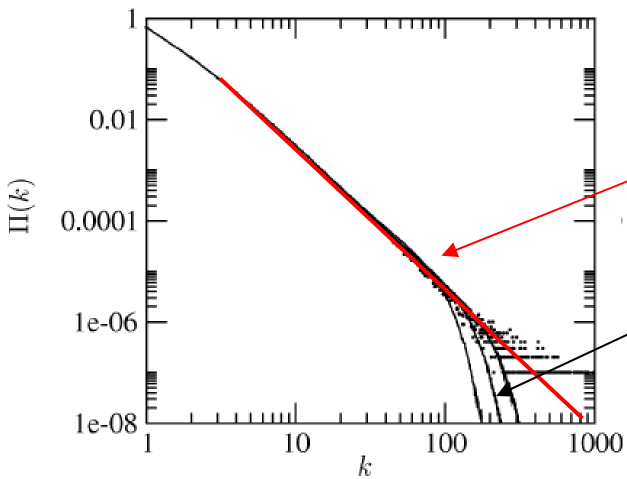
Procedure:

- Do the simulation with modified weights
- Measure quantities of interest
- Do "reweighting" to get averages in the original canonical ensemble

MUCA in networks

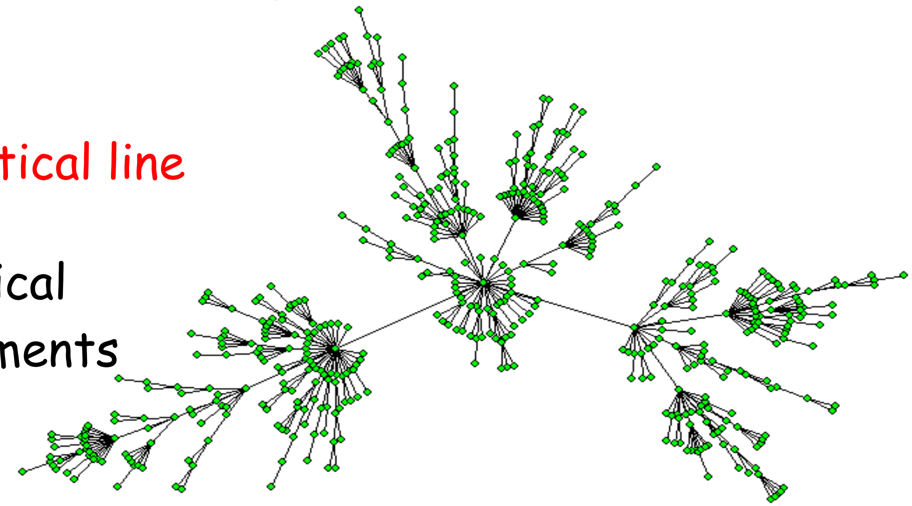
Sampling scale-free networks:

Degree distribution $\Pi(k)$ = prob. that a node has k neighbors



Theoretical line

Numerical experiments



exp. observed distribution

scaling exponent (depends on network)

$$\Pi_{exp}(k) \approx \Pi_{th}(k) \times w(k/N^{\alpha})$$

distribution for ∞ network

some cutoff function

$$\boxed{k_{max} \sim N^{\alpha}}$$

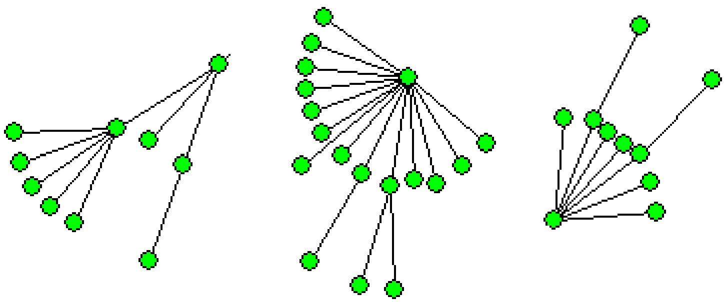
scaling of max. degree

We want to

1) measure α, 2) find the cutoff w(x)

Standard canonical simulation of networks:

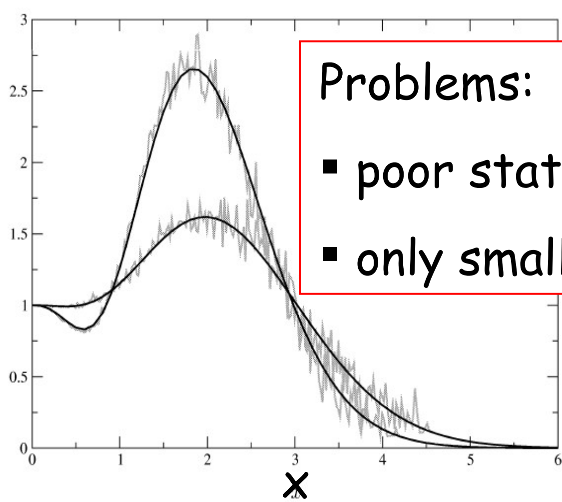
1) Generate graphs with weights $W(g)$



g1 g2 g3 . . . .

2) Measure $\Pi(k)$ averaged over graphs

3) Calculate $w(x)$ and α from data for different sizes



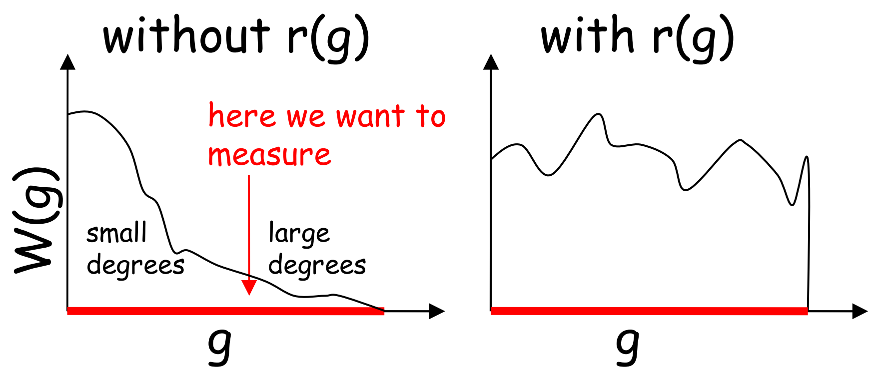
Problems:

- poor statistics
- only small graphs

Multicanonical simulation

1) generate graphs with weights $W(g) \cdot r(g)$, with some function $r(g)$

$r(g)$ chosen to increase the probability of rare events – in the tail



2) Calculate $\Pi(k)$ with „reweighting factor" $1/r(g)$:

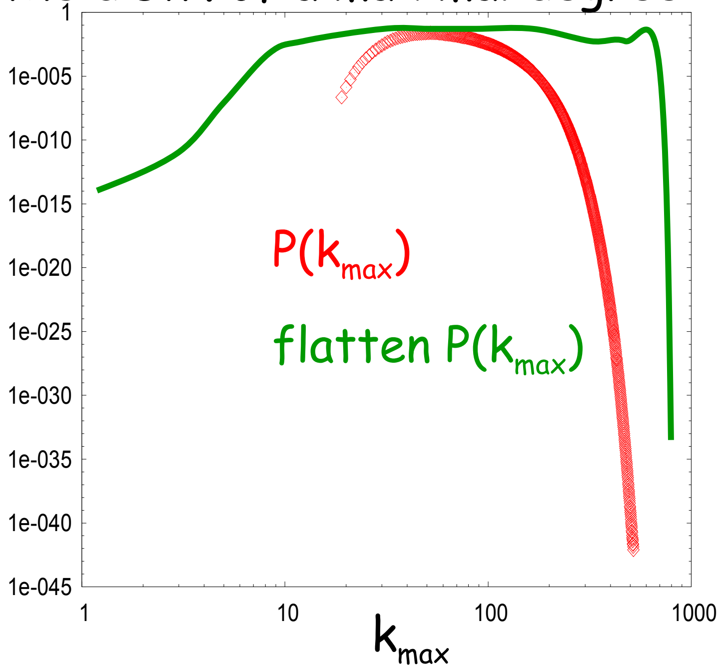
$$\Pi(k)=\Sigma_g \, \Pi_{measured}(k,g)/r(g)$$

Factors $r(g)$ cancel out and we get $\Pi(k)$ but with much better statistics in the tail!

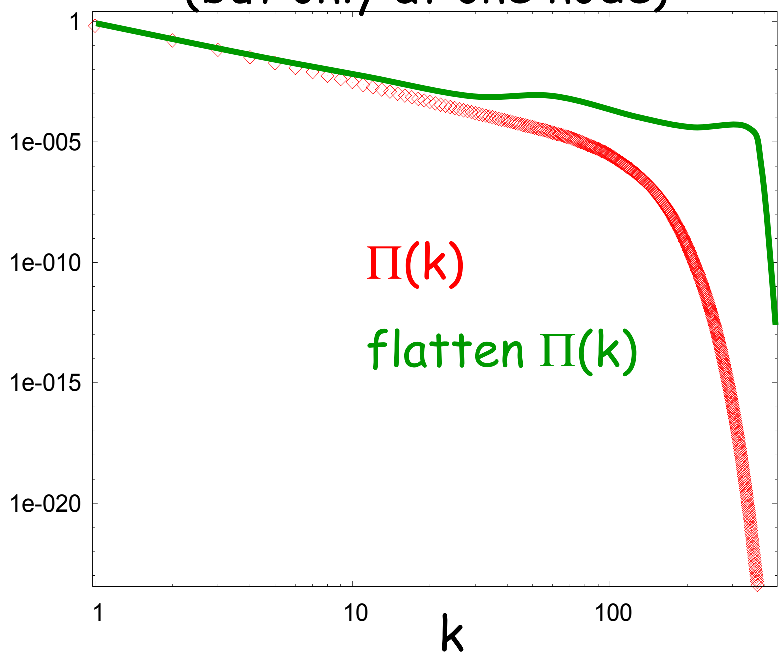
How to realize this procedure in practice, i.e. how r(g) should depend on graph's structure?

Two methods:

Indirect: Flattening $P(k_{max})$ – the distr. of a maximal degree



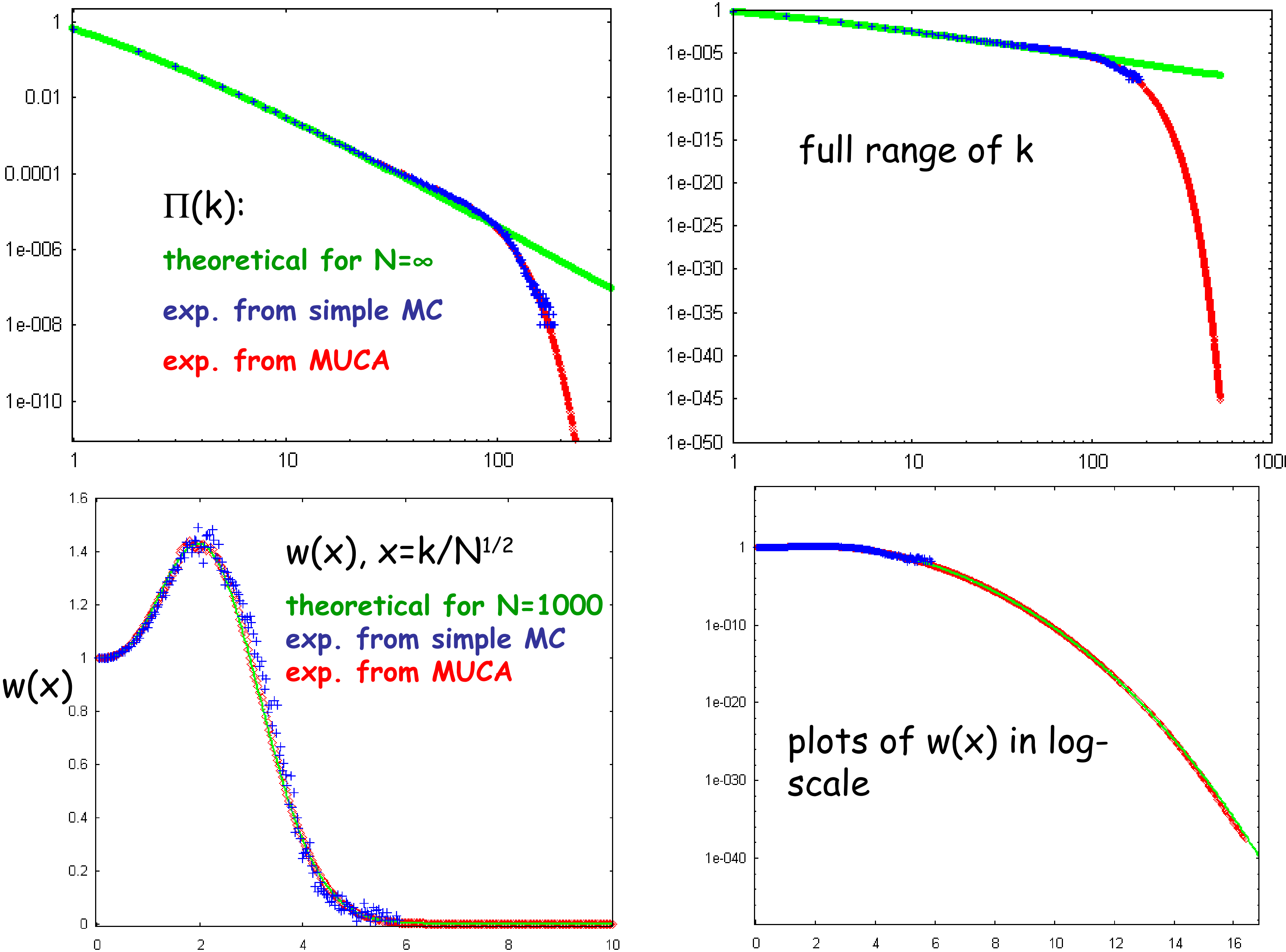
Direct: Flattening $\Pi(k)$ (but only at one node)

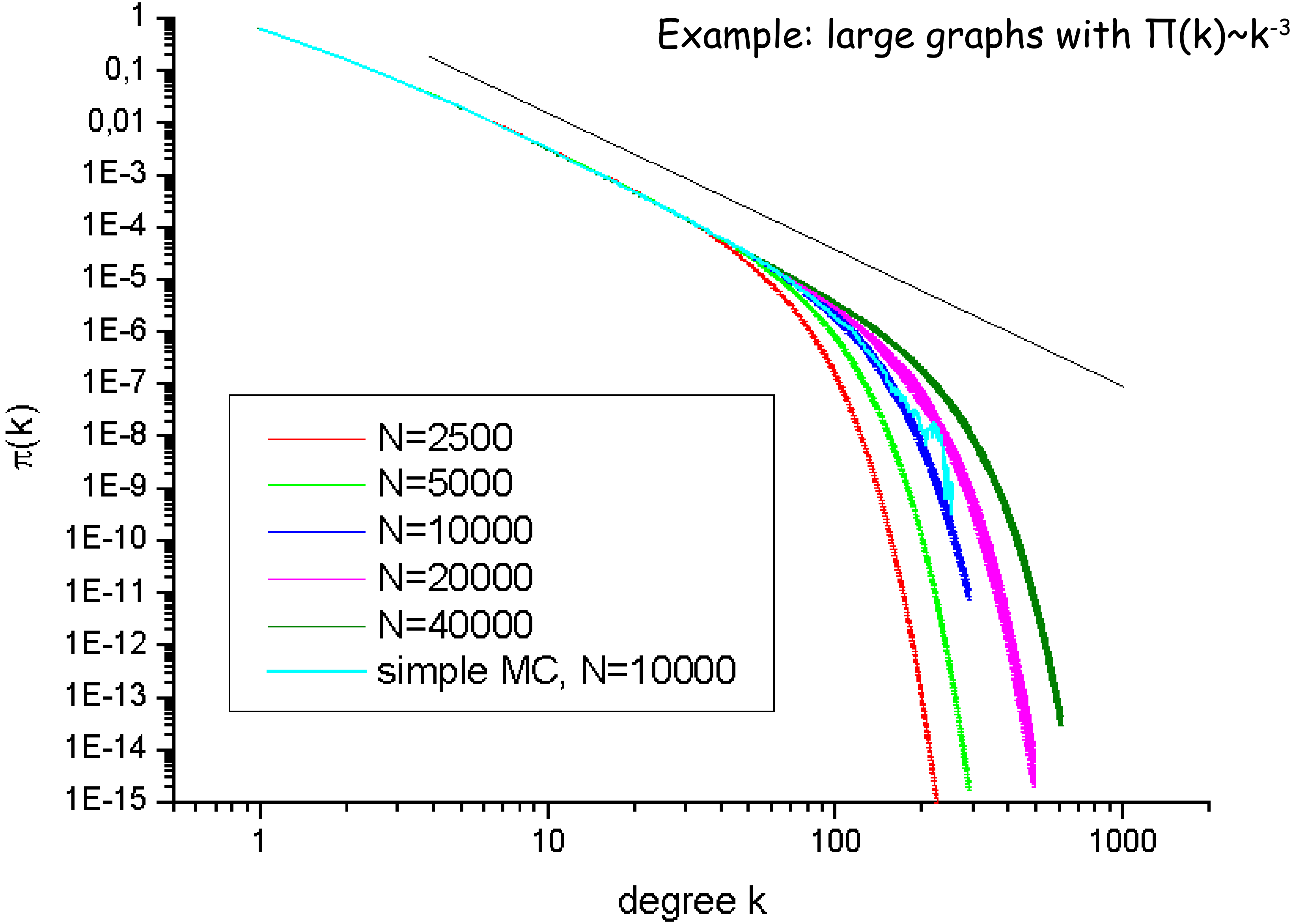


Algorithm:

- local moves like in simple MC, but weights multiplied by r(k) or $r(k_{max})$
- multicanonical recursion (Wang-Landau or W. Janke)
- range of k divided into sub-ranges – speed up simulations – then glueing results

Example: a graph with $\Pi(k) \sim k^{-3}$ and N=1000 nodes





Example: large graphs with Π(k)~k^-3
π(k)
degree k
1
0,1
0,01
1E-3
1E-4
1E-5
1E-6
1E-7
1E-8
1E-9
1E-10
1E-11
1E-12
1E-13
1E-14
1E-15
1
10
100
1000
N=2500
N=5000
N=10000
N=20000
N=40000
simple MC, N=10000