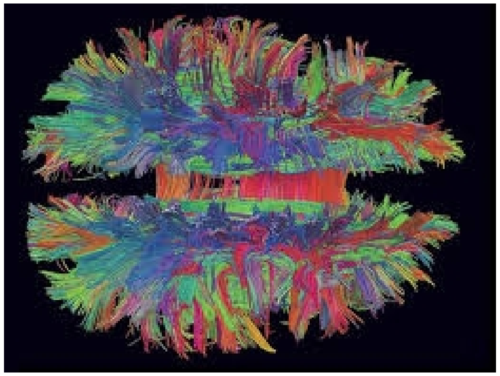
Critical synchronization dynamics of the Kuramoto model on a large human connectome

Géza Ódor EK-MFA Complex Systems Department, Budapest
Jeffrey Kelling HZDR Dresden, Gustavo Deco UPF Barcelona

Theoretical research and experiments suggest that the brain operates at or near a **critical state** between sustained activity and an inactive phase, exhibiting optimal computational properties (see: *Beggs & Plenz J. Neurosci. 2003; Chialvo Nat. Phys. 2010; Haimovici et al. PRL 2013*)

Individual neurons emit periodic signals: *(Y. Penn et al PNAS 113 (2016) 3341)*

→ Criticality at the **synchronization transition point ?**



Kuramoto oscillator model (1975) on a large human connectome

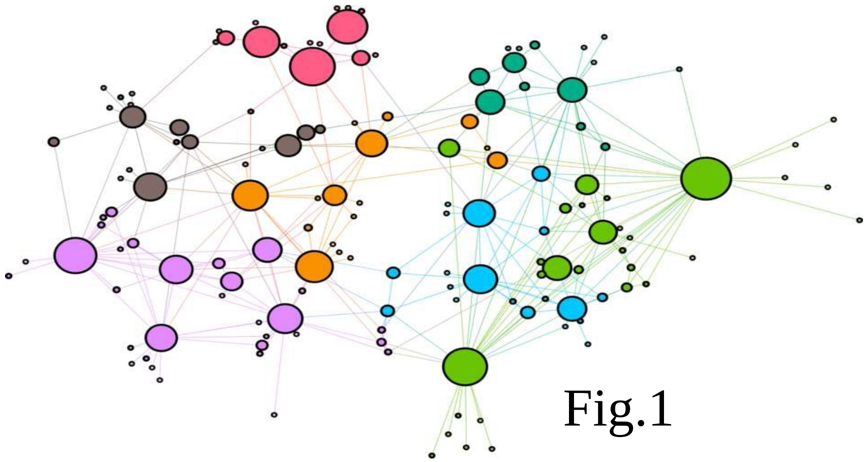


Fig.1

$$\dot{\theta}_i(t) = \omega_{i,0} + K \sum_j W_{ij} \sin[\theta_j(t) - \theta_i(t)]$$

phases $\theta_i(t)$

global coupling K is the control parameter

weighted adjacency matrix W_{ij}

$\omega_{i,0}$ is the intrinsic frequency of the i-th oscillator,

Order parameter : average phase: $$R(t) = \frac{1}{N} \left| \sum_{j=1}^{N} e^{i\theta_j(t)} \right|$$

$R(t \to \infty) > 0$ for $K > K_c$, $R(t \to \infty) = 0$ for $K \leq K_c$ as $R \propto (1/N)^{1/2}$

Exhibits an initial growth: $R(t,N) = N^{-1/2} t^{\eta} f_{\uparrow}(t/N^{\tilde{z}})$ from incoherent initial states

The ***KKI-18*** is a structural graph of $N \sim 8 \times 10^5$ nodes and $\sim 4 \times 10^7$ power-law distributed weighted links see *: Michael T. Gastner and Géza Ódor, Scientific Reports 6 (2016) 27249*

Dynamical scaling and frustrated synchronization sub-critically, see:

Géza Ódor and Jeffrey Kelling :
Critical synchronization dynamics of the Kuramoto model on connectome and small world graphs
Scientific Reports 9 (2019) 19621

Determination of the characteristic time exponent τ_t

Measure characteristic times t_x of first dip below: $R_c = (1/N)^{1/2}$

average over ~*10.000* independent ω_i distribution realizations

Histogramming of t_x at the transition point

Critical exponent: $\tau_t = 1.2\,(1)$ at $K_c \simeq 1.6$

Below the transition point : $K < 1.6$

non-universal power laws in the range of experiments of activity durations :

$1.5 < \tau_t < 2.4$ (*Palva et al 2013*)

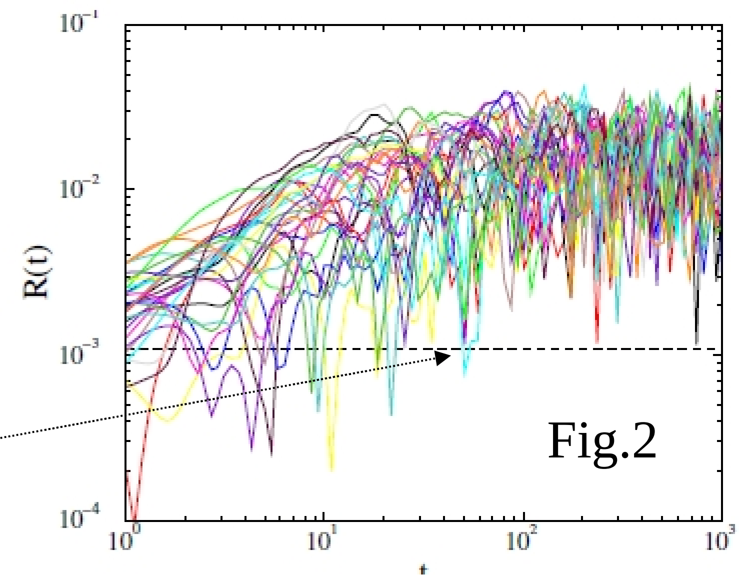


Fig.2

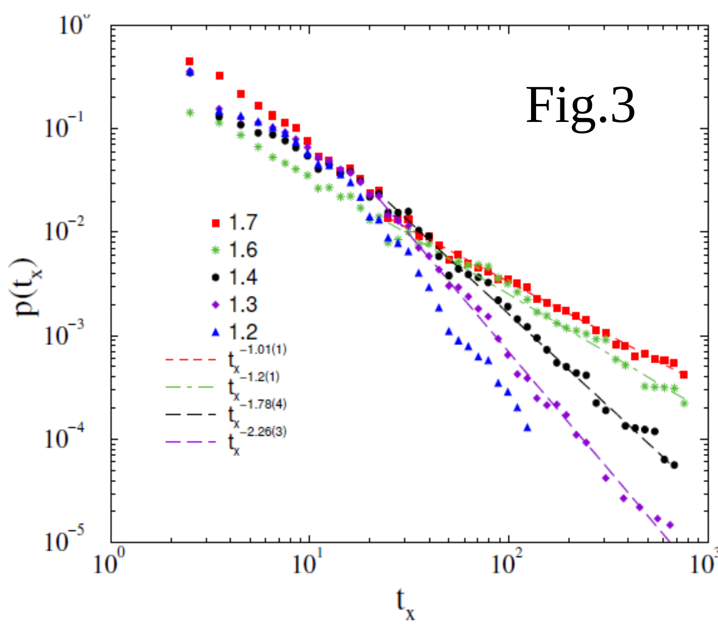


Fig.3

The effect of additive stochastic noise

Brain experiments: $\omega_i > 0$, distributions are narrow: $\sigma_i \sim 0.02$

and have mean value: $<\omega_i> \sim 0.05$

$<\omega_i> \neq 0$ can be gauged out by a rotating coordinate system

Rescaling of ω_i as : $\omega_i \rightarrow a\omega_i{}'$ $t \rightarrow (1/a)\, t'$ $K \rightarrow aK'$

Existing results can be transformed for later times and weaker couplings, thanks to Galilean invariace of the Kuramoto eq.

Gaussian distributed annealed noise is added:

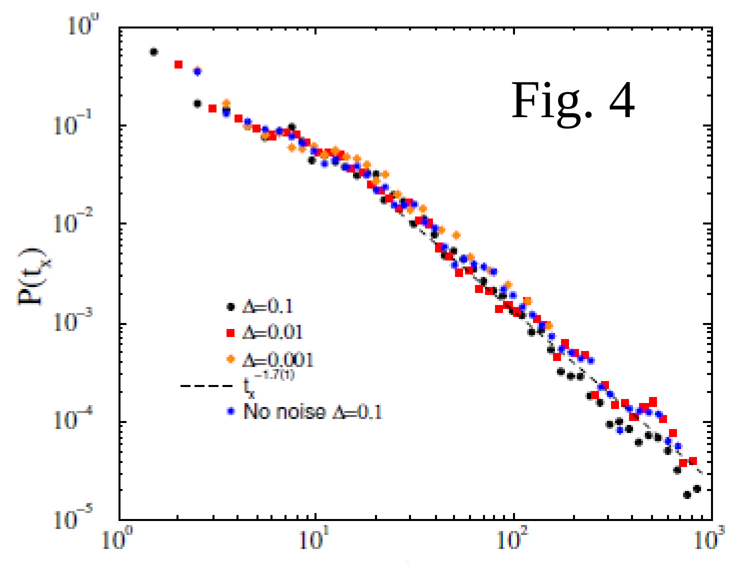
$$\dot{\theta}_i(t) = \omega_{i,0} + K \sum_j W_{ij} \sin[\theta_j(t) - \theta_i(t)] + s\xi(i)$$

Negligible effect of the weak noise

G. Ódor, J.Kelling, G. Deco:

To appear in J. Neuroscience

arXiv:1912.06018



Kuramoto equation solution on connectome

$$\frac{\partial \phi_j(t)}{\partial t} = \omega_j + \sum_{k \text{ NN of } j} \lambda_{jk} \cdot \sin\left[\phi_k(t) - \phi_j(t)\right]$$

- sparse, random graph
  - requires explicit storage network topology
    i.e. sparse representation, neighbor lists
  - random neighbor sums

⇒ **techniques for SIMT vectorization by tuned operation and memory ordering**

Implementation

- `boost::numeric::odeint` odeint.com
  - template library of ODE solvers
  - `boost::numeric` supports various vector backends for accelerators: e.g. Thust (CUDA), VexCL (CUDA/OpenCL)
- VexCL
  - library for offloading vector expressions via CUDA or OpenCL
  - direct support for custom kernels

- we use 4th order Runge-Kutta form `odeint`

⇒ computing derivates reamins and is the most time-consuming part

Efficiency

$$\frac{\partial \phi_j(t)}{\partial t} = \omega_j + \sum_{k \text{ NN of } j} \lambda_{jk} \cdot \sin\left[\phi_k(t) - \phi_j(t)\right]$$

- profile on tesla P100
  - global load efficiency: $\sim 47\,\%$
    saturating gross load bandwidth to $\sim 70\,\%$
  - data requests dominant stall reason $\sim 50\,\%$

$\Rightarrow$ remains memory-latency bound, due to random accesses to neighbors

- efficient inmplementation for integration on random graphs $\sim 20\times$ improved throughput over single CPU socket.
- easily adaptable to other models: we use it for 2nd order Kuramoto, too

Summary

Jeffrey Kelling implemented Kruramoto ODE solver running efficiently on GPU- even in case of sparse random graphs

My experience on Marenostrum-4 BSC Barcelona Configuration:

2 x IBM Power9 8335-GTH @ 2.4GHz (3.0GHz on turbo, 20 cores and 4 threads/core, total 160 threads per node)

4 x GPU NVIDIA V100 (Volta) with 16GB HBM2

Abut a speedup of factor x100 with respect 3GHz CPU-s

Allowed showing lack of effects of weak thermal fluctuations in 2 weeks.

Support from HPC-Europa3 programme and OTKA is acknowledged

Publication in J. Neuroscience is scheduled, a texnical paper to be written

Continuation to study on exact fruit-fly connectome is founded