

## Towards self-consistent runaway electron modeling

G. Papp<sup>1,2</sup>, A. Stahl<sup>3</sup>, M. Drevlak<sup>1</sup>, T. Fülöp<sup>3</sup>, Ph. W. Lauber<sup>2</sup> and G. I. Pokol<sup>4</sup>

<sup>1</sup>*Max Planck Institute for Plasma Physics, Garching/Greifswald, Germany.*

<sup>2</sup>*Max-Planck/Princeton Center for Plasma Physics.* <sup>3</sup>*Dept. of Applied Physics, Chalmers University of Technology, Gothenburg, Sweden.* <sup>4</sup>*Institute of Nuclear Techniques, Budapest University of Technology and Economics, Budapest, Hungary.*

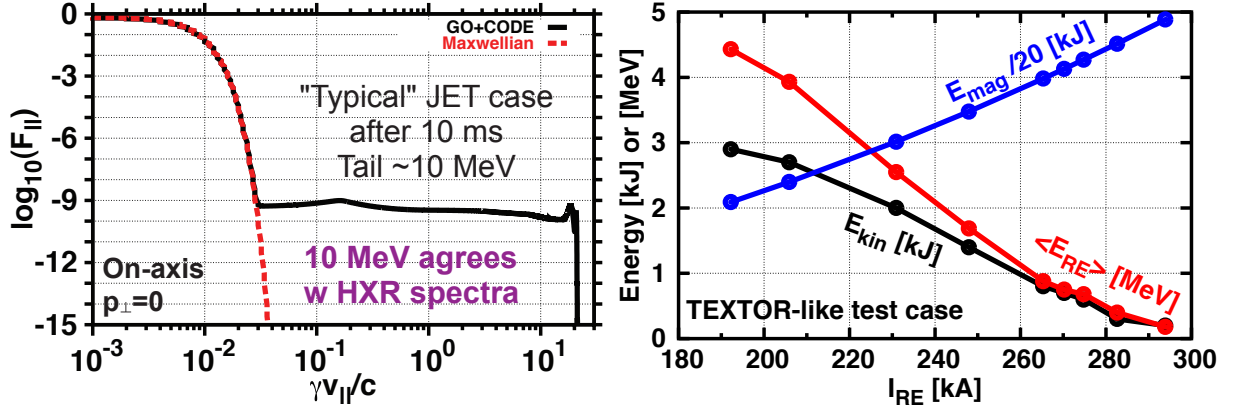
**Introduction** Reliable runaway electron (RE) mitigation after disruptions is one of the most important challenges for safe ITER operation [1]. A proper understanding of the generation and losses of REs is therefore essential. Recent experimental [2, 3] and theoretical [4] results suggest that knowledge of the electron distribution function is essential to properly describe the evolution of post-disruption runaway beams. One way to model the electron distribution function is by solving the Fokker-Planck equation [5, 6]. However, in order to properly account for the time variation of plasma parameters, a self-consistent evolution of the background plasma (temperature and density profiles, impurity composition etc) is necessary. This paper reports on the first results of our self-consistent combined fluid-type and Fokker-Planck simulations of post-disruption runaway evolution.

**Numerics** Here we consider two Fokker-Planck solvers. **CODE** [5] (Collisional Distribution of Electrons) is a fast 2-V continuum (Eulerian) fully implicit Fokker-Planck solver with arbitrary electric field and collisions, using a relativistic momentum-conserving collision operator valid for arbitrary energies. It also takes the effect of synchrotron radiation into account [4], which can significantly change the runaway growth rate under certain plasma conditions. In the future we plan on using an even more sophisticated tool, LUKE [6], which, among other things, includes toroidicity effects, electron heating & current drive, etc. In this paper, only results using CODE are discussed.

To follow the evolution of the plasma background we use the **GO** code [7]. GO calculates runaway electron generation in disruptions using a self-consistent fluid model of the electric field and other important plasma parameters. It includes an ADAS-based, temperature and density dependent collisional-radiative atomic physics model for an arbitrary ion composition. In the future, GO could be replaced by e.g. a self-consistent 3D MHD (such as JOREK or NIMROD) tool calculating the whole disruption phase including impurities. However, the relative simplicity and orders of magnitude smaller computational requirements of GO will always be an advantage.

Schematically the coupling is relatively simple: a separate distribution function is maintained for each radial point in GO, and in each GO time step, CODE is invoked to evolve the distribution. CODE determines the necessary number of sub-iterations to evolve the distribution, based on the plasma parameters. In order to save CPU time, each CODE call has an independent setting of numerical parameters (energy grid, number of Legendre polynomials, number of iterations, etc). CODE utilizes dynamic grids, where the grid is continuously adapting based on the distribution function and the plasma parameters. This gives roughly a factor of  $\sim 100\times$  (or more) speedup compared to a fixed grid approach. Load balancing is of particular importance: each (radial) CODE run has to finish before the global plasma parameters can be evolved and the amount of time it takes

to evolve the distribution can differ by orders of magnitudes for the different radial grid points. Paralellisation is utilized in an “MPI+openMP”-style approach. Each CODE call is a separate worker while the optimum number of threads is determined by the size of the sparse CODE matrix and the number of necessary sub-iterations. We observe good sparse scaling as a function of grid sizes and number of iterations. We believe a factor of 1.5-2x speedup could be achieved by further optimizing the paralellisation scheme, however it is uncertain if the potential improvement justifies the necessary time and effort. In general, the bottleneck is memory bandwidth, which might be improved by utilizing high memory bandwidth Intel Xeon Phi accelerator cards.



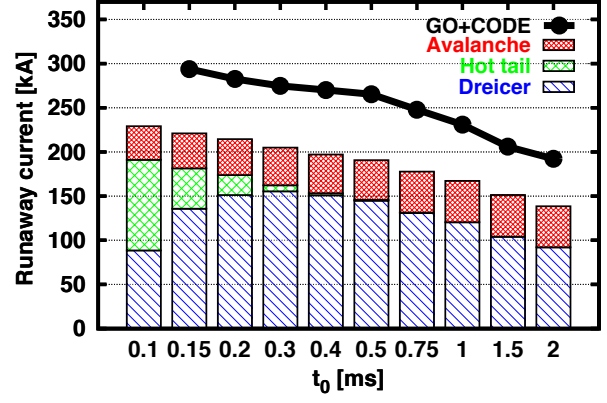
**Figure 1:** (a) Typical spectrum evolution from GO+CODE for JET parameter range. The maximum runaway energy agrees well with the energy usually inferred from HXR spectral reconstruction. (b) Magnetic energy, total kinetic energy and mean runaway energy as a function of post-CQ runaway current. In higher RE current scenarios the electric field decays faster and consequently the kinetic energy carried by the runaways is lower.

**First results** The initial test simulations were conducted with an “as simple as possible” approach. We use a TEXTOR-like plasma with  $T_e = 1.3$  keV predischruption central electron temperature,  $n_e = 1.7 \cdot 10^{19} \text{ m}^{-3}$  central electron density,  $B_0 = 2.1$  T,  $R = 1.8$  m,  $I_0 = 305$  kA. The atomic physics module in GO is inactive, the thermal quench is instead initiated by a forced exponential collapse of the electron temperature. In the figures presented in this paper the post-thermal-quench central electron temperature was 5 eV.

Figure 1b shows the magnetic energy, total kinetic energy and mean runaway energy as a function of post-CQ runaway current (scanned by changing the thermal quench time). We note that in scenarios that produce a larger post-thermal-quench runaway current, the electric field decays faster (as it is being “consumed” by the REs) and consequently the kinetic energy carried by the runaways is lower. This also means that mitigation of already formed runaway beams by high-Z material injection might be more effective in higher runaway current scenarios.

If we compare the saturation runaway currents between GO+CODE and GO we observe a 35% ( $\sim 60$  kA for these parameters) extra production of runaways, as is illustrated in figure 2. Deciding which model is closer to reality is a nontrivial task. In general, CODE shows a higher initial burst of runaways in the early stages of evolution compared to the analytical primary generation formulas [8, 9].

However, it should be noted that the Dreicer generation formula was derived for quasi-steady-state distributions. We will clarify the picture by comparing to other Fokker-Planck solvers [6]. In conclusion we can say that the coupling between GO and CODE *works*, and the first results are encouraging. However, there is still further optimization, testing and validation to experiments [3] to be done.

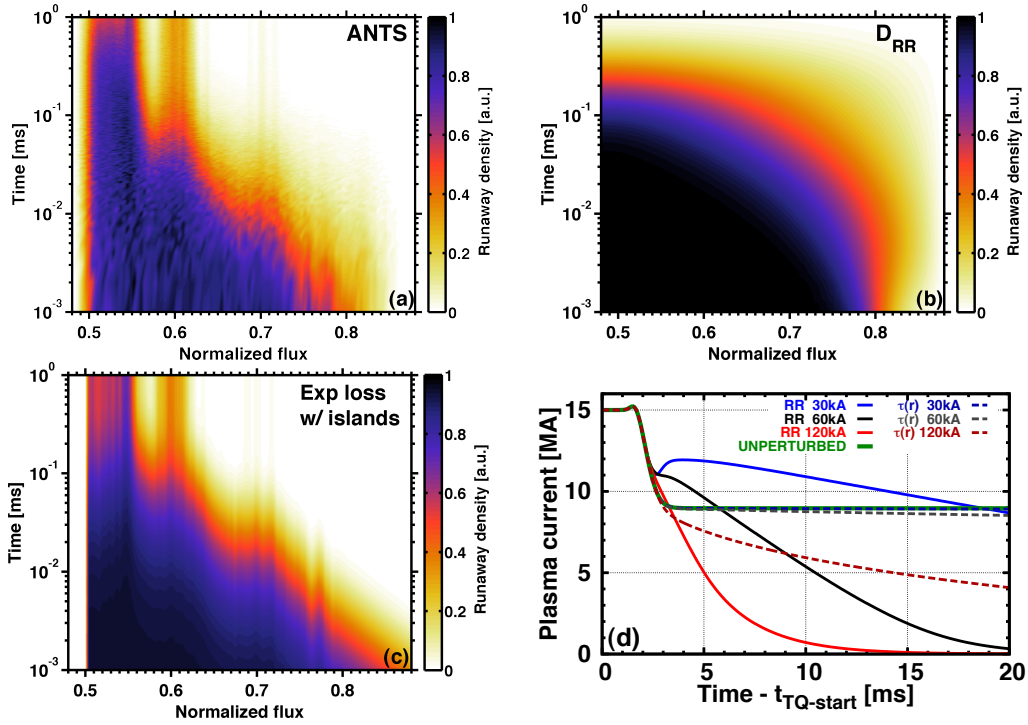


**Figure 2:** Comparison of GO+CODE and GO:  $\sim 35\%$  difference in runaway current is observed.

### The effect of Resonant Magnetic Perturbations

In the past few years we have analysed in detail the effect of Resonant Magnetic Perturbations (RMP) on runaway electron losses in ITER (see [10] and references therein). The most important conclusion was that runaways closer to the edge ( $r/a = 0.7$ ) can be quickly lost under the influence of RMPs, however, core particles are expected to be well confined. However, earlier computations were only using a test particle approach [10] and therefore we could not take into account the effect of RMPs on runaway generation with a self-consistently calculated electric field evolution. Figure 3a illustrates the runaway losses under the effect of RMPs. A flat runaway profile of 1 MeV particles is initialized and then followed up to 100 ms. Figure 3a shows the evolution of the runaway density profile as a function of time and normalized flux. Close to the edge particles can be lost as soon as after  $1\mu\text{s}$ , while particles in the central region are well confined.

The usual 1 dimensional description of transport caused by magnetic perturbations is the Rechester-Rosenbluth diffusion [11], which scales with  $(\delta B/B)^2$ . If we now introduce RR diffusion into GO [7] with the  $\delta B(r)$  profile calculated for the ITER ELM perturbation coils [10], we arrive to the profile evolution presented in figure 3b. If we compare figures 3a-b (mind the logarithmic time scale on the y-axis) we immediately see that the RR diffusion does not reproduce the test particle results obtained with ANTS. We also found that not only RR diffusion is not applicable, but the transport itself is not diffusive. Diffusive transport would imply a flux proportional to the local gradient, but we found that the flux is proportional to the local density, which is an indication of a convective type loss [12]. Even though the individual particle orbits are chaotic, the ensemble behavior is smooth (figure 3a), therefore we approximated the particle losses with an  $N_{\text{lost}}(t) \propto 1 - \exp\{-t/\tau(\psi)\}$  trend, where  $\tau(\psi)$  is the characteristic loss time associated with a certain initial  $\psi$  position for the particles. We have performed fits of  $\tau(\psi)$  to the evolution of fast electron density obtained by the 3D test particle simulations for various cases. We note that the dependence of  $\tau(\psi)$  on the radial coordinate is relatively smooth and is well approximated by an exponential dependence  $\tau(\psi) \propto \exp\{-\psi/\psi_0\}$ , where  $\psi_0 \propto \delta B/B$ . Figure 3c shows the density profile evolution based on the exponential losses using the fitted  $\tau(\psi)$  values. It clearly resembles the particle evolution calculated by ANTS.



**Figure 3:** (a) Evolution of a flat runaway density profile in 3D ANTS simulations. (b) Evolution of the density profile under RR diffusion and (c) with the exponential loss model. (d) The RR diffusion suggests that RMPs can suppress runaway formation with ITER, it is not the case with the loss model based on the ANTS results [10, 12].

To illustrate the differences between the implications of the diffusive and exponential-loss models, we have also implemented the latter in GO. Figure 3d shows a disruption simulation for the same ITER scenario that was used for the 3D simulations. Without losses a runaway beam of  $\sim 9$  MA would form. Using the RR diffusion model we get suppression of the runaways after 20 ms at  $I_{\text{RMP}} = 60$  kA and after 10 ms if  $I_{\text{RMP}} = 120$  kA. As the dashed lines clearly illustrate, **virtually no runaway suppression is achieved** with the exponential loss model. Although the perturbation penetrates the core, the test particle simulations show that the transport is essentially untouched inside the radius  $r/a = 0.7$  for  $I_{\text{RMP}} = 60$  kA. Therefore, even if the transport is largely increased towards the edge region in the exponential loss description, this does not contribute much to runaway suppression. The reason is that most of the runaways are formed in the plasma core, where there is good runaway confinement according to the test particle simulations.

**Acknowledgment** This work was funded by the Max-Planck/Princeton Center for Plasma Physics. This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014-2018 under grant agreement No 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

## References

- [1] E. M. Hollmann *et al.*, PoP **22** 021802 (2015)
- [2] G. Pautasso *et al.*, EPS 2015 P1.134
- [3] E. M. Hollmann *et al.*, PoP, **22** 056108 (2015)
- [4] A. Stahl *et al.*, PRL **114** 115002 (2015)
- [5] M. Landreman *et al.*, CPC **185** 847 (2014)
- [6] J. Decker *et al.*, PSFC/RR-05-3 (2005)
- [7] G. Papp *et al.*, NF **53** 123017 (2013)
- [8] J. Connor and J. Hastie, NF **15** 415 (1975)
- [9] H. M. Smith *et al.*, PoP **15** 072502 (2008)
- [10] G. Papp *et al.*, PPCF **54** 125008 (2012)
- [11] A. B. Rechester, *et al.*, PRL **40** 38 (1978)
- [12] G. Papp *et al.*, JPP (a.), arXiv:1504.05074