# ArcPIC 2D: Simulating arc ignition

Kyrre Sjobak[1], Lotta Mether[2], Helga Timkó, Walter Wuench

University of Oslo / University of Helsinki / CERN

MeVARC 2012, Albuerquerque, 1.-4. October

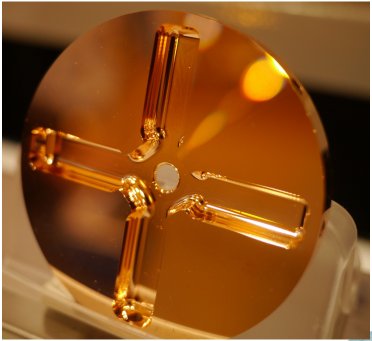
[1] k.n.sjobak (a-with-pigtail) fys.uio.no
[2] lotta.mether (a-with-pigtail) helsinki.fi

# Outline

# Motivation

## Motivation

- CLIC: High gradient ($E_{\mathrm{acc}} \approx 100\mathrm{MV/m}$), normal conducting $\mathrm{e}^{+}\mathrm{e}^{-}$ collider
- Arcs limit gradient of normal conducting RF accelerators
- Predict performance of accelerating structure geometry during RF design
- Understanding initiation of vacuum arcs
- Better description of involved physics needed

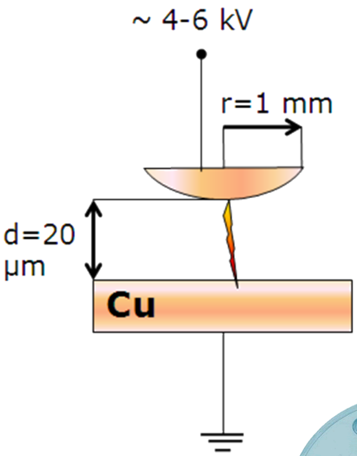


# DC spark experiment

The DC spark experiment at CERN measures:

- Voltage and current flow through the breakdown
- High repetition rate capability (separate geometry)

In-depth presentation by Nick Shipman tomorrow

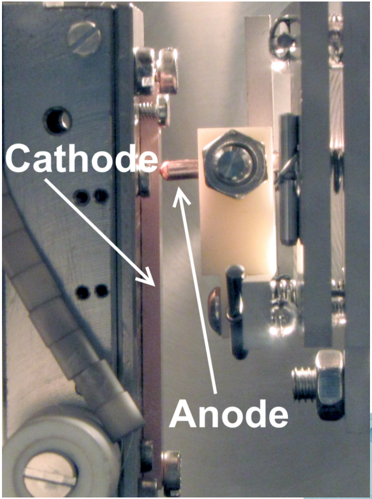



# DC spark experiment

The DC spark experiment at CERN measures:

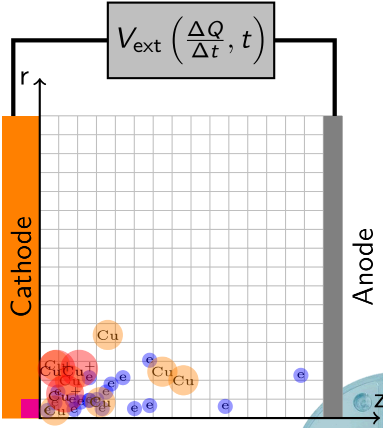
- Voltage and current flow through the breakdown
- High repetition rate capability (separate geometry)

In-depth presentation by Nick Shipman tomorrow



# ArcPIC 2D in a nutshell

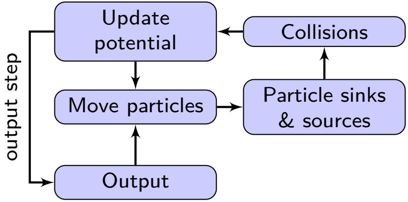
- 2d3v electrostatic particle-in-cell (PIC) code
  - cylindrical symmetry
  - uniform finite difference grid
- Geometry: Planar electrodes
- Particles: $\mathrm{e}^-$, $\mathrm{Cu}^+$, $\mathrm{Cu}$
- Goal: Testing physics models for breakdown early stages
- Physics (***modular part***):
  - ***External circuit***
  - ***Particle injection***
  - Collisions (el./inel.)
  - Electrostatic interaction
  - External magnetic field
- Language: C++, partly OO




Originally developed by Helga Timkó

# ArcPIC 2D in a nutshell

- 2d3v electrostatic particle-in-cell (PIC) code
  - cylindrical symmetry
  - uniform finite difference grid
- Geometry: Planar electrodes
- Particles: $\mathrm{e}^{-}$, $\mathrm{Cu}^{+}$, $\mathrm{Cu}$
- Goal: Testing physics models for breakdown early stages
- Physics (***modular part***):
  - ***External circuit***
  - ***Particle injection***
  - Collisions (el./inel.)
  - Electrostatic interaction
  - External magnetic field
- Language: C++, partly OO




# Code development

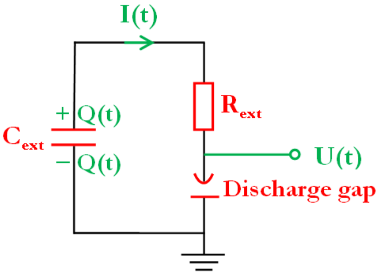
# Recent code improvements

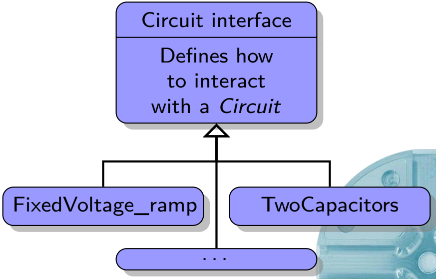
- More modern techniques (OO, STL) introduced
- Many “infrastructure” pieces rewritten
- Hot parts found using profiling
- Gains in modularity, maintainability, and performance
  - Memory footprint reduction: $\approx$90 GB $\rightarrow$ $\approx$2 GB
  - Modular/OO circuit and particle injection

$\downarrow$

# Modular circuit models

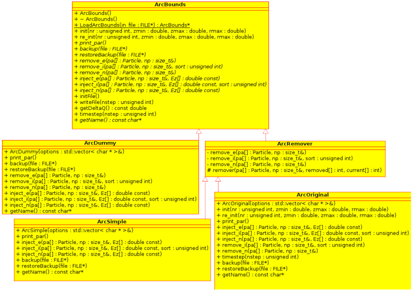
- Goals:
  - Mirror experimental setup(s)
  - Play with different ways of supplying external field
- Object-oriented system, classes representing different circuits
  - Charge stored on capacitor
  - Fixed voltage w/ linear turn-on
- Adding new models simple
- A model *may* provide extra instrumentation



# Particle boundaries

- Goals
  - Test multiple physics models
  - Reproduce experiment
  - Transfer models to RF
- Object-oriented system
  - Classes representing models
  - Inheritance enables partial re-use of models
  - Adding more models simple
- Each model responsible for injecting and removing particles on domain boundaries



# Near-future outlook (code)

## Handling of many particles

**Want:**

- Good MC resolution in BD initial stages
- Possibility to run far into breakdown

**Methods:**

- Parallelization (OpenMP)
- Dynamic particle weighting, merging

## Field resolution

**Want:**

- Properly resolve the boundary layer

**Methods**

- Non-uniform grid

## Non-symmetric breakdowns

**Want:** Spontaneous breaking of circular symmetry of emission spot

**Method:** Cylindrical 3D field solver & pusher

# Summary (code)

- We have a fairly robust PIC code
- Modular particle boundary conditions and circuit model
- Modeling DC spark experiment @ CERN

**Goal**

Find a combination of physics models describing arc ignition

# Physics models

# Conclusions