Simulation of gas-filled diodes with the VORPAL code

Peter Stoltz, Chet Nieter

Tech-X Corp.

***Work in collaboration with J.-L. Cambier, AFRL/PRSA, funded by the OSD SBIR program.**

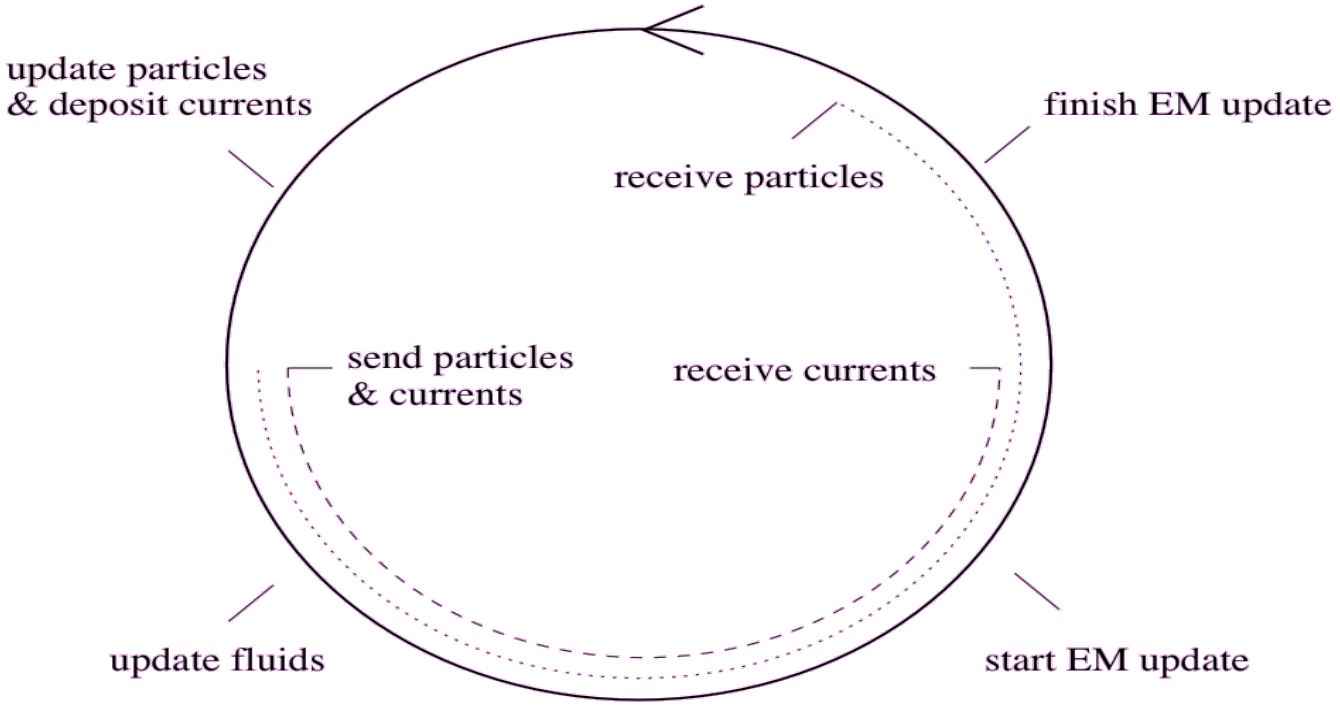
Abstract

Researchers are interested in rapid switching high-current diodes for applications including accelerators, microwave sources, and flat panel displays. Gas-filled diodes, such as the pseudospark, are one class of candidates for such high-current applications. To yield a better understanding of their operation, numerical simulations of these devices must include a number of physical processes: gas ionization, secondary electron emission and ion-induced electron emission, and radiation. We are enhancing the VORPAL code to model these physical effects, and we present here electromagnetic particle-in-cell (PIC) simulation results for a generic one-dimensional diode. These simulations include models of some of the processes mentioned above, as well as the effects of the self-consistent electro-magnetic fields of the particle beams. We discuss the numerical models used, including techniques to allow users to use these models from a variety of languages, including Java and C++. We examine the case of the planar diode for a variety of conditions and discuss the several physical regimes obtained in terms of analytical models. This study provides an excellent test-bed for verification and validation of both numerical and physical models.

VORPAL software architecture allows extensions for studying gas-filled diodes

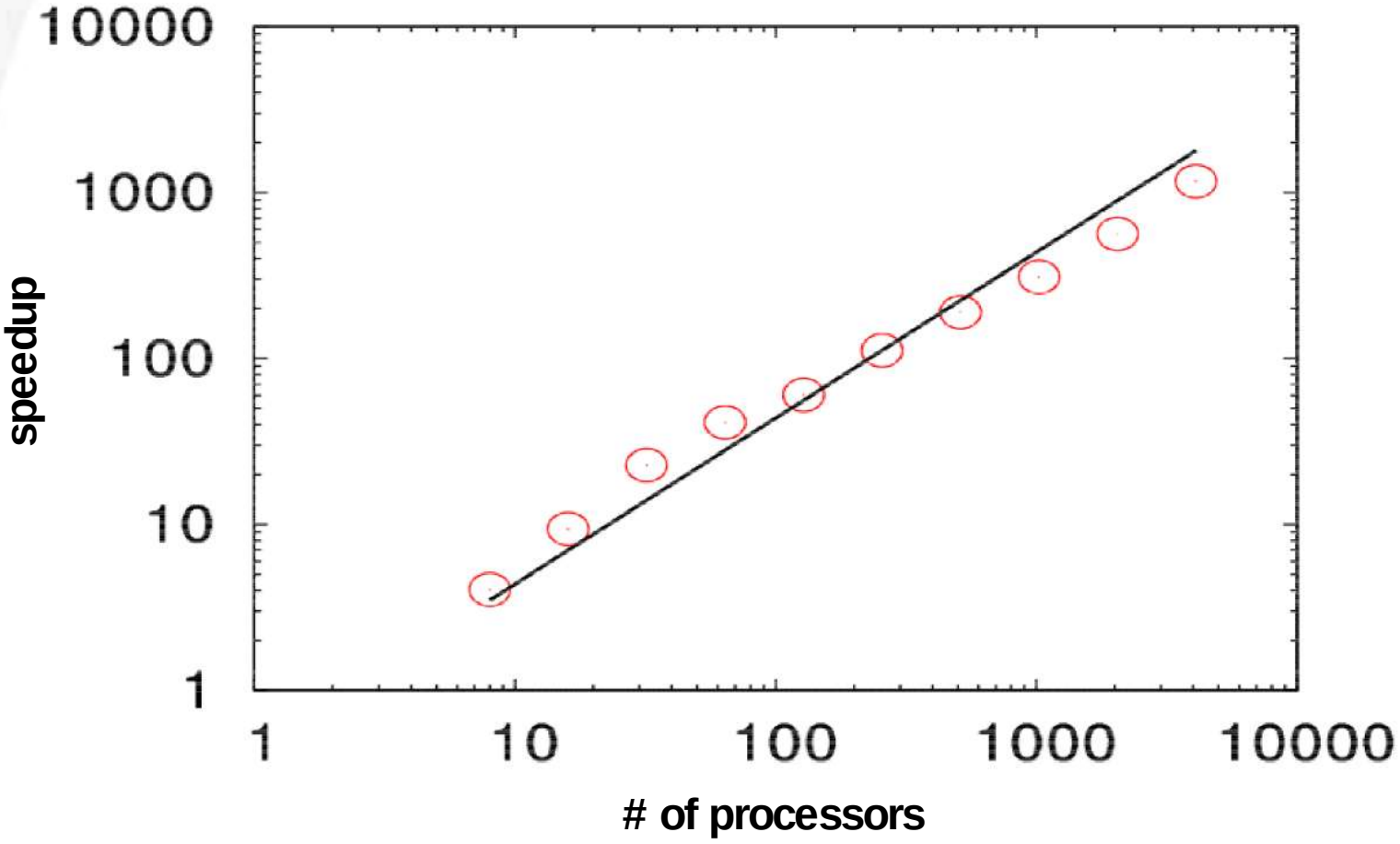
- Object-oriented framework supports plasma simulations with electromagnetics, electrostatics, neutral flow, etc.
- Template meta-programming allows a single code base to support simulations of 1,2 or 3 dimensions.
- Overlap of communication and computational result and effective domain decomposition in good parallel scaling.
- Independent numerical libraries for physics processes (e.g. ionization or secondary emission) extend VORPAL capabilities and allow other codes to benefit.

VORPAL has an efficient particle-in-cell scheme to couple particle and field dynamics



Overlapping communication (dashed lines) and computation (solid line) is one of the reasons VORPAL simulations scale to many processors

VORPAL simulation speedup scales to 1000s of processors



Simulations of laser-plasma interaction performed at NERSC show near-linear speedup to 4000 processors

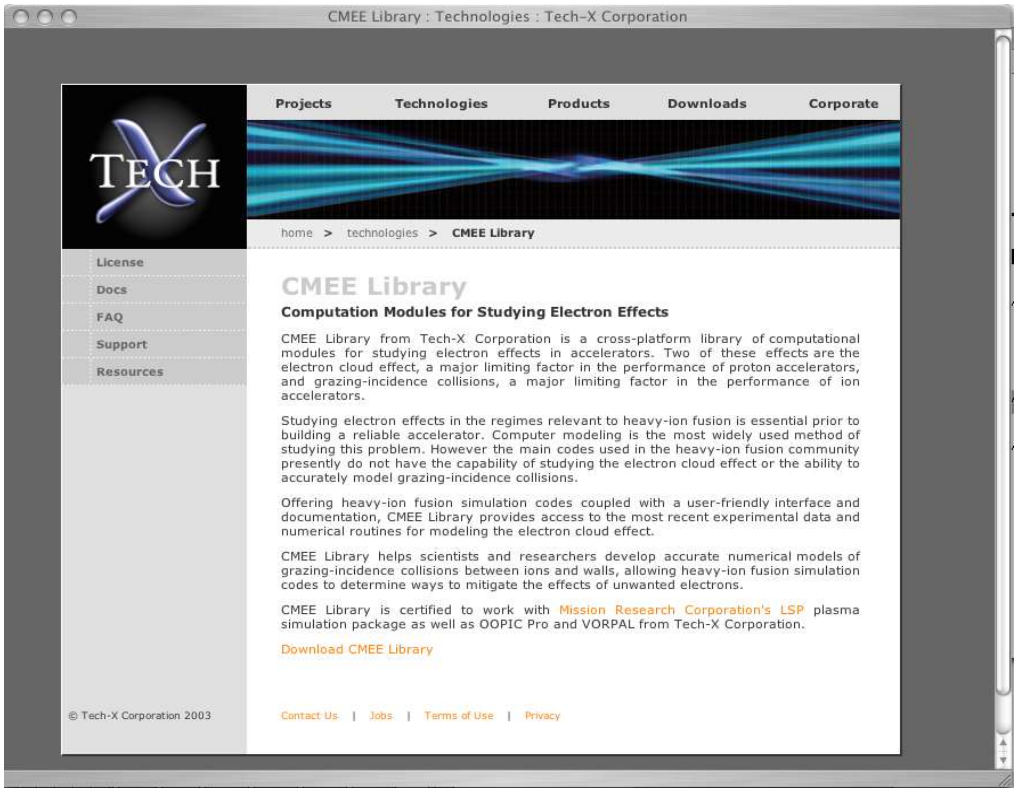
VORPAL uses CMEE library for including ionization and secondary electron models

- CMEE = Computational Modules of Electron Effects

- The goal is to provide routines for modelling
  - –secondary electron yield
  - –ion-induced electron yield
  - –ion and neutral gas desorption
  - –ion/neutral gas ionization and stripping

- The approach is
  - –use tested routines from community where possible
  - –use tables of data and interpolation routines
  - –make them available on any platform or language

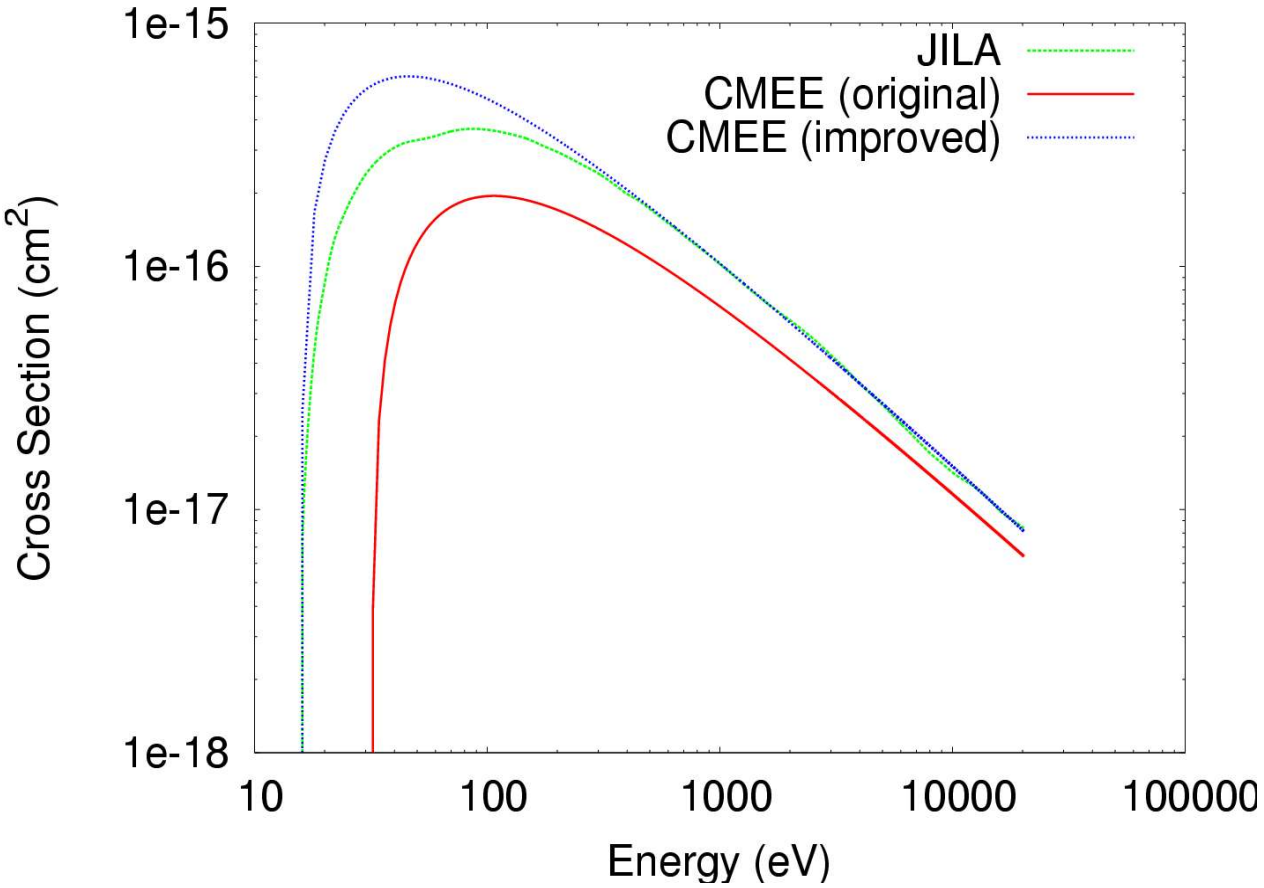
CMEE library v1.1 downloadable from the Tech-X website



http://www.txcorp.com/technologies/CMEE/

CMEE v1.1 includes documentation, examples, and bindings for Java, C, Python, and Fortran

We use the CMEE cross sections for impact ionization of Ar in diode simulations



We recently improved CMEE cross sections for Argon to match JILA data (L.J. Kieffer, Joint Institute for Laboratory Astrophysics, JILA report 13, 1973).

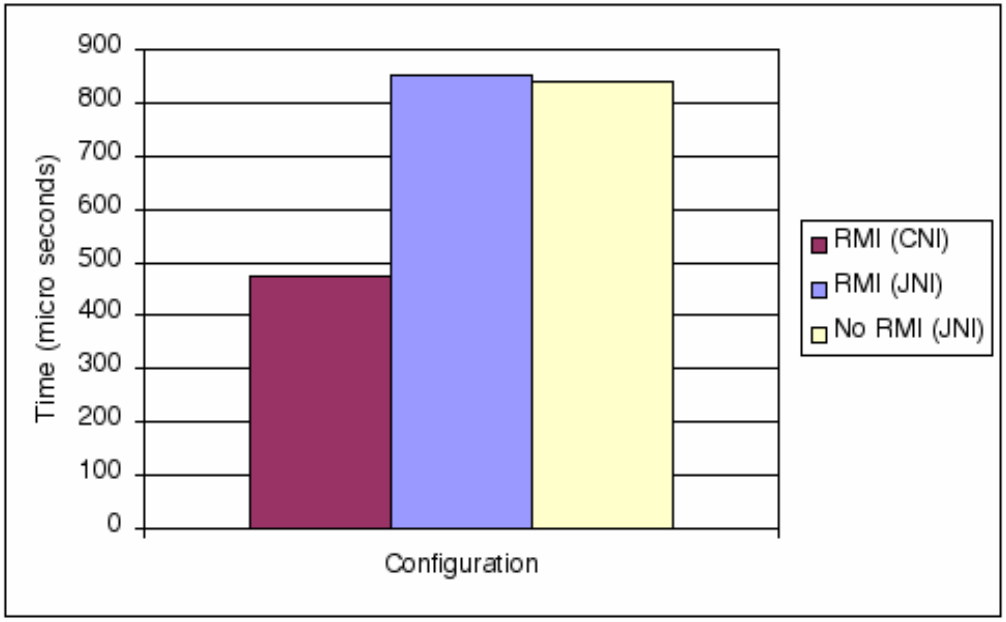
Example: Java code for using CMEE impact ionization routines

```
// These routines use C pointers so we set up some SWIG-provided
// pointers (intp and doublep)
//
// This is an integer flag to specify electron (0) or ion (1)
intp incident = new intp ();
incident.assign (0);

// Initialize the kinds of gases
//
// 0 = H, 1 = H2, 2 = He, 3 = CO2, 4 = CO, 5 = O2, 6 = N2, 7 = Ar, 8 = Ne
intArray inames = new intArray (numEls.value());
inames.setitem (0, 1);
inames.setitem (1, 2);
inames.setitem (2, 7);
inames.setitem (3, 8);

// This call fills the variables numColls and which_gs
// which_gs is the integer ID of the gas a particles collides with,
// It will be -1 if a particle doesn't collide
cmee.get_Collisions(numColls.cast(), which_gs.cast(), dt.cast(),
                    nparts.cast(), numEls.cast(), vels.cast(),
                    inames.cast(), gas_dens.cast(), incident.cast());
```

We are also looking at Cygnus Native Interface (CNI) to speed Java access to CMEE

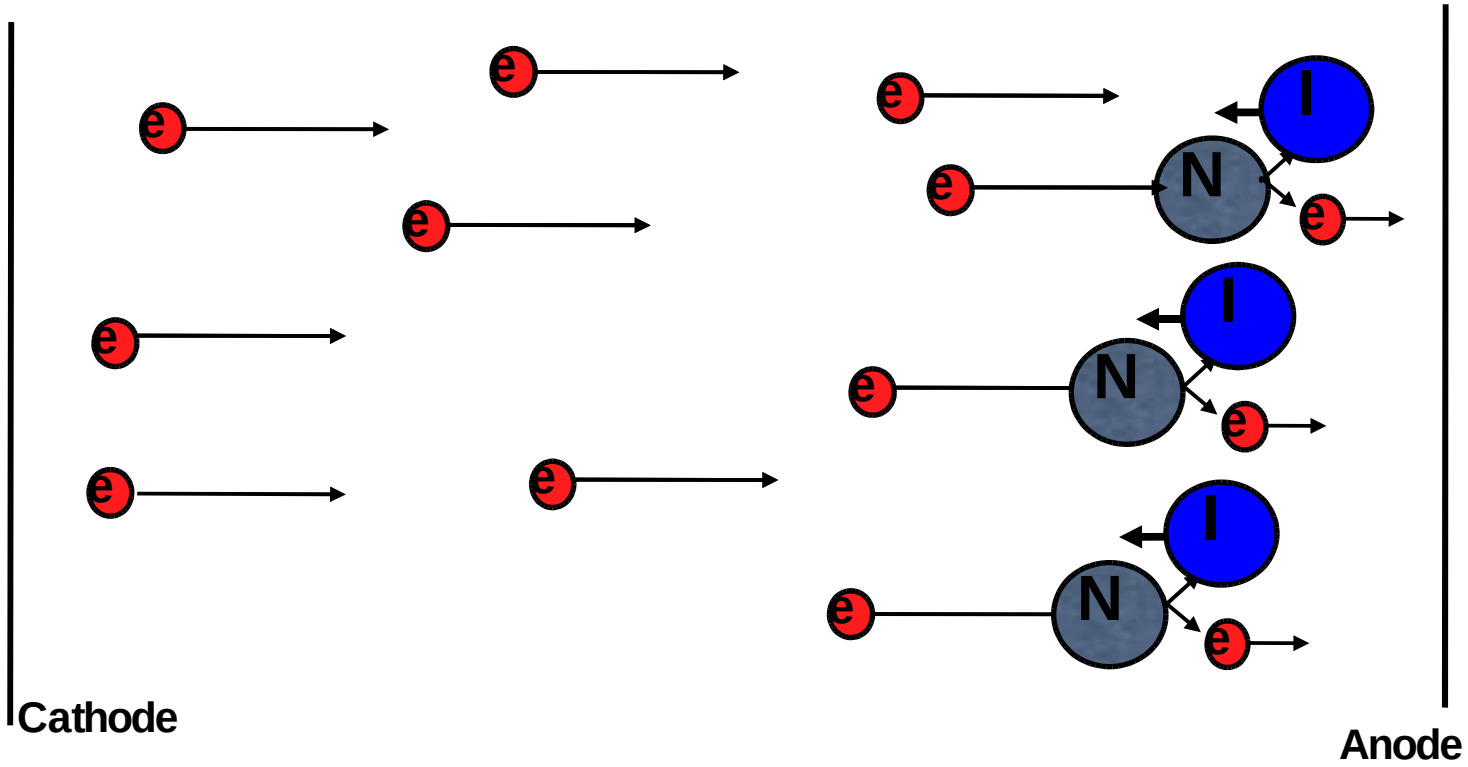


In CNI, Java classes are C++ classes (but not the other way around). The entire application (C++ and Java) is compiled to an executable, making access faster than through JNI.

We choose the conditions such that ionization is most likely near the anode

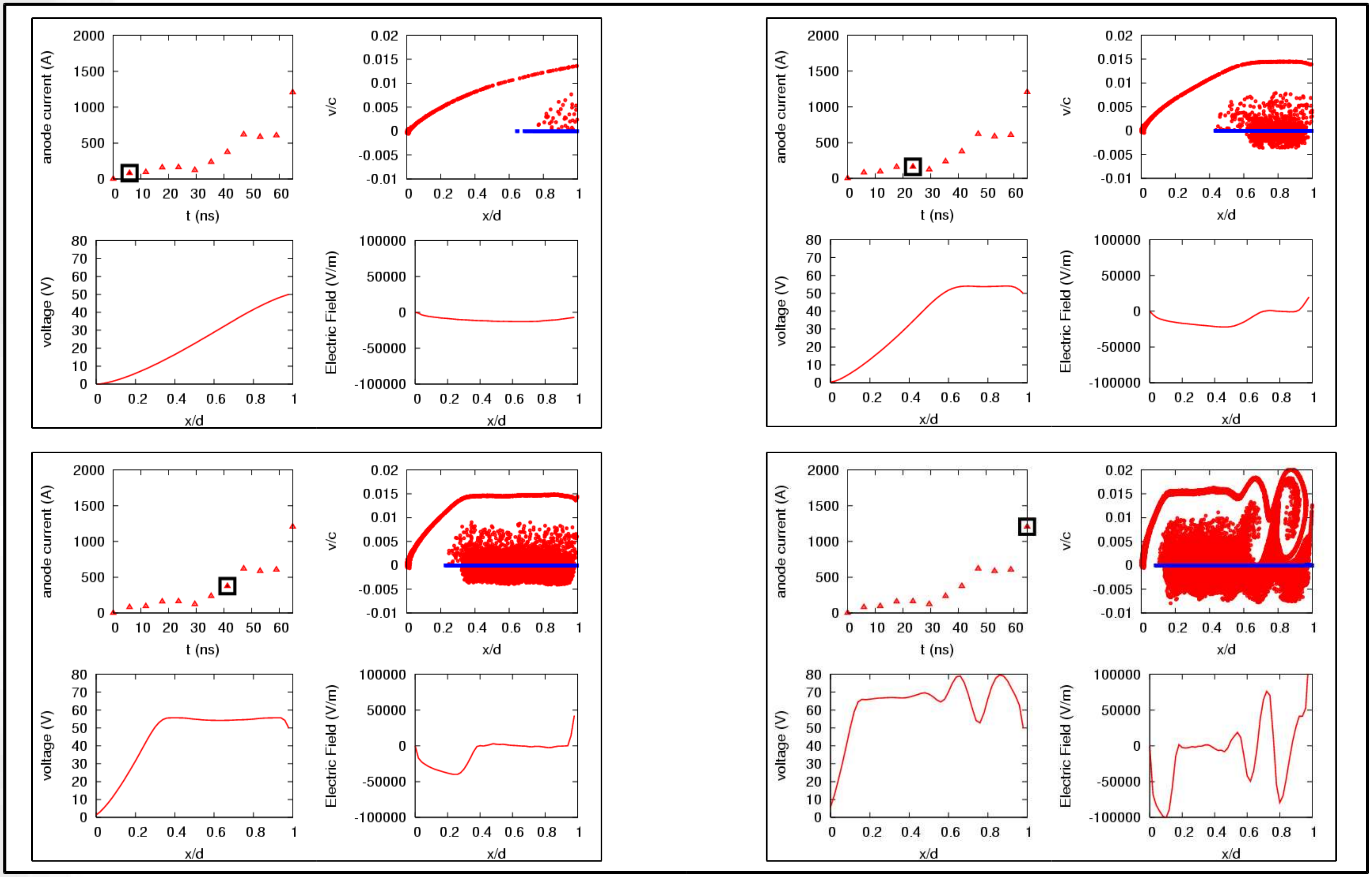
Physical Parameters
- V=50 V
- d=5 mm
- p=177 Pa Argon

Simulation Parameters
- dx = 100 microns
- dt = 10 ps
- $N_{pseudo\text{-}p}=2\ \ 10^9$



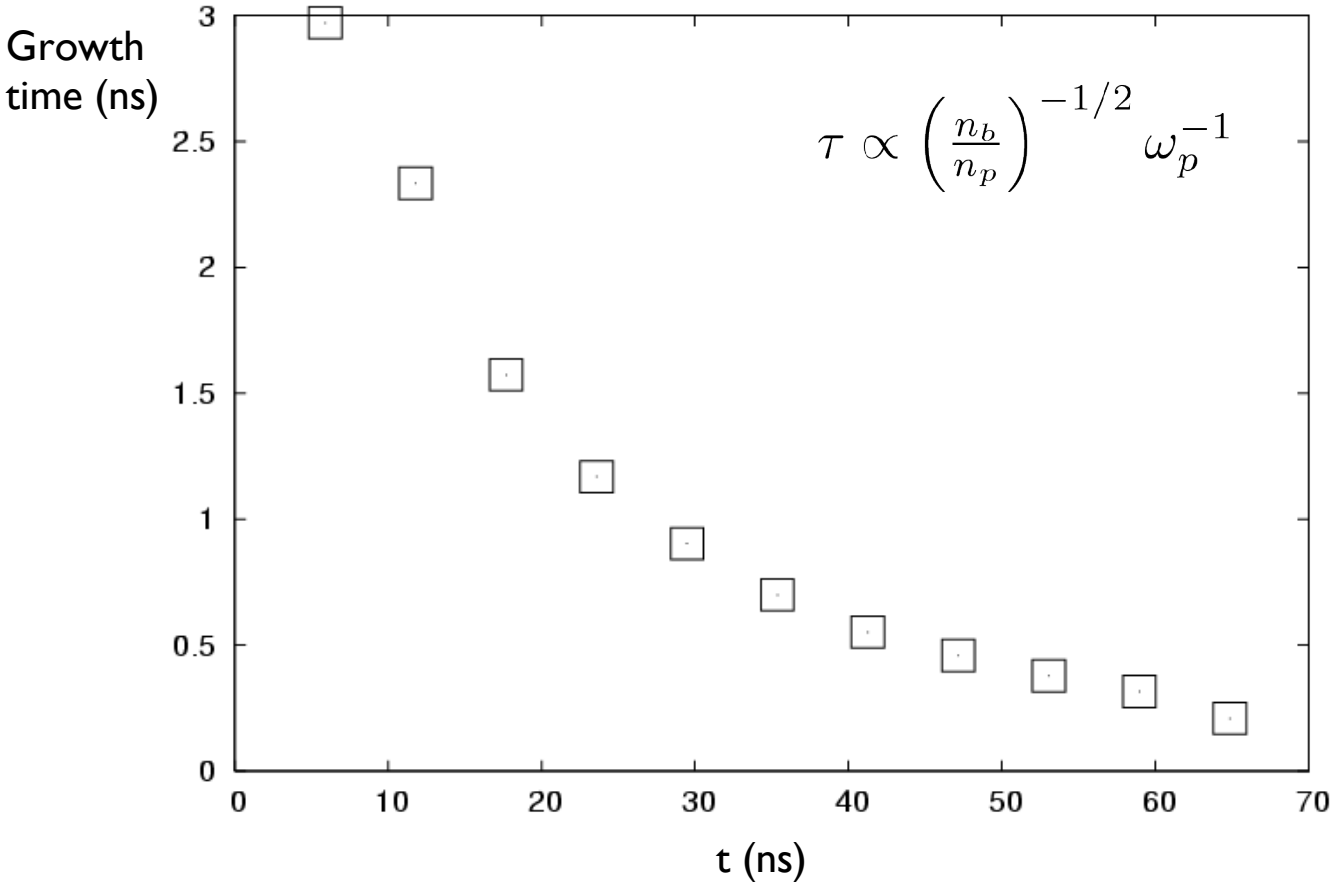
Simulations show enhanced current and a two-stream instability after 65 ns

Electrons (red) impact ionize background gas, leaving behind ions (blue)

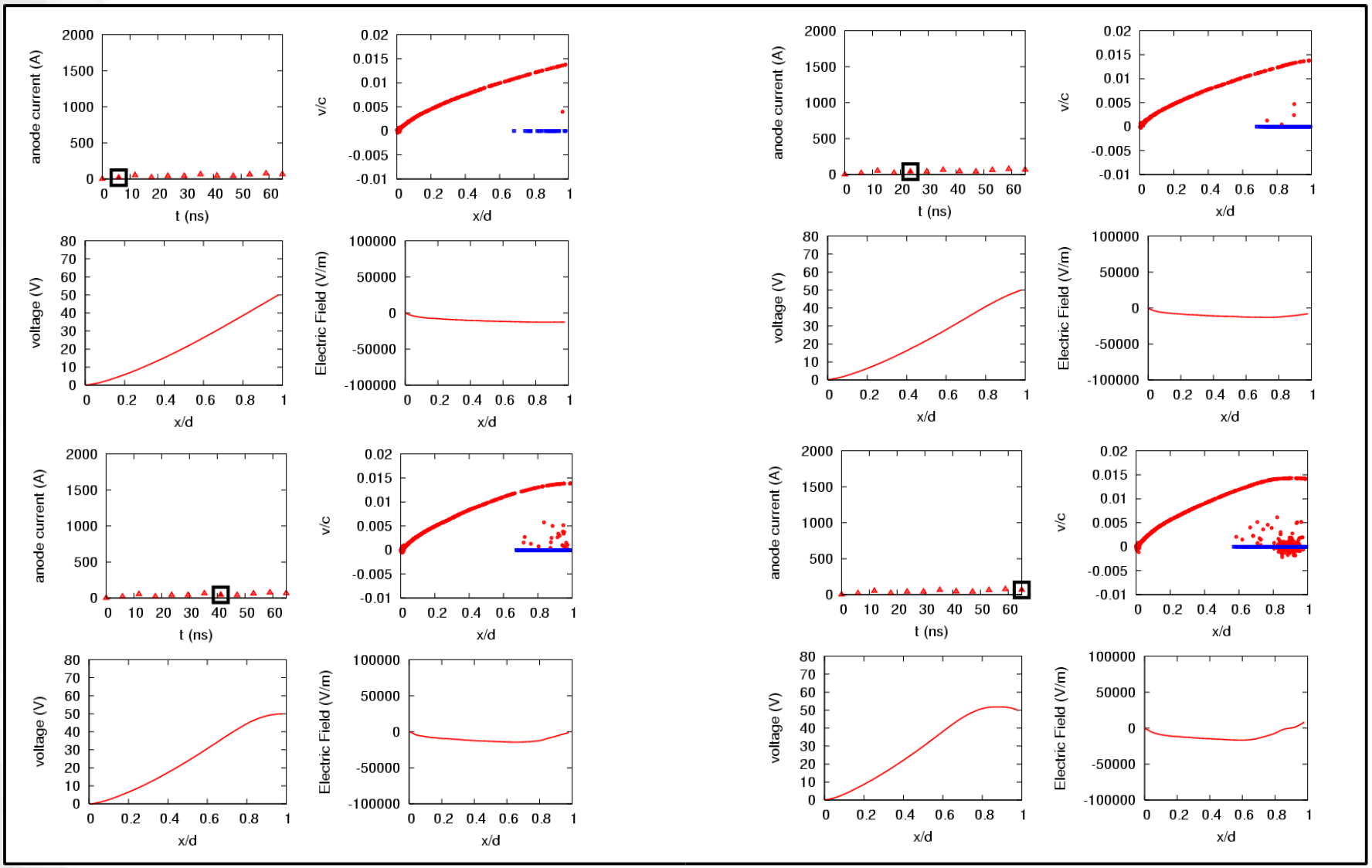


Two-stream linear growth rate is consistent with onset of instability

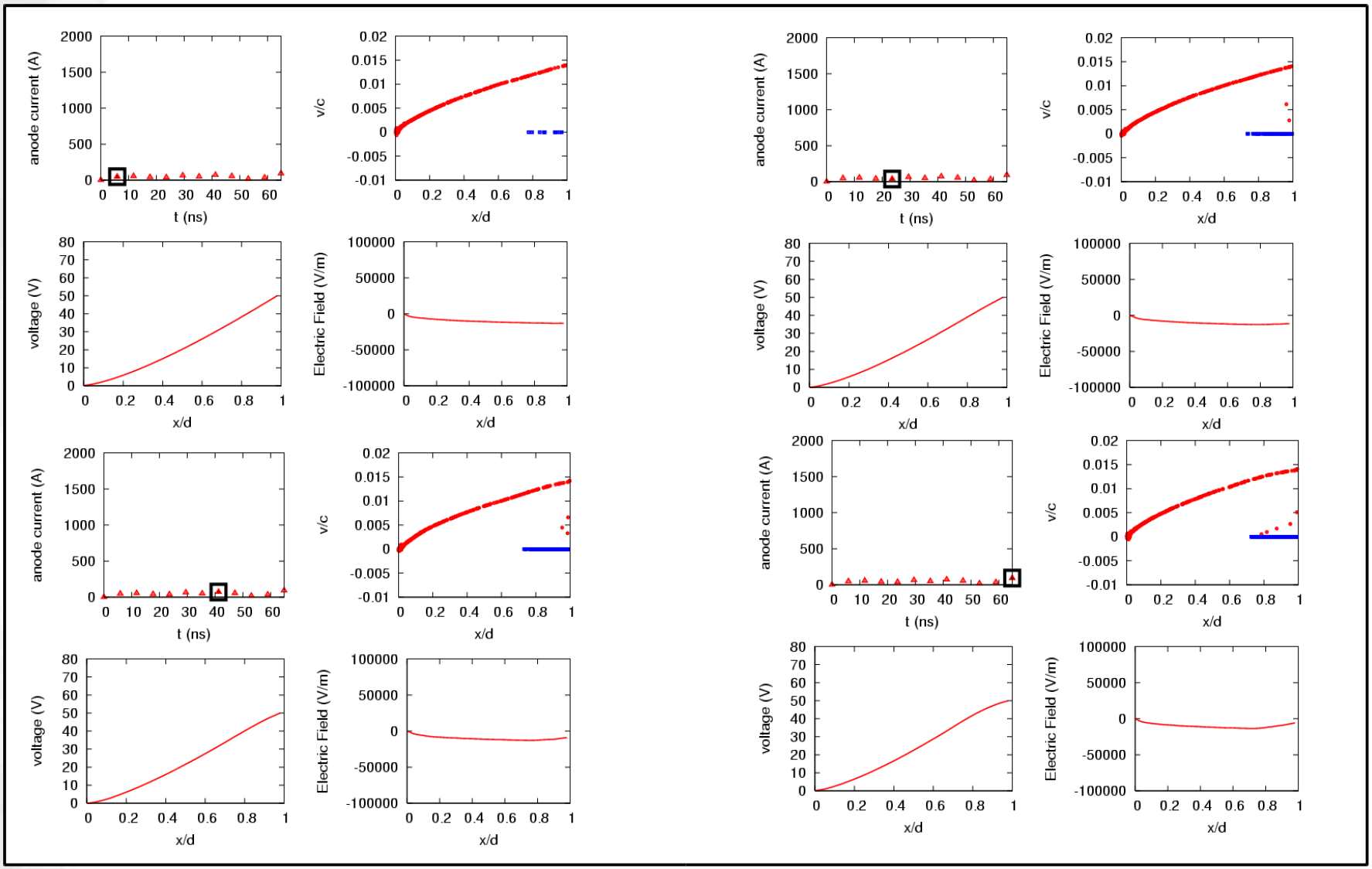
The growth time from linear theory is consistent with an instability over a few 10s of nanoseconds, as seen in the simulations



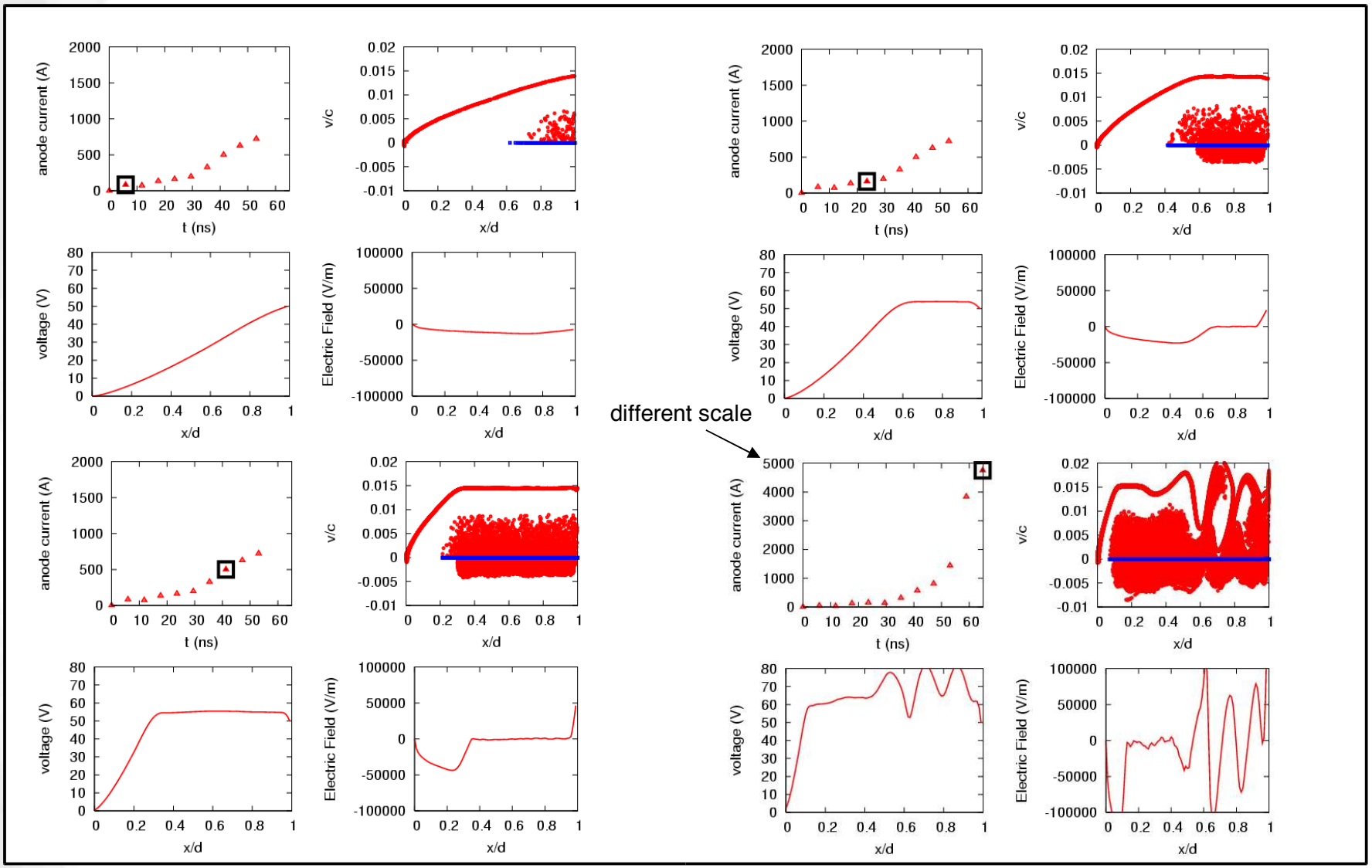
Simulations with 1/5th the density show slower growth rate



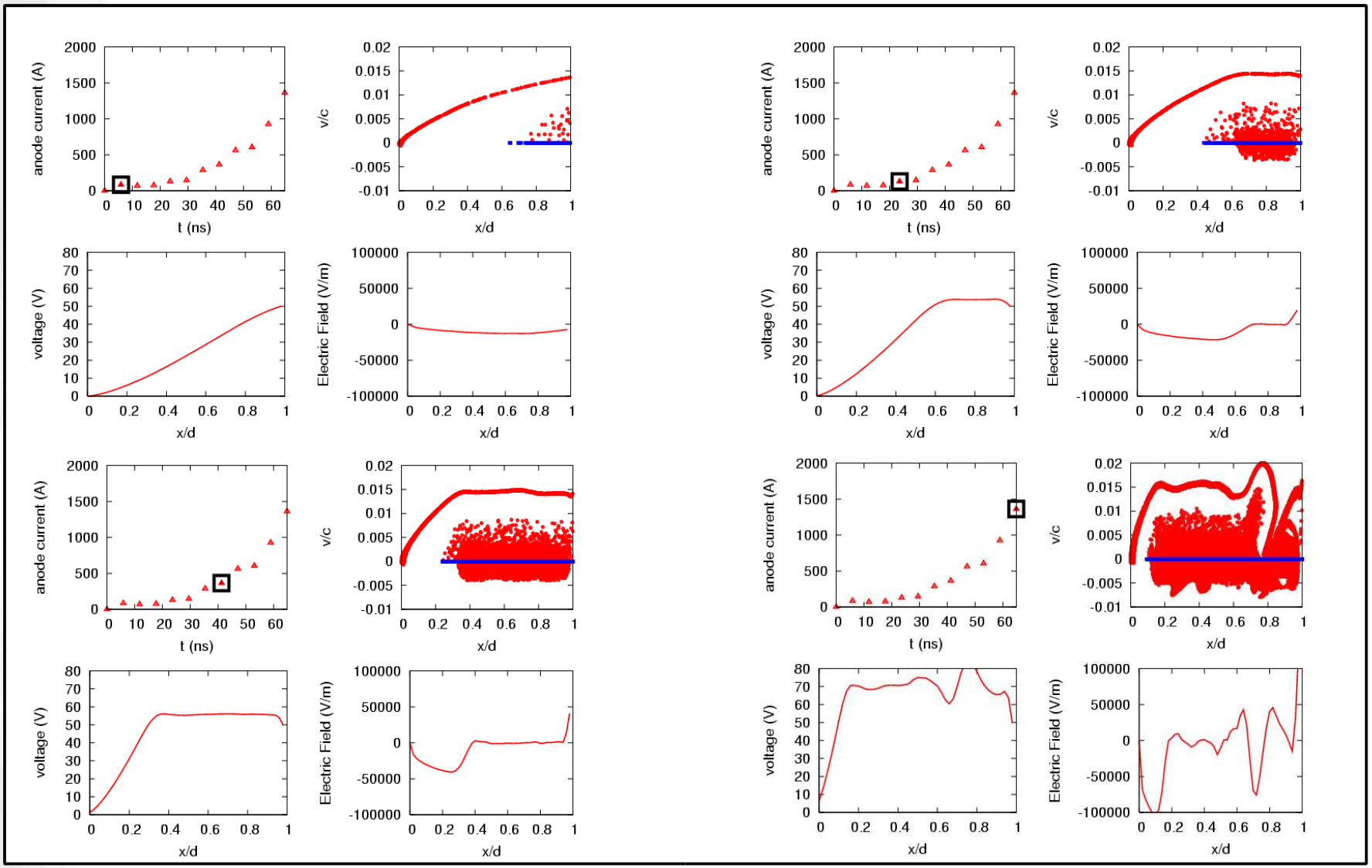
Simulations with 1/10th the density also show slower growth rate



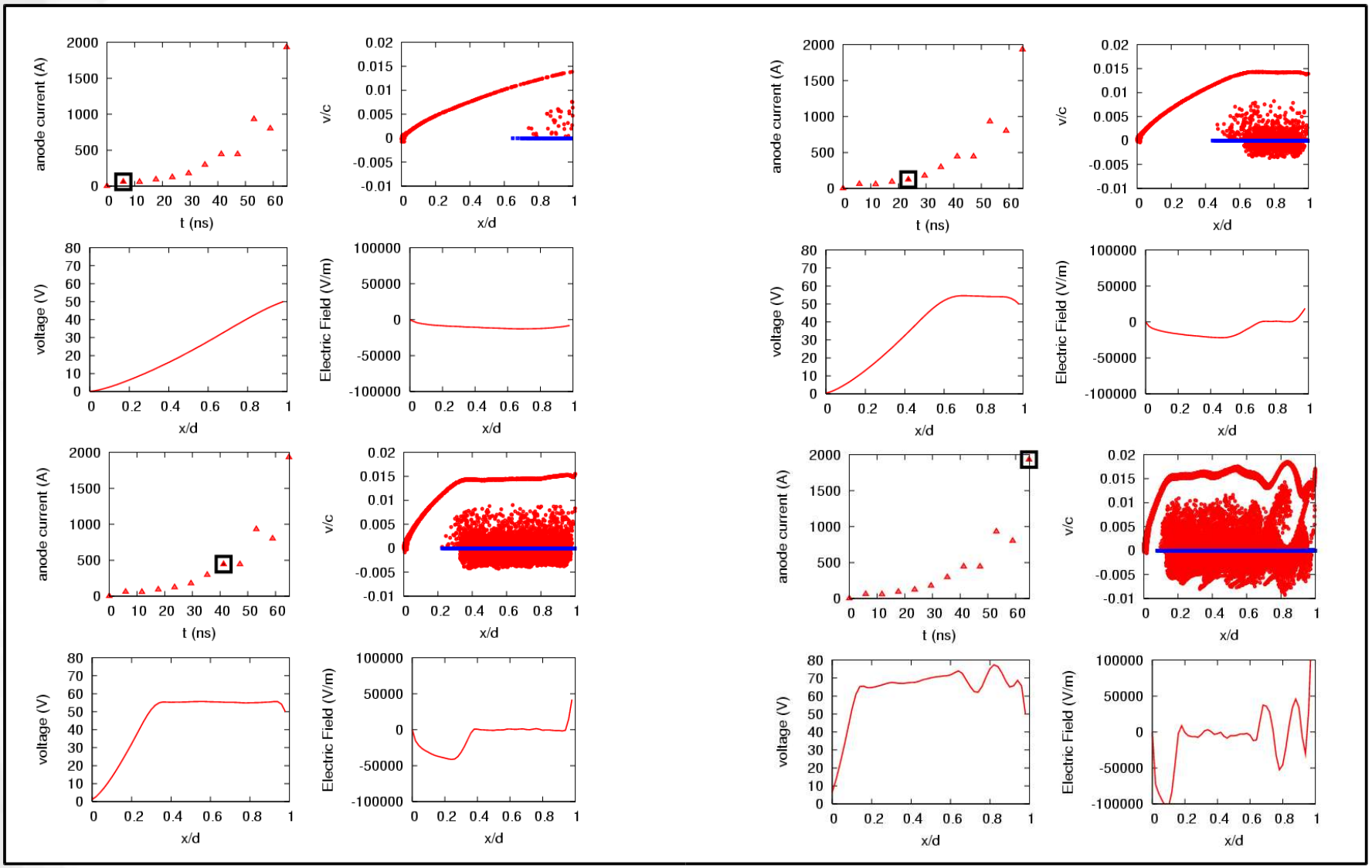
Simulations at twice the resolution show similar results



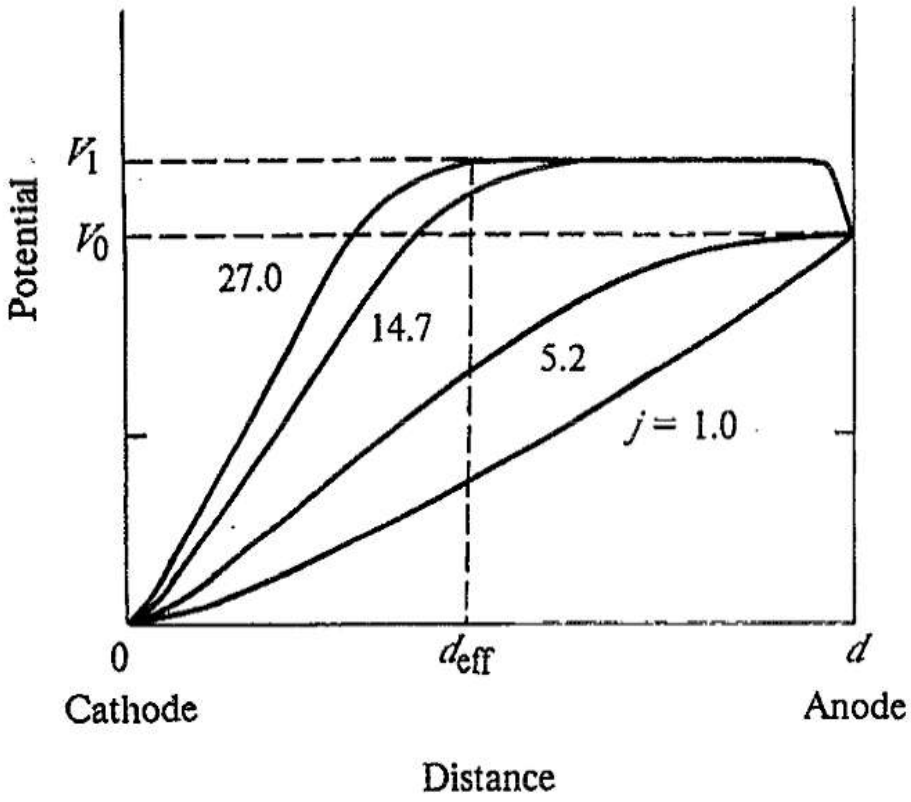
Simulations with immobile ions also show similar results



Simulations with ionization only from the beam show similar results as well



Simulated potential in qualitative agreement with theoretical model of Kozyrev, et al.*



The potential in the simulations develops similarly to the predictions by the theory of Kozyrev et al., providing more confidence in the simulation results

*A. Kozyrev, Y. Korolev, V. Rabotkin and I. Shemyakin, *J. Appl. Phys*. **74**(9), 5366-5371 (1993)

CONCLUSIONS

- We are modifying the VORPAL code to include effects important for simulating gas-filled ion diodes

- We are including effects such as impact ionization through the CMEE library, a cross-platform, cross-language collection of routines for modelling electron effects.

- Simulations of gas-filled diodes with parameters such that the mean-free-path of an electron is of the order of the diode spacing show a two-stream instability

- This instability persists for varying gas densities and ion mobilities