Kinetic simulation of runaway electrons in tokamaks

Hungarian Plasma Physics and Fusion Technology Workshop

G. Csépány[1*], G. I. Pokol[1], G. Papp[1,2], T. Fülöp[2], J. Rydén[2]
Former authors: L.-G. Eriksson[3], P. Helander[4]

**email: csepany.gergely@reak.bme.hu*

[1] *BME-NTI, Association EURATOM-HAS, Budapest, Hungary*

[2] *Chalmers University, Association EURATOM-VR, Göteborg, Sweden*

[3] *European Commission, Brussels, Belgium*

[4] *IPP-Greifswald, EURATOM Association, Greifswald, Germany*

The publication of the work reported in the poster has been supported by ETDB at BME, and it has been developed in the framework of the project „Talent care and cultivation in the scientific workshops of BME" project. This project is supported by the grant TÁMOP-4.2.2/B-10/1-2010-0009.

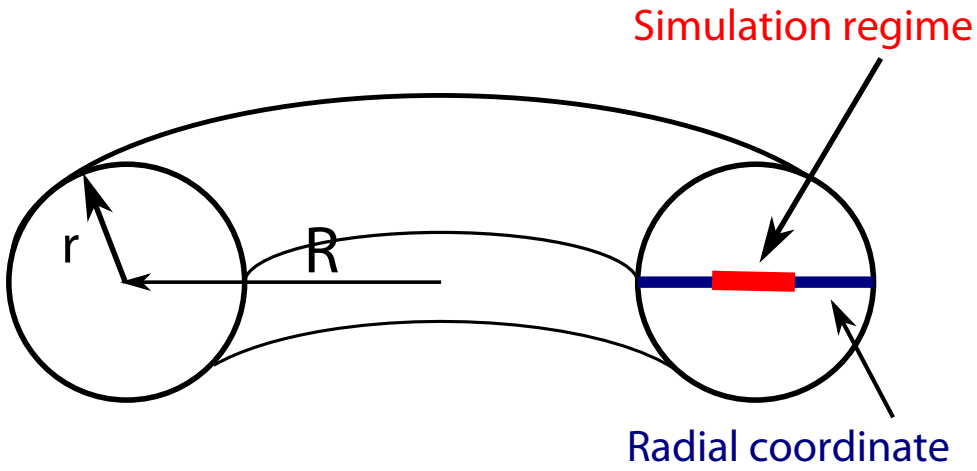
MŰEGYETEM 1782

CHALMERS

ARENA

Analysis of Runaway Electrons by Numerical Algorithms

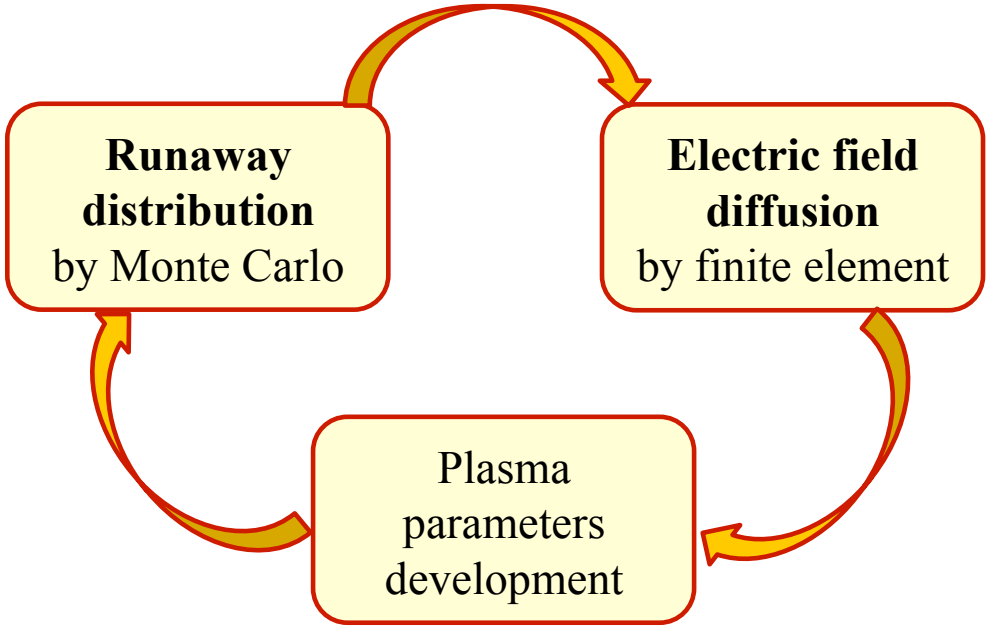
- Runaway electrons are dangerous for large tokamaks $\Rightarrow$ numerical simulation is essential
- Written in 1998, developed until 2004[1,2]
- Solves the bounce-averaged kinetic Fokker Planck equation by Monte Carlo methods in 1+2D phase space



[1] L.G. Eriksson and P. Helander, Computer Physics Communications, 154 (2003)(3), 175–196.
[2] L.G. Eriksson et al., Physical Review Letters, 92 (2004)(20), 1–4.

ARENA Basics

- **Runaway electron growth rate** calculation (primary and secondary)
- **Self-consistent** calculation of the electric field



- **Promising results** for ITER and JET parameters[2]
  - Was too slow for thorought simulations

Motivation

- **Bring ARENA back to life**
  - „It had worked" once
  - No documentation
  - No released versions or source code versioning
- **Speed up**
  - The fusion group would warmly welcome a released version
  - Use for more realistic scenarios
  - Opens the possibility to include new features
- **Extend it**
  - Introduce new physical effects
  - Implement a collision operator valid for arbitrary energies [*Papp et al., NF 2011*]
  - Modularize the disruption simulation

Code works

- **In the beginning** we had
  - Fortran 77 code (development version)
  - Two articles (as „documentation”)
- **Finished** works
  - Conversion to Fortran 90
  - Decrypt most of the 6-letter variable names
  - Write developer documentation
- **Work in progress**
  - Testing and benchmarking
  - User documentation
  - ITM integration

```
TEDOTN =-1.0 / TTEDEC * (ZTEMP0 - TEMIN) *
        EXP(-TIME(IT) / TTEDEC)

TEDOTNB =-2.0 / TTEDEC * (ZTEMPEB0 - TEMIN) *
        EXP(-2.0*TIME(IT) / TTEDEC)

PVTRAT  = SQRT(TEMPE0 * PVTRAT)

ED0 = ZED0 * ZTEMP0 / TEMPE0
```

Numerical tests

Why do we need a numerical test?

- To see that we didn't brake the code while working on it

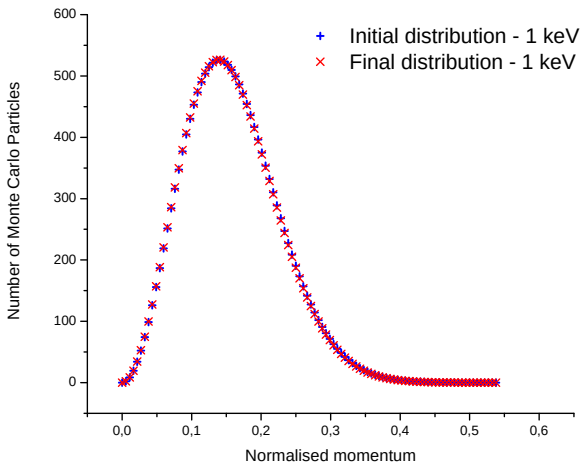
How can a numerical simulation tested?

- Ideal case
  - All functions and routines are documented and each tested automatically by unit tests
- ARENA case
  - Basically one huge routine
  - Tailored to do a disruption simulation
  - Not much individual modules and „old" Fortran 77
  - Make it unit testable means almost a total rewrite
- Solution
  - Set up a test disruption run which uses most of the code
  - Run the original and every sequential version with these parameters

Physical tests

Make sure the code calculates what we want

- **Maxwellian-test**: the collision operator should keep the Maxwellian without external forces
  - Status: Done and OK, up to 15 keV



- **Dreicer-test**: growth rate of the primary generation
  - Status: In progress, qualitatively OK
- **Avalanche-test**: growth rate of secondary generation
  - Status: Planned

What is ITM?

Integrated Tokamak Modelling

Aim: create a tokamak simulator based on European codes

Problem: Simulations use different data structures, physical units and programming languages

Solution: Standardized units, data structure and common programming platform.

- Units: SI, conventions for coordinate systems, signs, currents, betas, etc.
- Data: Consistent Physical Object[3]: hierarchical structure (CPO)
- Platform: ITM Gateway[4], a joint computing facility

[3]F. Imbeaux et al., Computer Physics Communications, Volume 181, Issue 6, June 2010, Pages 987-998
[4] `http://www2.efda-itm.eu/`

Infrastructure

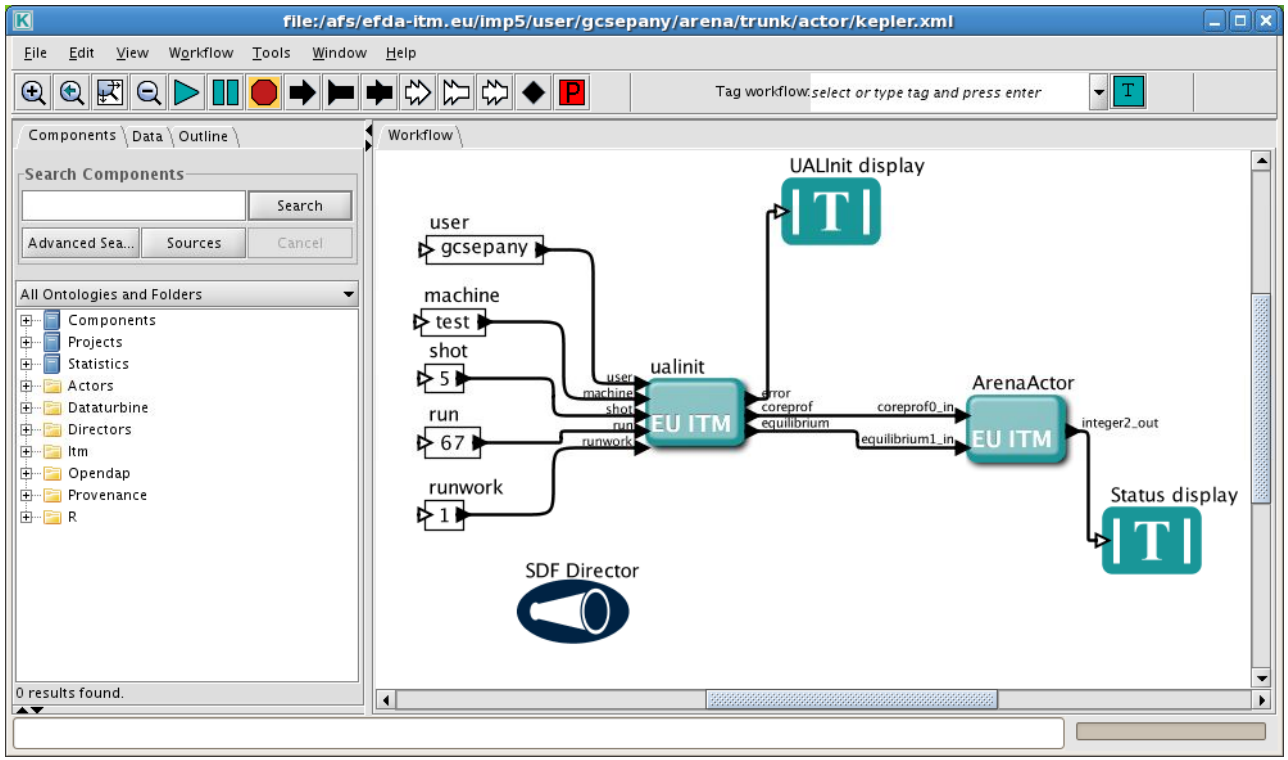
ITM gives a significant support to acheive its aim

- **Gateway**: Multicomputer environment with all necessary softwares
  - Official support to resolve problems
- **GForge**: Collaborative Development Environment
  - Task and bug tracker
  - Mailing list
  - Wiki
  - SVN
- **Doxygen**: Standardized documentation system

ITM workflows

Codes are connected in Kepler[5] workflows

- Java based, easy-to-use program
- Codes can be compiled in Kepler (as shared objects)
- Kepler runs the programs, connects them and outputs results



[5] https://kepler-project.org/

LUKE and ARENA

LUKE: fast numerical solver for the 3D relativistic bounce-averaged electron Drift Kinetic Equation[6]

Benchmark

- Both LUKE and ARENA can calculate growth rate from the primary generation
- They use different approach to calculate it (MC vs FE)

Connecting

- LUKE is much faster than ARENA for primary generation
- LUKE cannot calculate secondary generation
- Long term solution: connect them in ITM and serve as the basic disruption simulation

[6] Y. Peysson et al., Aip Conference Proceedings 1069, 176-187 (2008).