



UNIVERSITY OF HELSINKI

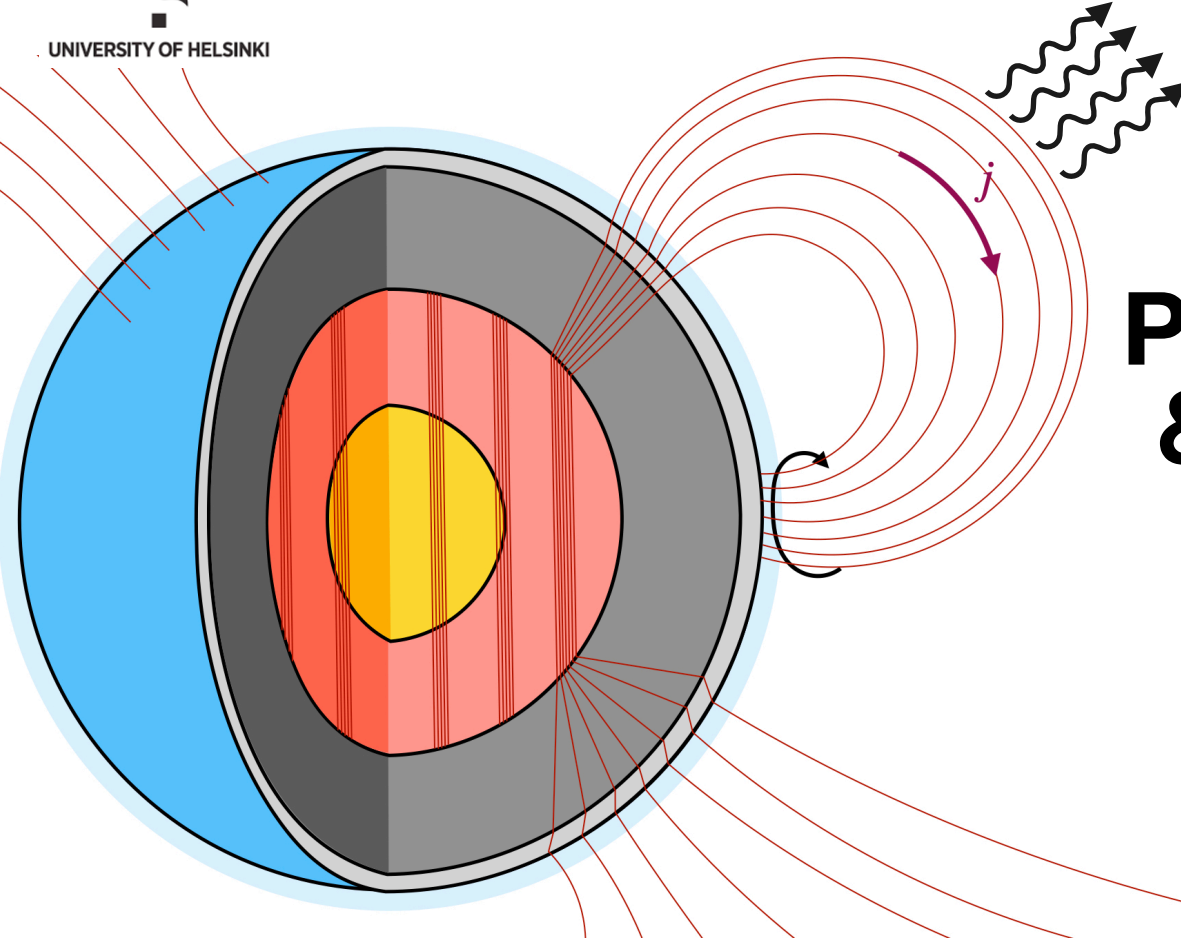


Centre of Excellence  
**neutronstars**.fi

Suomen Akatemia  
Finlands Akademi  
Research Council of Finland



European  
Research  
Council



# Pair discharges & radio waves from pulsars

Joonas Nättilä

# Extreme astrophysical plasmas

$$\sigma = \frac{\text{magnetic energy density}}{\text{plasma rest mass}} = \frac{B^2}{4\pi\rho c^2}$$

$$\frac{v_{\text{wave}}}{c} = \sqrt{\frac{\sigma}{\sigma + 1}}$$

$\sigma \sim 0.1$



Solar corona

$\sigma \sim 1$



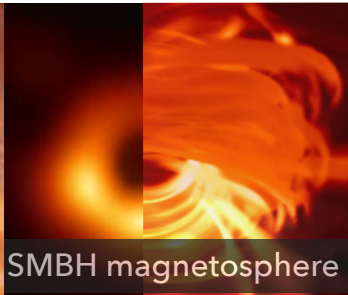
PWNs

$\sigma \sim 1$



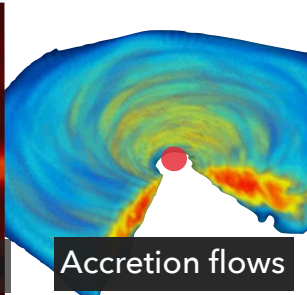
GRBs

$\sigma \sim 1$



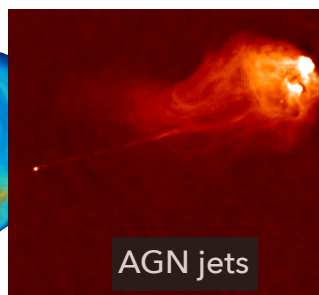
SMBH magnetosphere

$\sigma \sim 10$



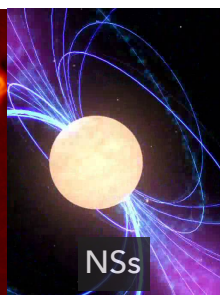
Accretion flows

$\sigma \gtrsim 100$



AGN jets

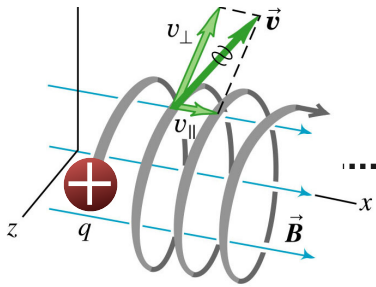
$\sigma > 10^{18}$



NSs

# First-principles modeling of plasmas

## Kinetic scales



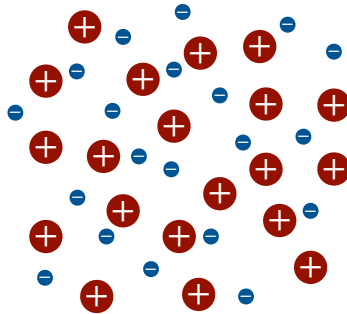
Lorentz + Maxwell's

$$\frac{d(\gamma \mathbf{v})}{dt} = \frac{q}{m} \left( \mathbf{E} + \frac{\mathbf{v}}{c} \times \mathbf{B} \right)$$

$$\frac{\partial \mathbf{E}}{\partial t} = c \nabla \times \mathbf{B} - 4\pi \mathbf{j}$$

$$\frac{\partial \mathbf{B}}{\partial t} = -c \nabla \times \mathbf{E}$$

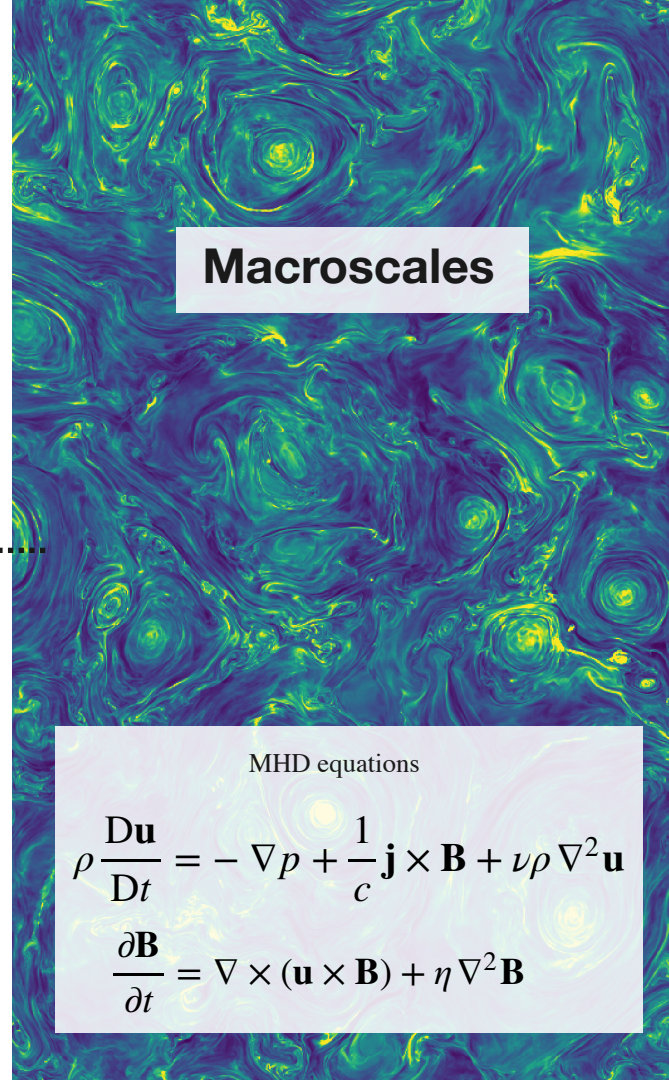
## Mesoscales



Vlasov equation

$$\frac{\partial f}{\partial t} + \frac{\mathbf{u}}{\gamma} \cdot \nabla_{\mathbf{x}} f + \frac{\mathbf{F}}{m} \cdot \nabla_{\mathbf{u}} f = 0$$

## Macroscales



MHD equations

$$\rho \frac{D\mathbf{u}}{Dt} = -\nabla p + \frac{1}{c} \mathbf{j} \times \mathbf{B} + \nu \rho \nabla^2 \mathbf{u}$$

$$\frac{\partial \mathbf{B}}{\partial t} = \nabla \times (\mathbf{u} \times \mathbf{B}) + \eta \nabla^2 \mathbf{B}$$

# Numerical modeling of plasmas

[github.com/hel-astro-lab](https://github.com/hel-astro-lab)

**runko** Public

Modern C++/python CPU/GPU plasma toolbox

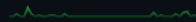
Python 59 GPL-3.0 22 10 1 Updated 4 hours ago



**corgi** Public

C++17 template library for massively parallel computational grids

C++ 5 Apache-2.0 3 0 0 Updated 2 weeks ago



**tyvi** Public

HPC C++ parallelization library.

C++ 0 GPL-3.0 0 4 0 Updated 2 weeks ago



**mpi4cpp** Public

User-friendly MPI headers for modern C++14 with template metaprogramming

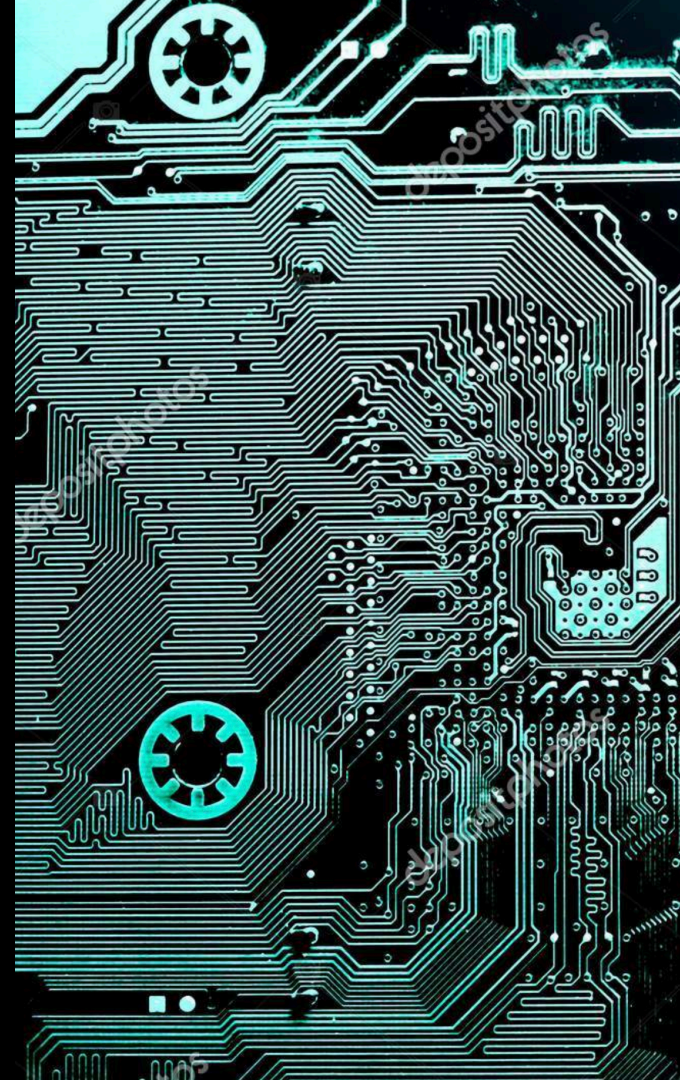
C++ 7 Apache-2.0 2 0 0 Updated 3 weeks ago



**SFQED** Public

Python package for calculating QED cross sections in a strong background magnetic field.

Jupyter Notebook 1 MIT 0 0 0 Updated on Mar 26

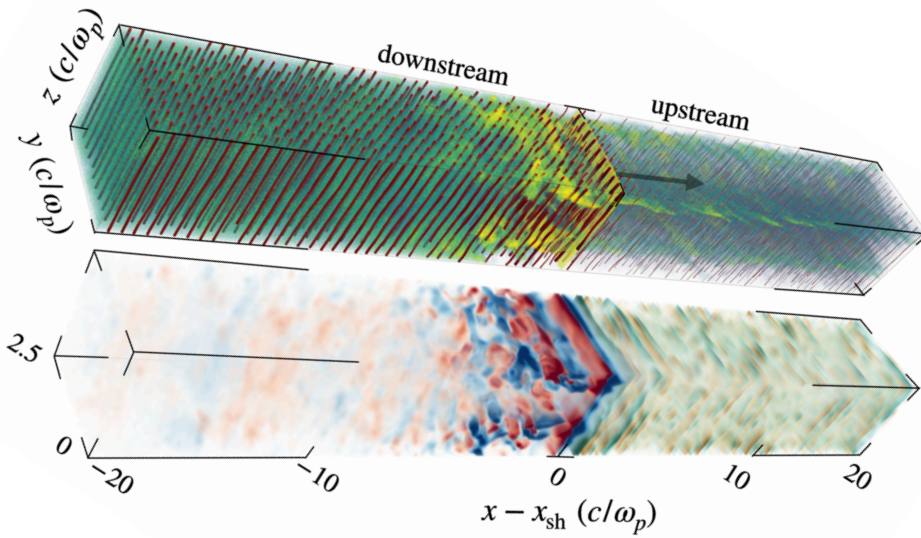


# runko

## Open-Source Simulation Toolkit

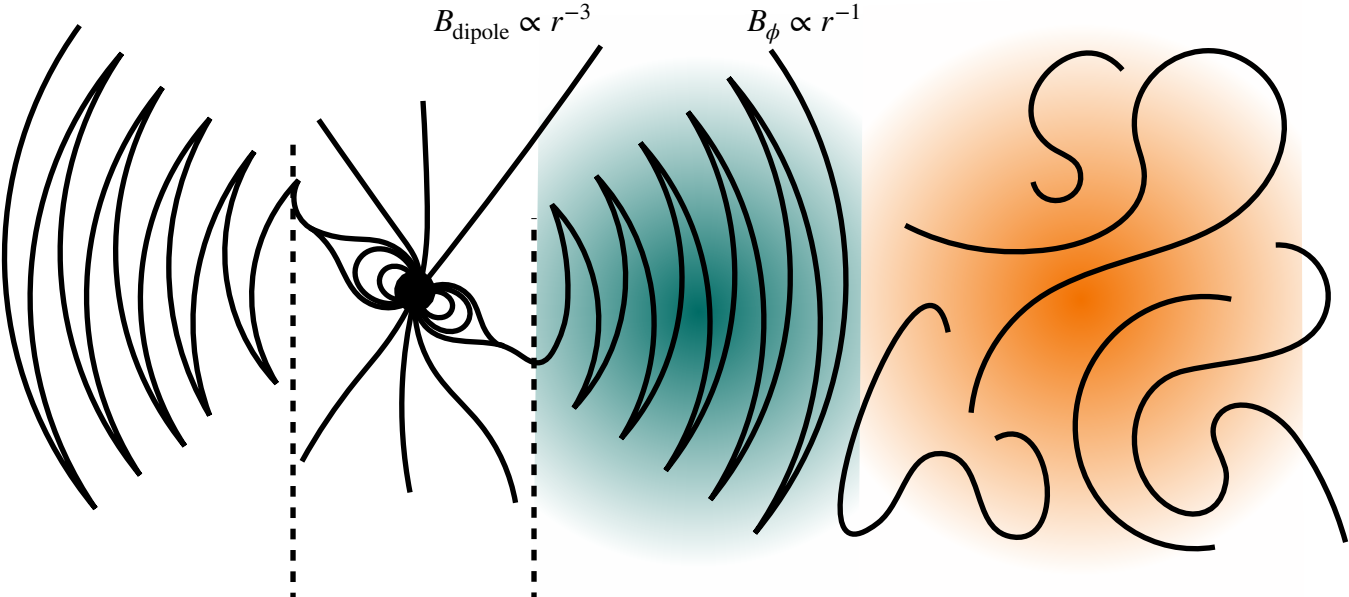
### Technological features:

- CPU/GPU portable 3D particle-in-cell code
- Modern C++26/Python3 high-performance code
- Multi-level parallelization w/ SIMD + MPI
- Open source (incl. problem setups, analysis scripts, etc.)



```
6 #ifndef GPU
7 #include <nvtx3/nvToolsExt.h>
8 #endif
9
10 using namespace sign;
11
12 template<size_t D, size_t V>
13 void pic::HigueraCaryPusher<D,V>::push_container(
14     pic::ParticleContainer<D>& con,
15     pic::Tile<D>& tile)
16 {
17
18     #ifndef GPU
19         nvtxRangePush(__PRETTY_FUNCTION__);
20     #endif
21
22     const double c = tile.cfl;
23     const double qm = sign(con.q)/con.m; // q_s/m_s (sign only bec
24
25
26     // loop over particles
27     UniIter::iterate( [= ] DEVCALLABLE (size_t n, pic::ParticleConta
28         double vel0n = con.vel(0,n);
29         double vel1n = con.vel(1,n);
30         double vel2n = con.vel(2,n);
31
32         // read particle-specific fields
33         double ex0 = ( con.ex(n) + this->get_ex_ext(0,0,0) ) * 0.5 * qm;
34         double ey0 = ( con.ey(n) + this->get_ey_ext(0,0,0) ) * 0.5 * qm;
35         double ez0 = ( con.ez(n) + this->get_ez_ext(0,0,0) ) * 0.5 * qm;
36
37         double bx0 = ( con.bx(n) + this->get_bx_ext(0,0,0) ) * 0.5 * qm;
38         double by0 = ( con.by(n) + this->get_by_ext(0,0,0) ) * 0.5 * qm;
39         double bz0 = ( con.bz(n) + this->get_bz_ext(0,0,0) ) * 0.5 * qm;
40
41         //-----
42         // first half electric acceleration
43         double u0 = c*vel0n + ex0;
44         double v0 = c*vel1n + ey0;
45         double w0 = c*vel2n + ez0;
46
47         //-----
48         // intermediate gamma
49         double g2 = (c*c + u0*u0 + v0*v0 + w0*w0)/(c*c);
50         double b2 = bx0*bx0 + by0*by0 + bz0*bz0;
51         double gtnv = 1./sqrt( 0.5*(g2-b2 + sqrt( (g2-b2)*(g2-b2) +
52
53         //-----
54         // first half magnetic rotation; ctnv is multiplied to B fie
55         bx0 *= gtnv/c;
56         by0 *= gtnv/c;
57         bz0 *= gtnv/c;
58
59         double f = 2.0/(1.0 + bx0*bx0 + by0*by0 + bz0*bz0);
60         double u1 = (u0 + v0*bz0 - w0*by0)*f;
61         double v1 = (v0 + w0*bx0 - u0*bz0)*f;
62         double w1 = (w0 + u0*by0 - v0*bx0)*f;
63
64         //-----
65         // second half of magnetic rotation & electric acceleration
66         u0 = u0 + v1*bz0 - w1*by0 + ex0;
67         v0 = v0 + w1*bx0 - u1*bz0 + ey0;
68         w0 = w0 + u1*by0 - v1*bx0 + ez0;
69
70         //-----
71         // normalized 4-velocity advance
72         con.vel(0,n) = u0/c;
```

# Magnetosphere



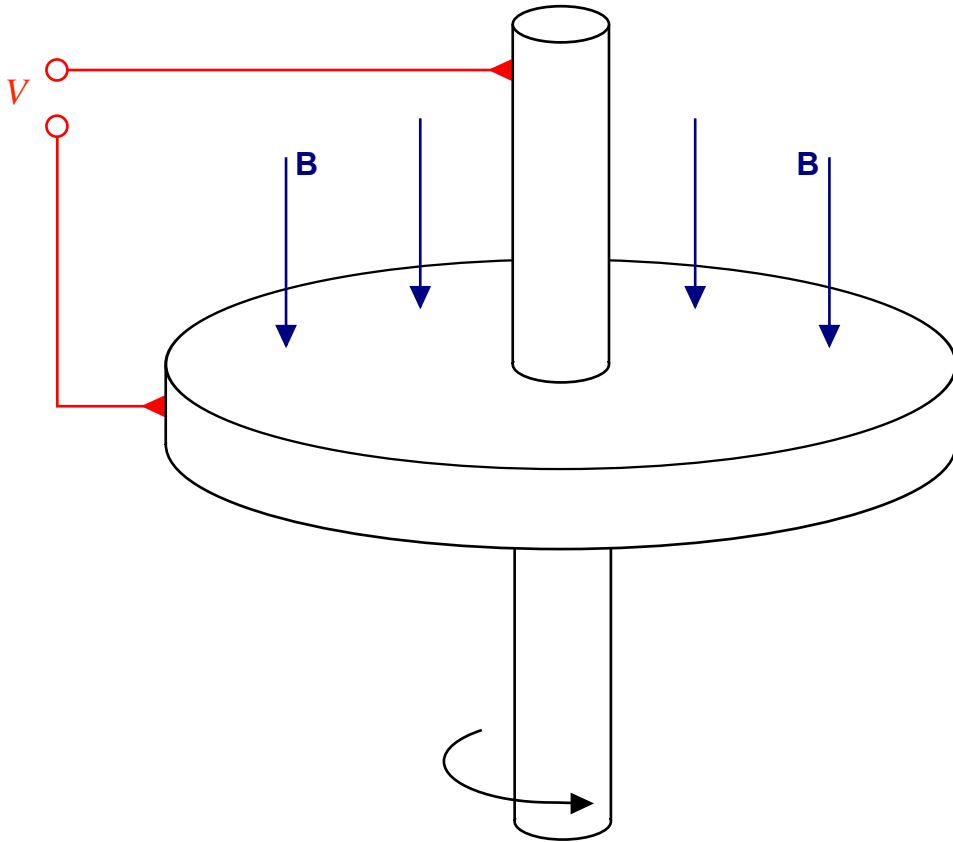
magnetosphere

wind

turbulent nebula

$$v_{\phi} = c$$
$$R_{\text{LC}} = c/\Omega \sim 1000 R_{\star}$$

# Faraday disk



## Electric circuits:

$$\text{Voltage } V \sim \Omega\Phi/c$$

$$\text{Power } P \sim V^2/R = IV$$

$$\text{Current } I \sim \Omega BR_{\text{disk}}^2/c$$

## Crab pulsar:

$$B \sim 10^{12} \text{ G}$$

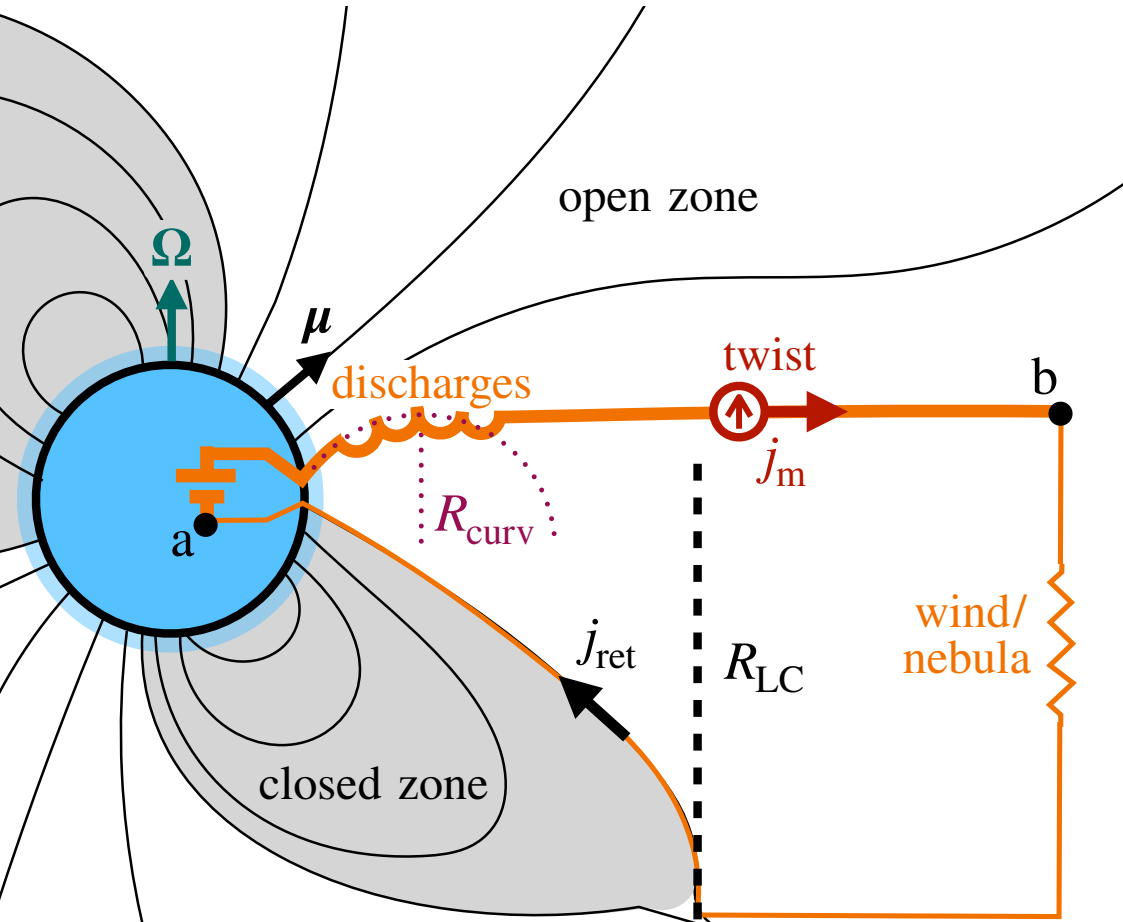
$$\Omega \sim 200 \text{ rad s}^{-1}$$

$$V \sim 10^{16} \text{ Volt}$$

$$I \sim 10^{14} \text{ Ampere}$$

$$P \sim 10^{38} \text{ erg s}^{-1}$$

# Magnetospheric circuit



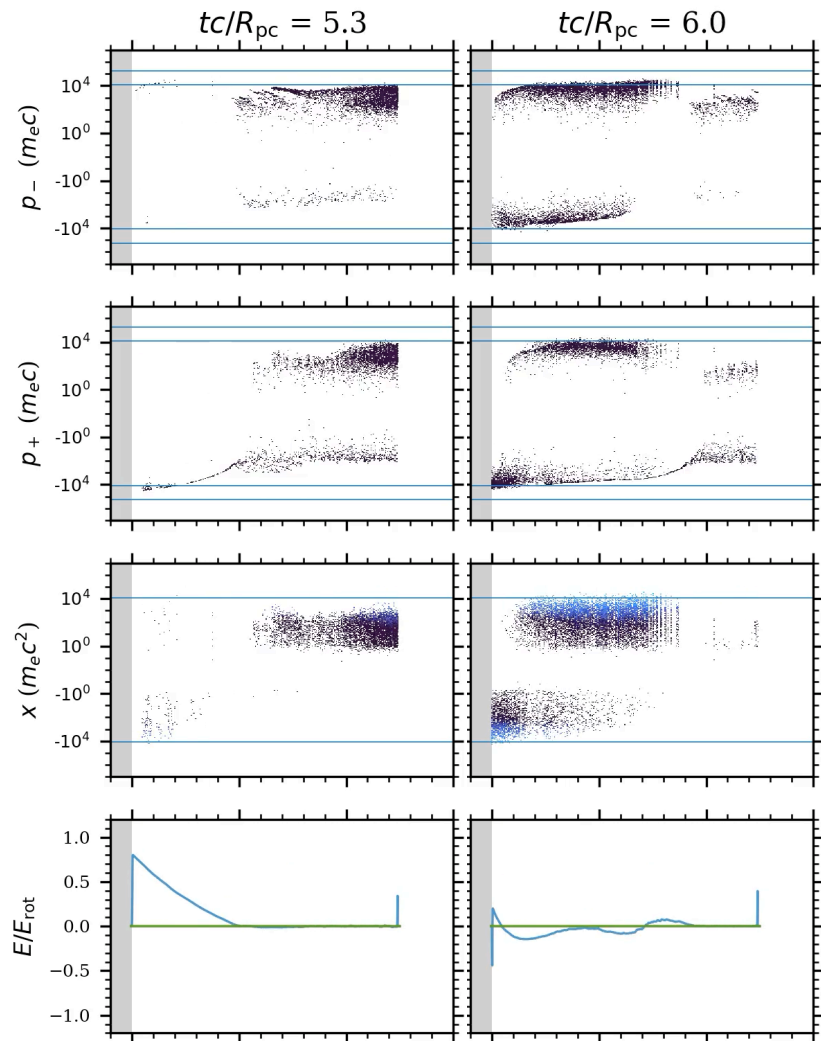
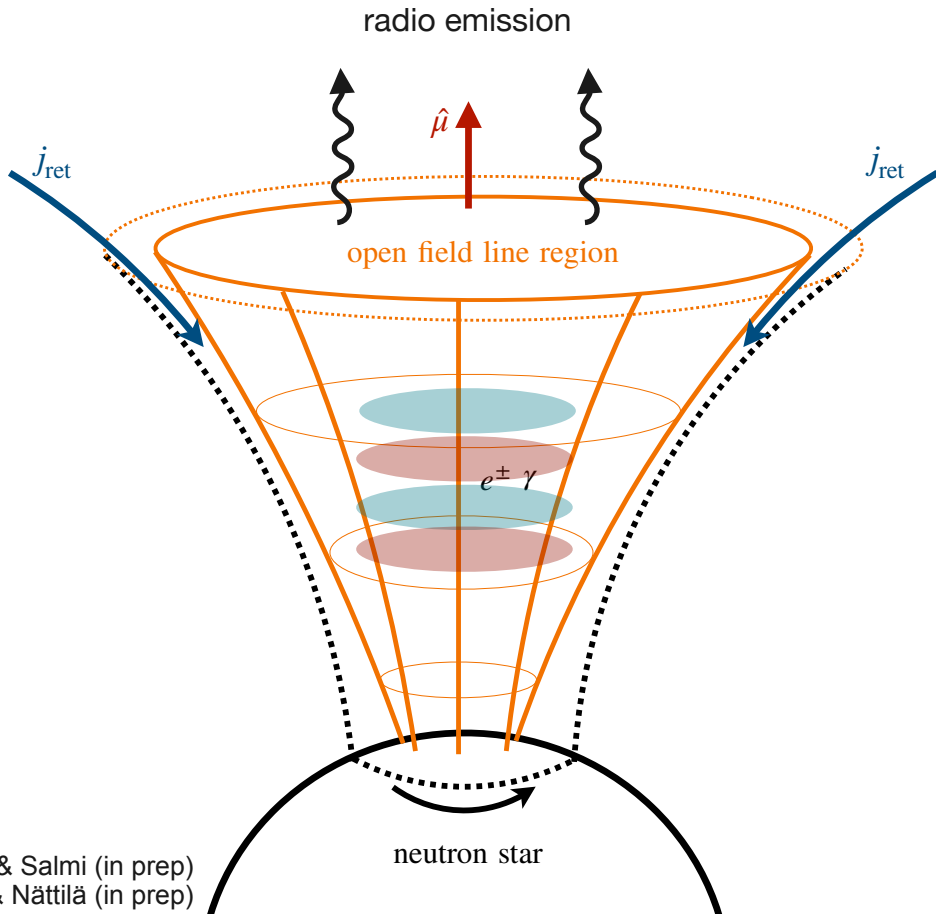
$$n_{\text{co}} = \frac{|\nabla \cdot \mathbf{E}|}{4\pi e} \sim \frac{\Omega_{\star} B_0}{2\pi c e}$$

$$\sim 10^{10} B_{12} P_0^{-1} \text{ cm}^{-3},$$

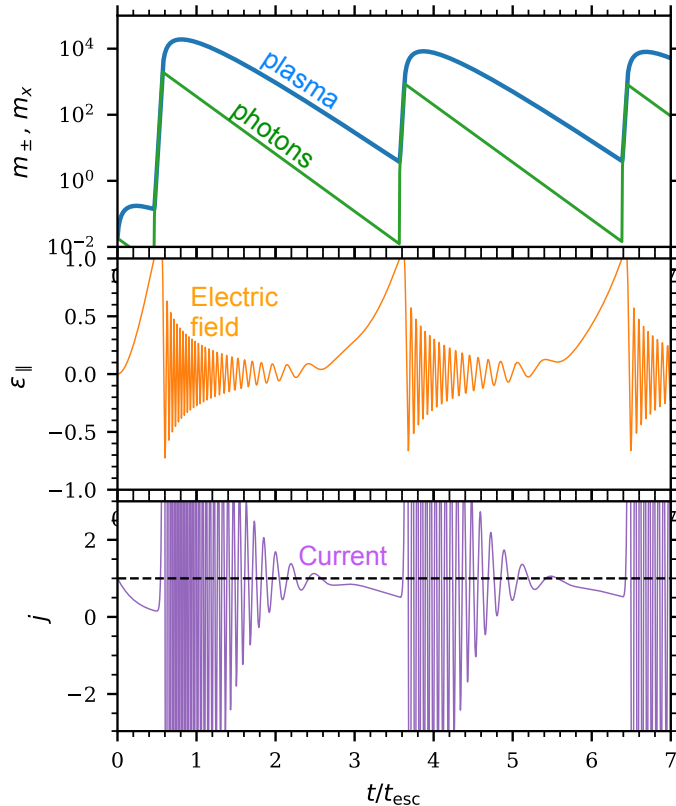
$$\gamma_{\text{gap}} = \int_0^H \frac{d\gamma}{dh} dh = \frac{e\Delta V}{m_e c^2}$$

$$\sim 10^7$$

# Polar cap discharges



# Pulsar discharge cycles



Simple ODE set for the plasma/radiation dynamics

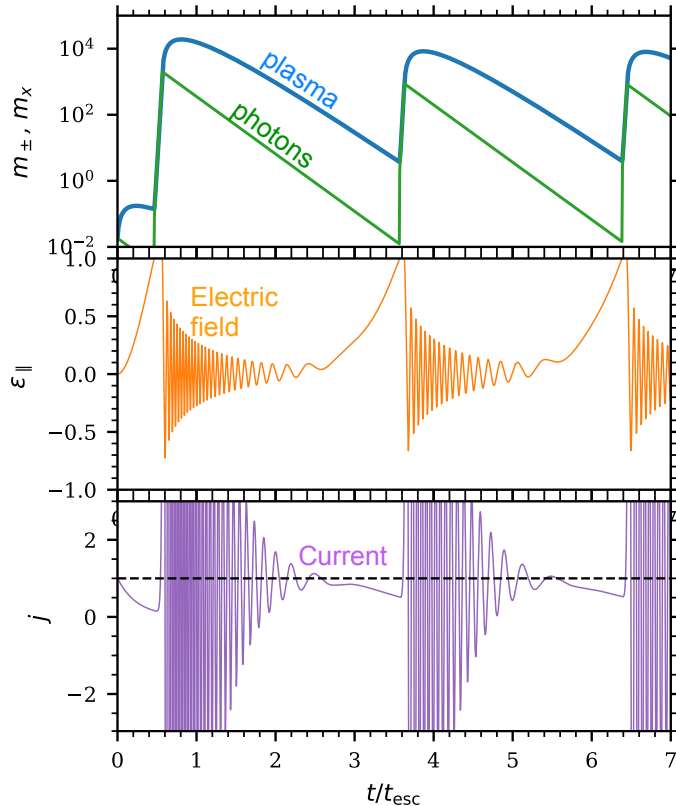
pairs 
$$\dot{m}_{\pm} = \frac{m_x}{t_{\pm}} - \frac{m_{\pm}}{t_{\text{esc}}}$$

photons 
$$\dot{m}_x = -\frac{m_x}{t_{\pm}} + \frac{m_{\pm}}{t_{\text{curv}}} f(\epsilon) - \frac{m_x}{t_{\text{esc}}}$$

electric field 
$$\dot{\epsilon} = \frac{j_m - j}{t_{\text{esc}}}$$

current 
$$\dot{j} = -\frac{j}{t_{\text{esc}}} + \frac{m_{\pm}\epsilon}{t_{\text{acc}}} \left\langle \frac{1}{\gamma^3} \right\rangle$$

# Pulsar discharge cycles



Simple ODE set for the plasma/radiation dynamics

pairs  $\dot{m}_{\pm} = \frac{m_x}{t_{\pm}} - \frac{m_{\pm}}{t_{\text{esc}}}$

photons  $\dot{m}_x = -\frac{m_x}{t_{\pm}} + \frac{m_{\pm}}{t_{\text{curv}}} f(\epsilon) - \frac{m_x}{t_{\text{esc}}}$

electric field  $\dot{\epsilon} = \frac{j_m - j}{t_{\text{esc}}}$

current  $\dot{j} = -\frac{j}{t_{\text{esc}}} + \frac{m_{\pm} \epsilon}{t_{\text{acc}}} \left\langle \frac{1}{\gamma^3} \right\rangle$

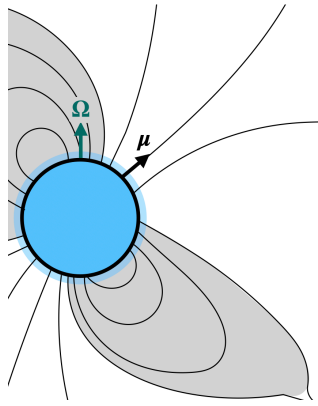
$$M_{\pm} \sim \frac{10 t_{\text{acc}}}{t_{\pm} \langle \gamma^{-3} \rangle}$$

$$t_{\text{cycle}} \sim \ln(M_{\pm}) t_{\text{esc}}$$

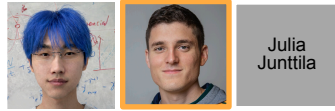
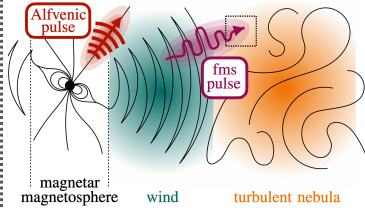
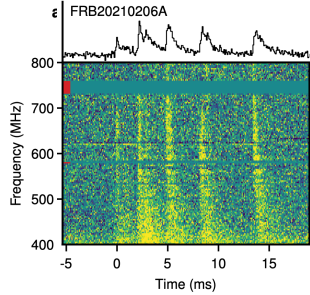
$$\omega_{\text{osc}} \sim \sqrt{\frac{M_{\pm} \langle \gamma^{-3} \rangle}{t_{\text{acc}} t_{\text{esc}}}}$$

# Research areas of plasma astrophysics research group

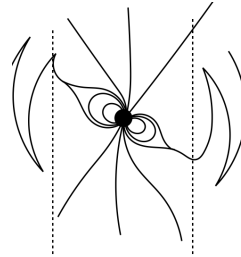
## Pulsars



## Fast radio bursts

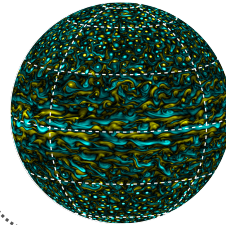


## Magnetars & QED



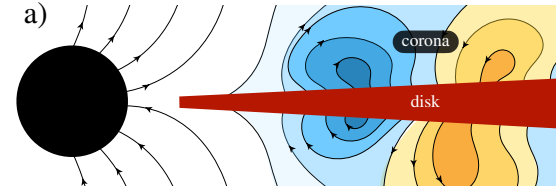
Veneri Sarjomaa

## Atmosphere dynamics



Emil Issakainen

## Turbulent accretion flows



## Equation of state

