UKAEA-CCFE-PR(21)19

S. H. Ward, R. Akers, A. Jacobsen, P. Ollus, S. D. Pinches, E. Tholerus, R. G. L. Vann, M. A. Van Zeeland

Verification and validation of the high-performance Lorentz-Orbit Code for Use in Stellarators and Tokamaks (LOCUST)

Verification and validation of the high-performance Lorentz-Orbit Code for Use in Stellarators and Tokamaks (LOCUST)

S. H. Ward, R. Akers, A. Jacobsen, P. Ollus, S. D. Pinches, E. Tholerus, R. G. L. Vann, M. A. Van Zeeland

This is a preprint of a paper submitted for publication in Nuclear Fusion

Verification and validation of the high-performance Lorentz-Orbit Code for Use in Stellarators and Tokamaks (LOCUST)

S. H. Ward

York Plasma Institute, Department of Physics, University of York, York YO10 5DD, UK
Culham Centre For Fusion Energy, Culham Science Centre, Abingdon, OX14 3DB, UK
ITER Organization, Route de Vinon-sur-Verdon, CS 90 046, 13067 St. Paul Lez Durance Cedex, France

R. Akers

Culham Centre For Fusion Energy, Culham Science Centre, Abingdon, OX14 3DB, UK

A. S. Jacobsen

P. Ollus

Department of Applied Physics, Aalto University, FI-00076 Aalto, Finland

S. D. Pinches

ITER Organization, Route de Vinon-sur-Verdon, CS 90 046, 13067 St. Paul Lez Durance Cedex, France

E. Tholerus

Culham Centre For Fusion Energy, Culham Science Centre, Abingdon, OX14 3DB, UK

R. G. L. Vann

York Plasma Institute, Department of Physics, University of York, York YO10 5DD, UK

M. A. Van Zeeland

General Atomics, P.O. Box 85608, San Diego, California 92186-5608, USA

E-mail: samuel.ward@york.ac.uk

Abstract.

A novel high-performance computing algorithm, developed in response to the next generation of computational challenges associated with burning plasma regimes in ITER-scale tokamak devices, has been tested and is described herein.

The Lorentz-Orbit Code for Use in Stellarators and Tokamaks (`LOCUST`) is designed for computationally scalable modelling of fast-ion dynamics, in the presence of detailed first wall geometries and fine 3D magnetic field structures. It achieves this through multiple levels of single instruction, multiple thread (SIMT) parallelism and by leveraging general-purpose graphics processing units (GPGPU). This enables `LOCUST` to rapidly track the full-orbit trajectories of kinetic Monte Carlo markers to deliver high-resolution fast-ion distribution functions and plasma-facing component power loads.

`LOCUST` has been tested against the prominent `NUBEAM` and `ASCOT` fast-ion codes. All codes were compared for collisional plasmas in both high and low-aspect ratio toroidal geometries, with full-orbit and guiding-centre tracking. `LOCUST` produces statistically consistent results in line with acceptable theoretical and Monte Carlo uncertainties. Synthetic fast-ion D-α diagnostics produced by `LOCUST` are also compared to experiment using `FIDASIM` and show good agreement.

1. Introduction

The size, power and performance of the ITER tokamak represents a paradigm shift in both experimental and computational fusion science. ITER will be one of the first tokamaks to generate a burning plasma, where a significant population of energetic particles must be sufficiently confined to simultaneously sustain the reaction and protect the plasma-facing components (PFC). Hence the transport of energetic particles in ITER remains a vital area of study [1] (see the activities of the ITPA Energetic Particle Physics Topical Group). Nevertheless, the spatiotemporal scales involved make detailed systematic analyses challenging with traditional computational methods. Resolving heat loads over the greater PFC surface areas of larger tokamaks requires evaluating additional gyro-resolved trajectories. Similarly, the extra compute time required to track ions over the increased ($\sim$ 1s) slowing-down time is compounded by the need for smaller timesteps ($\sim$ ns) and finer topological resolution to minimise numerical drift. With similar large-scale devices on the horizon, such as STEP, this type of metaproblem has recently begun to attract more urgent attention [2][3][4].

There are multiple ways to advance computational capabilities, and adapting to novel or specialised hardware is one. This approach is advantageous for a number of reasons other than an immediate speed boost: the lower cost, energy consumption and space required can make specialised hardware more efficient for specific tasks; specialised hardware can be more accessible at the hardware level, for example interfacing with workstation devices directly via PCIe buses to avoid the need for remote data centres; minimal adaptation is required for modular or encapsulated code; and future hardware generations bring passive performance improvements more rapidly, depending on the type of hardware market [5]. This costs time tuning and adapting codes and algorithms. Additionally, sometimes hardware may be inaccessible if particularly new or prohibitively expensive for individual users. But high-performance computing platforms, including the cloud, now routinely offer heterogeneous hardware, often featuring combinations of central processing units (CPU), GPGPUs and even field-programmable gate arrays (FPGA). Likewise, user-level accessibility is constantly improved by high-level APIs [6][7]. For embarassingly parallel tasks, which describe most linear energetic particle physics, GPGPUs in particular offer scalable hardware acceleration for little deployment cost; GPGPUs are energy-efficient, have a low capital cost per computational thread and can be hosted by basic desktop computers. Despite this, the application of GPGPUs to computational plasma physics is not widespread.

`LOCUST` is an algorithm designed specifically to use off-the-shelf GPGPU hardware to reach a computational performance that enables routine simulations of fast-ions in ITER-scale devices. For an ITER burning plasma scenario (electron density $\sim 10^{20}\mathrm{m}^{-3}$, electron temperature $\sim$ 25keV, 12 3D field components at 1cm precision and a wall mesh comprising 6×10^7 tetrahedra/3×10^6 surface triangles), `LOCUST` can track $250,000$ markers to thermalisation over 2s in 1ns time steps in 30 hours on a node with eight Nvidia P100 GPGPUs controlled by one Intel Xeon E5-2689. This performance itself demonstrates that `LOCUST` is readily capable of studying the most challenging of problems.

As part of the software lifecycle, it is vital to continually verify that new tools like `LOCUST` are correctly implemented and to validate the accuracy of any underlying models by comparing with experiment. To this end, `LOCUST` has been benchmarked over a range of test scenarios, with the aim that its performance matches multiple well-established tools. Emphasis has subsequently been placed on accurately verifying the fundamental implementation of the code, whilst reducing possible interference from high-order physics models, before validating this against experiment.

In this paper, `LOCUST` is both described and rigorously tested. Section 2 describes the physical model, including its assumptions, their computational implications and the resulting information calculated by `LOCUST`. A description of the algorithm and its execution, alongside the required inputs and ouputs, is given in section 3. Results from cross-code benchmarks are presented in 4. Finally a summary of findings is made in section 5.

2. Model overview

The primary goal of `LOCUST` is to calculate the steady-state distribution function of fast-ion species as efficiently as possible, whilst also resolving individual ion trajectories in a realistic 3D geometry to calculate PFC power loads. To complement massively parallel memory-bound hardware, we opt for a mathematically simple but computationally intensive approach: solving the Lorentz equation of motion (1) for individual kinetic markers i representing real-space position $\boldsymbol{r}_i(t)$ and velocity $\boldsymbol{v}_i(t)$ Monte Carlo samples of the fast-ion distribution function:

$$\frac{\mathrm{d}^2\boldsymbol{r}_i}{\mathrm{d}t^2}=\frac{\mathrm{d}\boldsymbol{v}_i}{\mathrm{d}t}=\frac{q_i}{m_i}(\boldsymbol{v}_i\times\boldsymbol{B}(\boldsymbol{r}_i)+\boldsymbol{E}(\boldsymbol{r}_i)). \quad (1)$$

with $\boldsymbol{B}(\boldsymbol{r}_i)$ and $\boldsymbol{E}(\boldsymbol{r}_i)$ the magnetic and electric fields evaluated at the i^{th} marker position $\boldsymbol{r}_i$. In LOCUST these trajectories are evaluated either by tracking the particle's guiding-centre (GC) [8] and gyro-phase or its real-space position over the full orbit (FO), using a fixed-step numerical scheme to minimise thread divergence. Multiple such integrators are included: Strang-Splitting [9]; the recent BGSDC [10]; Runge-Kutta-type integration methods such as RK45 [11], McClements-Thyagaraja-Hamilton, Fehlberg [12], Cash-Karp [13], Dormand-Prince [14], and Goeken-Johnson [15]; and Euler methods such as the popular Leapfrog/Boris [16].

LOCUST relies on approximations to enhance the computational tractability of the model. Firstly, the electromagnetic field experienced by each fast particle is evaluated without the relatively weak contributions from other fast-ion species. If we further assume that the background plasma is in static equilibrium and self-consistent with the fast-ion distribution, we can simplify the computation in three ways:

(i) We ignore the influence of fast ions on the thermal plasma. If and only if we chose to do so then it becomes valid to track independent sub-samples of the fast-ion distribution (see item (ii)). Without this approximation, these sub-samples of the fast-ion population would be self-coupled via their exchanges with the bulk plasma, but they can be treated independently in the case where the background equilibrium is assumed to be static and self-consistent with the final steady-state fast-ion distribution. In practice, equilibria are taken from preconverged time-dependent transport simulations, which use simplified fast-ion models. In a given LOCUST simulation, the equilibrium is held constant by fixing the background plasma temperature, density and rotation profiles as well as the magnetic field.

(ii) We utilise the entire trajectory history of each marker when calculating the distribution function such that a marker ensemble of constant size need only be tracked from source to sink once to estimate the steady-state distribution function. By extending the logic from item (i), one can treat individual points along a marker trajectory as independent subsamples of a common distribution function. This effectively parallelises the calculation across time similar to other fast-ion codes such as ASCOT [17].

(iii) Each fast ion is tracked independently in parallel by non-blocking processes. This is again enabled by item (i).

Without interactions between fast-ion species, the equilibrium $\boldsymbol{E}(\boldsymbol{r}_i)$ and $\boldsymbol{B}(\boldsymbol{r}_i)$ terms are dominated by background sources, such as external field coils and short-range Coulomb interactions with the thermal species. Whilst the background equilibrium is prescribed numerically, the short-range interactions with bulk species are replicated using a stochastic perturbation to the marker velocity vector term in equation 1. This is implemented with a Monte Carlo Fokker-Planck collision operator that includes terms for diffusion and drift of pitch angle (λ) and energy (ϵ). These operators are based upon the binomial operators derived in [18]:

$$\Delta\lambda_i = -\lambda_i \nu_{\mathrm{d}} \Delta t \pm \left[\left(1-\lambda_i^2\right)\nu_{\mathrm{d}}\Delta t\right]^{1/2} \tag{2}$$

$$\begin{aligned}\Delta\epsilon_i = &-2\nu_\epsilon \Delta t\left[\epsilon_i - \left(\frac{3}{2}+\frac{\epsilon_i}{\nu_\epsilon}\frac{\mathrm{d}\nu_\epsilon}{\mathrm{d}\epsilon}\right)k_{\mathrm{B}}T_{\mathrm{th}}\right]\\ &\pm 2\left(k_{\mathrm{B}}T_{\mathrm{th}}\epsilon_i\nu_\epsilon\Delta t\right)^{1/2}\end{aligned} \tag{3}$$

where ± is a random sign with equal probability and the ν_ϵ and ν_{d} terms represent collision frequencies as functions of the Coulomb logarithm $\log(\Lambda)$ and error function [19][20]. For a given fast-ion population, these equations are solved for each bulk species. An option to accelerate the effects of these equations is also included [21].

By solving these equations, LOCUST calculates and outputs a range of physics results in many formats. 3D power loads are derived from the intersections of orbits and PFCs. Similarly, Hamiltonian field line trajectories can be efficiently evaluated to create various types of Poincaré maps, each designed to illustrate particular magnetic field structures. The distribution function can be generated in $[R,Z,v,\lambda]$ and constants-of-motion $[\epsilon,P_\phi,\sigma,\mu]$ spaces. Here P_ϕ is the fast-ion canonical angular momentum, σ the sign of the first-order guiding-centre pitch and μ the magnetic moment, which is expanded to first order [22] to improve accuracy when binning markers in devices with steep field gradients. LOCUST also calculates one-dimensional poloidal flux profiles of fast-ion-driven current, torque ($\boldsymbol{J}\times\boldsymbol{B}$ and collisional), pressure and heating to bulk species channels.

3. Code design and execution

The overall data flow to and from LOCUST is illustrated in figure 1, along with preprocessing stages and related external physics solvers. The background equilibrium fields describing the 2D axisymmetric and 3D perturbative components are passed to LOCUST as separate numerical representations. These can be in the form of IMAS Interface Data Structures (IDS), GEQDSKs, 3D rectilinear grids, or Fourier-decomposed data. The bulk species temperature and density are supplied to the collision operator numerically as interpolated functions of poloidal magnetic flux. The initial fast-ion ensemble is read from individual marker phase-space positions, often generated by external plasma heating codes. Finally, the 3D PFC power flux can be calculated from an axisymmetric limiter outline or an unstructured volumetric tetrahedral mesh, avoiding the need for runtime octrees. 3D meshes are generated by defeaturing and repairing elements of computer-aided design (CAD) engineering models, typically using CADfix or SpaceClaim, before remeshing volumetrically in Attila [23]. Equally this geometry can be represented by the `IMAS` generic grid description (GGD) in the wall IDS.

Once runtime settings and input data are specified, execution proceeds through stages 1-7 in figure 1. First, the X-point, magnetic axis and last-closed flux surface (LCFS) are precisely located. Since `LOCUST` is not storage bound, the rectangular 2D field is then effectively cached by storing pre-computed bicubic spline coefficients for each knot - either just the required derivatives and cross-derivatives or the entire set. Whilst this method is also offered in `LOCUST` for storing 3D fields as tricubic splines on a rectilinear grid, a Fourier-decomposed format is also available. This latter option is preferred if on-board memory begins to limit grid resolution or when resolution is only needed in particular dimensions; the freed space may be used to redistribute resolution, reducing magnetic field divergence enough to enable linear interpolation - which is faster. The 3D mesh is then cached by labelling vacuum-facing triangles as PFC surfaces. For rapid and synchronous tracking of tetrahedra traversal, nodes comprising the unstructured wall mesh are also mapped to a coarse cartesian grid and adjacent tetrahedra are linked [33].

After runtime preprocessing, marker tracking is performed in two stages: markers which start and remain inside the LCFS are first tracked *without* PFC interception checks before tracking all remaining markers *with* PFC checks. Considering the constraints outlined in section 2, optimal performance is achieved if the number of occupied threads is maximised, indicating GPGPUs are currently the most suitable hardware - with thousands of threads per in-built streaming multiprocessor each able to track an individual marker in parallel. The Nvidia `CUDA` application-programming interface (API) is used to interface with this hardware, and the core tracking algorithm is written in PGI `FORTRAN`. `CUDA` also offers libraries for efficient random number generation, including `CURAND` and a Mersenne Twister [34] implementation which are optional in `LOCUST`.

Implementing SIMT algorithms of this type requires some hardware-specific design considerations. The marker tracking algorithm, which comprises the stages in box 5 of figure 1, is illustrated in figure 2. The required data is first transferred from the host CPU to the GPGPU device before a non-blocking device kernel is triggered to advance markers in phase space. Simultaneously, fast-ion positions from the previous timestep are sum-reduced on the host. Care is taken to overlap these processes; upon their completion, all processes are synchronised by a transfer of the next fast-ion positions to the host. These positions take the integer form of the corresponding distribution function bin indices to lower data throughput. This process is scaled across multiple devices within a node using `OpenMP`.

As marker trajectories are evaluated, the latest positions are cumulatively binned to build up the steady-state distribution function. This process, illustrated in figure 3, occurs until markers either strike a PFC, reach a tracking time limit or slow to a prescribed velocity cut-off - typically $3T_{\mathrm{bulk}}/2$. Upon completion, incident PFC power is collated and adaptively refined across the surface mesh.

4. Testing

4.1. Orbit tracking

The most fundamental test is to examine the full-orbit vacuum trajectories calculated by `LOCUST`. These are compared to equivalents calculated by `MPI_ORBF` [35] in the presence of a PFC wall model (in this case DIII-D). Figure 4 shows that, for markers initially distributed uniformly in $[R,\lambda]$ space along the outer horizontal midplane, both codes measure the same prompt loss boundary to within ($\mathrm{d}\lambda = 0.04$, $\mathrm{d}R = 1.0$cm). Variations of this magnitude in the loss boundary amount to changes of ∼0.1-0.4% in total loss fraction: 6.8% (`MPI_ORBF` with 2D wall), 6.9% (`LOCUST` with 2D wall) and 6.5% (`LOCUST` with 3D wall). Therefore the differences are within variations caused by the wall model, as shown by the strike point distribution.

4.2. Collisional transport

To test collisional transport in the presence of a toroidally symmetric background plasma, comparisons were made against `ASCOT4` [17] (version 5 has since been shown to produce similar results [36]) and the `NUBEAM` [21] module of `TRANSP` [37] via `OMFIT` [38]. `LOCUST`, `ASCOT` and `NUBEAM` all employ collision operators which reduce to the same Fokker-Planck equation (though different definitions of the Coulomb logarithm are used). Using data from a single time slice at 3s into DIII-D shot #157418 [39], the deposition of a monoenergetic counter-current-injected 80keV deuterium neutral beam into a static plasma

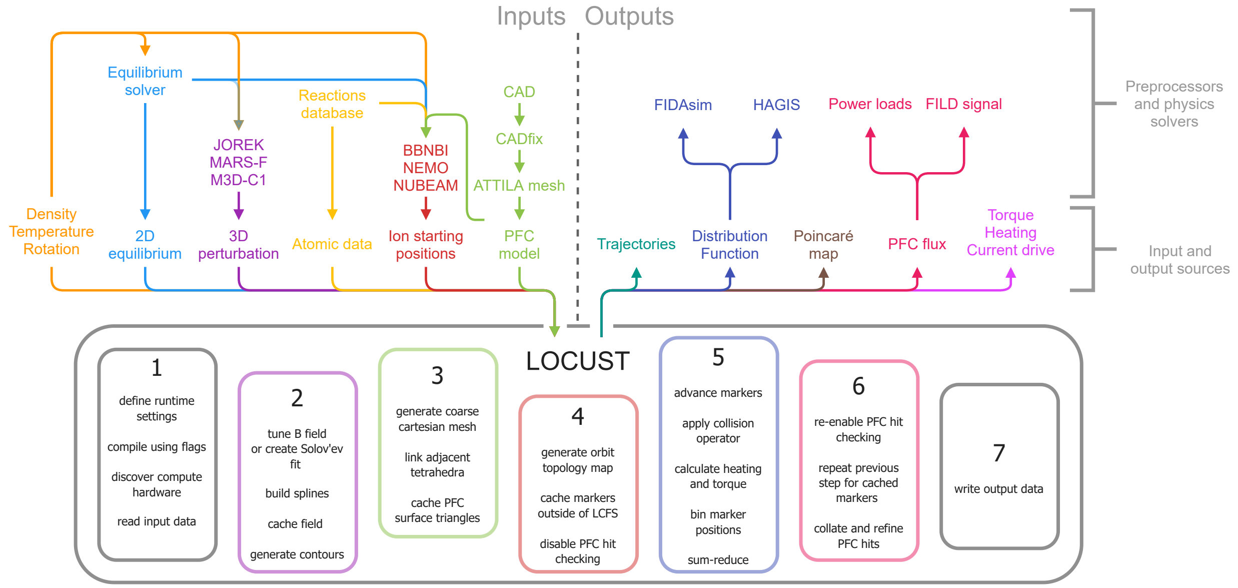

Figure 1. Information flow (arrows) to and from `LOCUST`, itself shown in the solid grey box with main pre-processing, runtime and postprocessing stages. These stages execute in ascending order one to seven and are colour-coded to match relevant input/output data. Included are examples of external physics codes which have been used to pre-process or post-process data [23][24][25][26][27][28][29][30][31][21][32].

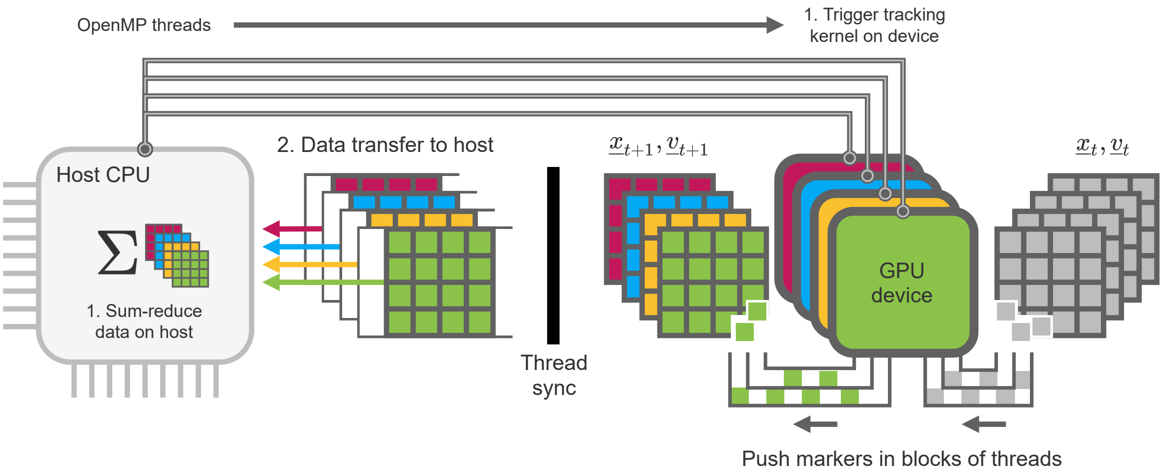

Figure 2. Execution model for the kernel-level marker tracking algorithm. Steps 1 and 2 execute in serial, but operations therein are performed asynchronously. In step 1 the GPGPU tracking kernel is triggered whilst the previous marker positions are sum-reduced on the CPU to form the distribution function. These processes are synchronised by a memory transfer of the new marker positions from the device to host in step 2.

was calculated by `NUBEAM` and used in all subsequent simulations between all codes. Co-injection was also explored, but it was concluded that the larger volume of phase-space explored by counter-injected markers is vital for creating a test rigorous enough to expose discrepencies between the codes. Because all codes were forced to use an identical starting marker list from `NUBEAM`, which only provides the weight, real-space position, pitch and energy of markers, this limited the study to guiding-centre tracking. Consequently, when calculating collisional effects and wall interceptions, there is an unavoidable systematic spatial error $\sim r_{\mathrm{Larmor}}$ in the marker position introduced by differences in the finite Larmor radius (FLR) model implemented by each code:

- `TRANSP` assigns markers a random gyrophase, assuming a circular orbit
- `LOCUST` optionally tracks gyrophase, opting to ignore FLR corrections in this study

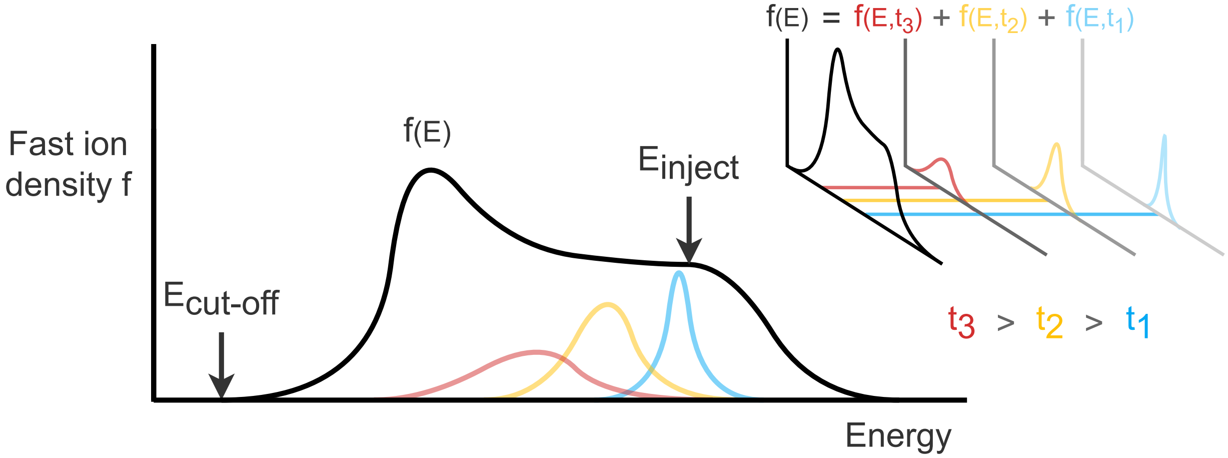

Figure 3. Numerical algorithm for generating the steady-state distribution function f, shown here in black as a single function of marker energy $f(\epsilon)$. This is formed by cumulatively summing $f(\epsilon, t_i)$, which are calculated independently at discrete time intervals t_i.

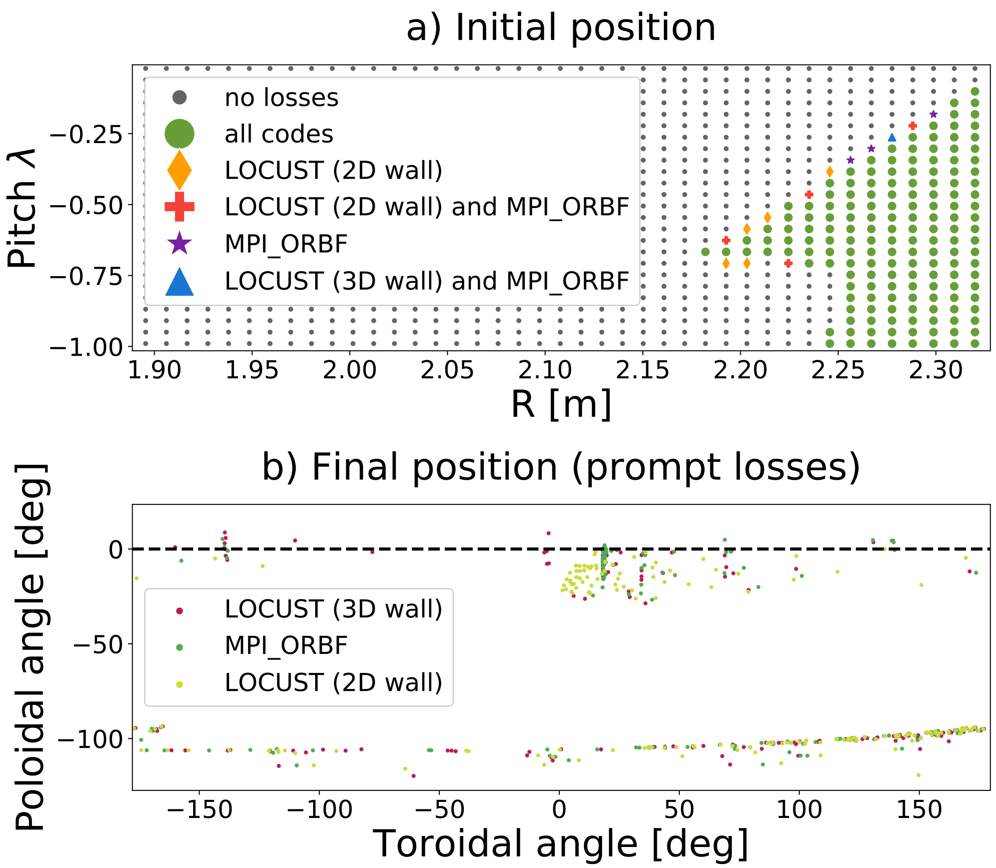

Figure 4.
a) Phase-space locations for markers initially distributed on a uniform pitch-radius grid along the outer horizontal midplane. Pitch is measured relative to the plasma current. Marker type denotes which simulations measure losses. Discrepancies are localised to the loss boundary to within dλdR.
b) Final real-space positions of all promptly lost markers. All three runs used a slightly different wall model but predict most losses at the outboard midplane limiter ($\theta_{\rm pol} = 0°$) and divertor ($\theta_{\rm pol} = -100°$).

Table 1. `Notable TRANSP namelist settings`

setting	value	comment
`GOOCON`	20	low orbit acceleration
`AVGTIM`	0	no Monte Carlo smoothing
`NZONE_FB`	40	—
`NZONE_FP`	40	—
`NZONE_NB`	200	typically 20-60 *with* smoothing
`NZONES`	200	—
`XBMBND`	1.5	previously locked to 1.3
`NPTCLS`	10^7	1.7×10^5 in `ASCOT`, 1.3×10^5 in `LOCUST`

- `ASCOT` ignores FLR corrections unless near the PFC wall, where a random gyrophase is chosen

Guiding centres were followed for 100 milliseconds and collisions disregarded for markers outside the LCFS, as is required by `NUBEAM`. Furthermore, impurities, neutral species, bulk rotation, electric fields, atomic physics and all beam-beam interactions were removed. All codes employed a lower energy cut-off at $3T_{\text{bulk}}/2$. It is worth noting that, to achieve the required fidelity without resorting to Monte Carlo smoothing, which may mask possible discrepencies, `NUBEAM` simulations for DIII-D were conducted with the untypical settings shown in table 1.

All codes produce similar results when fast-ion dynamics are isolated to within the plasma. To achieve this, an artificial axisymmetric PFC surface was created concentric to the LCFS but 5% further from the magnetic axis ($R_{\text{fac}} = 1.05$) - the closest permitted by `NUBEAM`. Figures 5 and 6 show the $[R,Z]$ and $[\epsilon,\lambda]$ projections respectively of the calculated distribution functions collected at the marker guiding centres. Subfigures (a) show the same density contours as produced by each code whilst subfigures (b) and (c) both show the absolute element-wise differences, δf and $\delta f\prime$, between the `ASCOT` and `LOCUST` distribution functions in real space. In subfigures (a) and (b) `LOCUST` aims to match the `ASCOT` collision operator, and in figures (c) `LOCUST` aims to match `NUBEAM`. In the former case the operator is fully expanded whilst in the latter ion-electron collisions were truncated to $\sim \mathcal{O}(V^5)$ and ion-ion collisions to zeroth order. The definition of $\log(\Lambda)$ was also varied. A similar comparison against `NUBEAM` could not be performed reliably, since `NUBEAM` collates the fast-ion density onto a flux-aligned grid.

In most regions, $\delta f \sim 3\%$ between all codes, which is within the fundamental uncertainties in the theoretical formulation of the Coulomb logarithm. A `NUBEAM`-like collision operator creates a lower density in the core and outboard edge but higher density towards the X-point. Within the LCFS, this leads to the average δf increasing from 3.6% to 4.7%, a change still in line with theoretical variations. Nevertheless, when the collision operator is matched, only regions near the X-point and wall retain any distinguishable difference - caused by the influence of FLR model on wall interceptions. The remaining noise in the core plasma is likely caused by differences in any tuning applied to the equilibrium field by each code, which may perturb the position of the flux surfaces and caused a flux-aligned noise pattern. Furthermore, the slight mismatch in the core is due to the adaptive time step in `ASCOT` - as described later.

The calculated flux profiles in figure 7 also show good agreement but highlight the importance of the plasma boundary. The uptick in edge electron heating is caused by the FLR corrections in `NUBEAM` spreading deposited power over a gyro-orbit width; the orbits of markers with guiding centres located just outside the LCFS concentrate their deposited power into a thin shell where the orbit overlaps the plasma. In `LOCUST` this power is instead collected at the guiding centre - outside the plasma for these markers - and is thus ignored. This effect is artificial, and simulations can avoid this by imposing ion sinks outside the plasma boundary, such as a neutral density for charge-exchange or an extrapolated plasma density.

The measured $J \times B$ torque suggests some discrepancy in orbit topology, especially at the edge [40], but this information cannot be directly extracted from `NUBEAM`. To explore this further, additional simulations were performed with similar artificial limiters up to $R_{\text{fac}} = 1.5$. This allows orbits to populate the vacuum region between the plasma and first wall. The `XBMBND` setting in `TRANSP`, which, in all DIII-D simulations previously, registered any fast ion located at $\sqrt{\psi_{\text{toroidal}}} >$ `XBMBND` as hitting a PFC, was permanently increased to avoid artificial termination of markers. Figure 8 shows the measured steady-state PFC power flux as the limiter distance is increased. As prompt losses are sensitive to wall model implementation, some disagreement between the codes is expected, especially at high R_{fac} when vacuum orbits may have a significant r_{Larmor}. Hence the agreement between `ASCOT` and `LOCUST` is satisfactory, as it is mostly within the variations expected from differences in wall model. However, it is unclear why the `NUBEAM` power fluxes diverge so quickly, though it is encouraging that the resulting discrepencies in the distribution function remain solely at the plasma edge.

The previous methodology was repeated for a spherical tokamak topology. Such devices tend to have steeper gradients [41], meaning any inaccuracies in fast-ion models will be exacerbated; in MAST for example r_{Larmor} can approach $\sim R_{\text{major}}$, and the impact on the validity of the guiding-centre approximation has long been questioned [42][43].

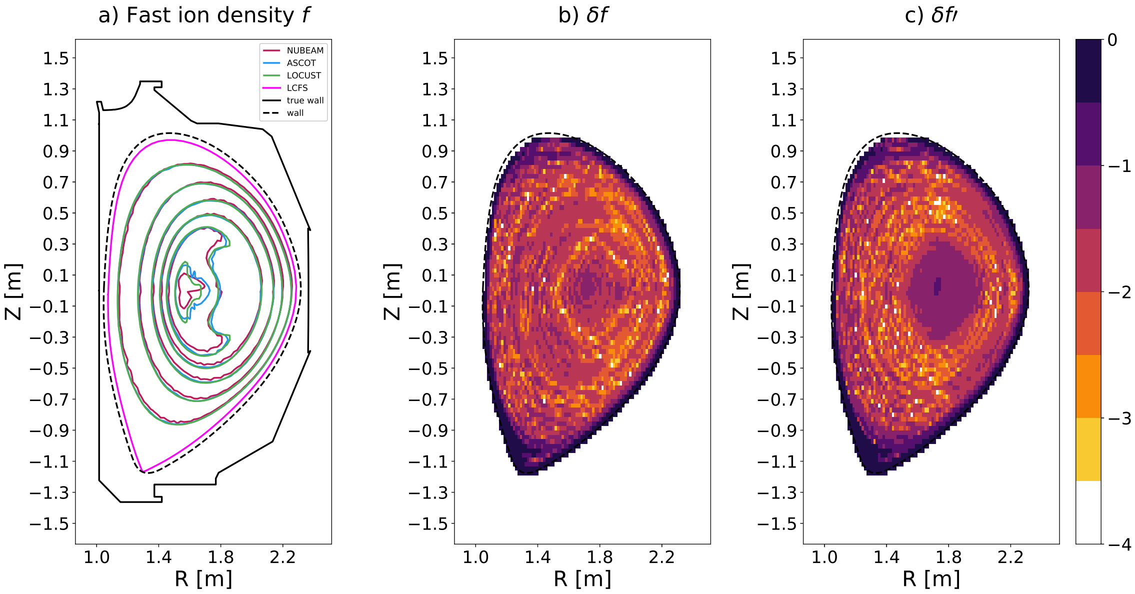

Figure 5.
a) Contours of fast-ion density integrated over velocity space. The real and artificial limiter profiles are shown in black solid and dashed respectively. `NUBEAM` bins according to a flux-aligned spiralised grid whereas `ASCOT` and `LOCUST` use rectilinear, hence some variation near the magnetic axis is expected due to numerics.
b) Absolute element-wise difference between `LOCUST` and `ASCOT` distribution functions $\delta f \equiv \log_{10}(|f_{\mathrm{LOCUST}} - f_{\mathrm{ASCOT}}|/\max(f_{\mathrm{LOCUST}}, f_{\mathrm{ASCOT}}))$. A high-order collision operator was used in `LOCUST`, as well as the `ASCOT` definition of $\log(\Lambda)$.
c) The same as plot b) except with `LOCUST` using a truncated collision operator and the `NUBEAM` definition of $\log(\Lambda)$. This choice primarily affects the core region, though some differences on the outboard side are noted. The prompt loss region near the X-point remains.

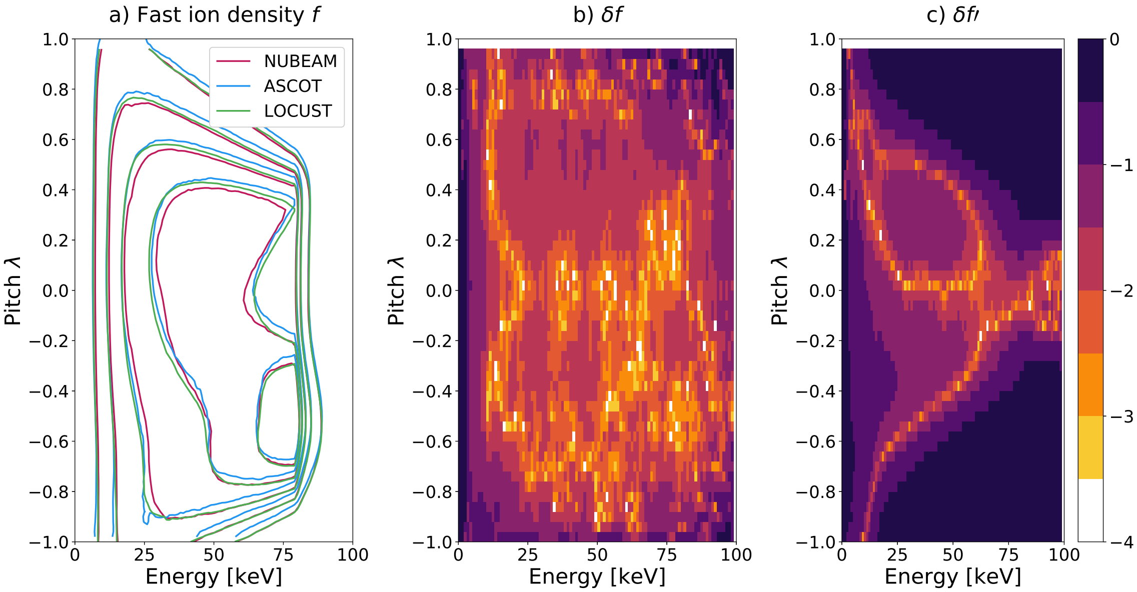

Figure 6.
a) Contours in pitch-energy space of fast-ion density integrated over real space, where λ is defined with respect to the direction of toroidal current flow as is convention in `NUBEAM`.
b) The absolute element-wise difference between the `LOCUST` and `ASCOT` distribution functions $\delta f \equiv \log_{10}(|f_{\mathrm{LOCUST}} - f_{\mathrm{ASCOT}}|/\max(f_{\mathrm{LOCUST}}, f_{\mathrm{ASCOT}}))$. A high-order collision operator was used in `LOCUST`, as well as the `ASCOT` definition of $\log(\Lambda)$.
c) The same as plot b) except with `LOCUST` using a truncated collision operator and the `NUBEAM` definition of $\log(\Lambda)$. The injection energy is 80keV, so diffusive noise can be expected above here.

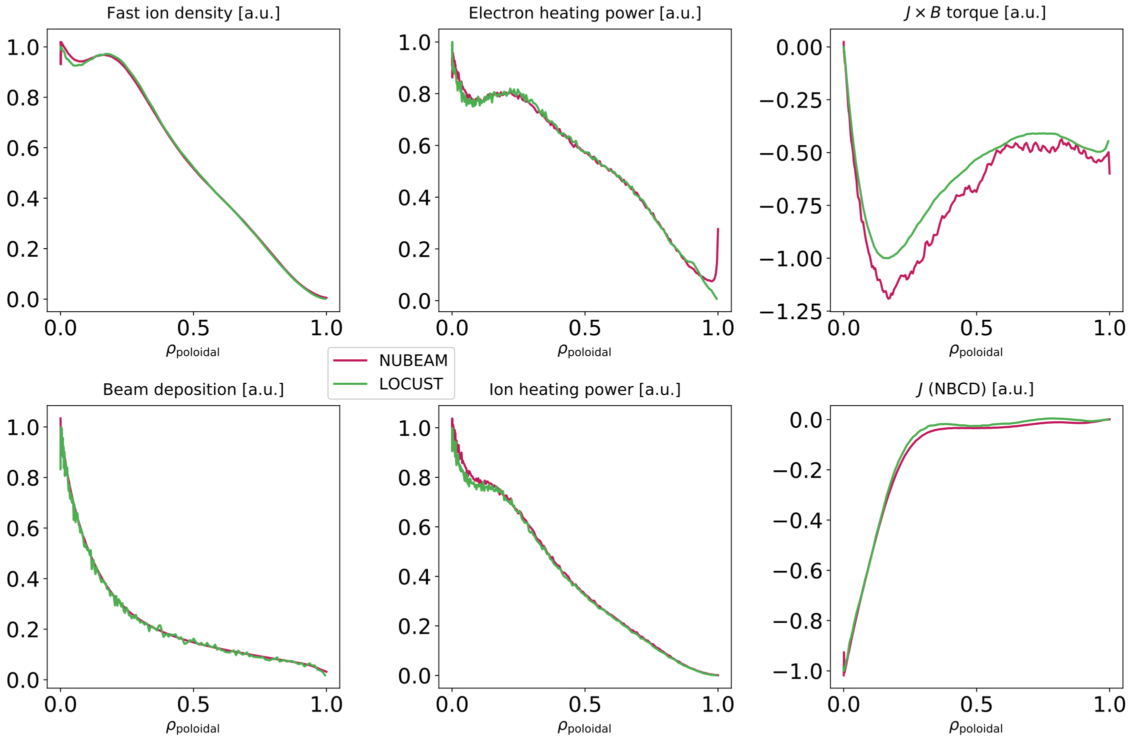

Figure 7. Normalised quantities measured against normalised poloidal flux ρ_{poloidal}. Some discrepency around $\rho = 0$ is expected due to binning width. Most important is the discrepency in the $\boldsymbol{J} \times \boldsymbol{B}$ torque, which suggests a difference in the measured orbit width.

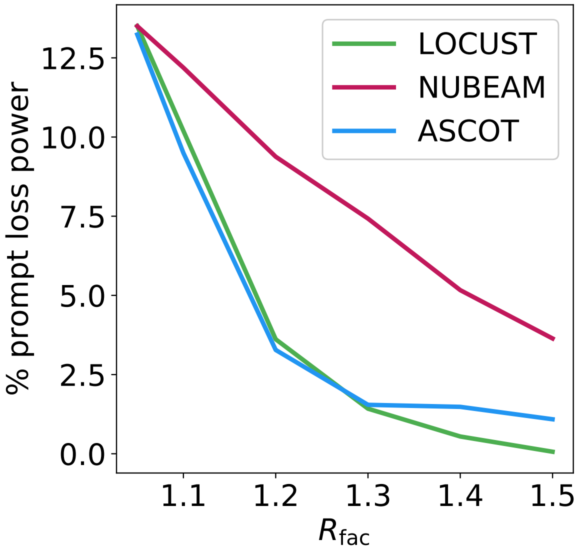

Figure 8. PFC intercept fraction as a function of limiter wall radius where $R_{\text{fac}} \equiv r_{\text{limiter}}/r_{\text{LCFS}}$ with r representing minor radius.

MAST shot #29034 was selected to allow for comparison with measured fast-ion D-α (FIDA) emission [44]. The NBI deposition code `BBNBI` was included to enable comparison of full-orbit simulations. Like before, a single time slice of data describing the background plasma was extracted at 360ms. A quiescent period was chosen during the flat-top phase when core electron temperature and density were relatively constant. To create a realistic deposition, time-resolved NBI data from `OMFIT` were used to generate the `NUBEAM` deposition for the south-south neutral beam, whilst settings as similar as possible were chosen for `BBNBI`: a 62keV co-current beam with 62% full, 27% half and 11% third energy fractions. Hence it is technically inappropriate to cross-compare the results attached to each beam code in this case. GC and FO trajectories were then calculated by each code over 100ms - enough to reach steady state.

The co-current NBI confines the fast ions to the plasma core, where discrepencies are hard to distinguish and there is a systematic shift in spatial density due to the FLR displacement, so instead we examine $f(\epsilon)$, which still encodes some real-space information through the effects of the steep temperature and density gradients on the fast-ion diffusion rate. The only unexpected discrepency in real space is caused by noise in the NUBEAM distribution function, since this case decreased NPTCLS to 10^5.

The average $f(\epsilon)$ across all simulations is shown normalised in black in figures 9 and 10. The residuals - the difference between each simulation and the unnormalised average - are also shown normalised against the average. Most simulations are within ±3% of the mean, except at lower energies due to collision operator truncation.

As a figure of merit, for simulations using a deposition calculated by BBNBI in figure 9, the maximum difference in total fast ions is 5%, falling to $\leq$ 2% for pairs of simulations which follow similar assumptions - even including the ASCOT GC simulation, which differs from the LOCUST equivalent by 1.7% (the FO equivalents differ by 0.9%). The reason for the increased density at high energies measured by ASCOT GC is likely due to the adaptive time step used. NUBEAM, which also uses orbit acceleration, shows a similar feature in figure 10. For FO and GC simulations of a homogeneous plasma by LOCUST and ASCOT, the only discrepency is an overestimation of fast-ion density of $\sim$ 20% in the high-energy diffusive tail by ASCOT GC. Whilst this effect is much more pronounced in MAST than DIII-D, the total number of fast ions in the homogeneous case still only differs by $\sim$ 0.2% - small enough to be affected by slight variations in beam deposition as shown in figure 10 where the effect is lessened.

To compare these predictions more quantitatively, calculated Kolmogorov-Smirnov (KS) statistics D [45], and their corresponding probabilities $P(D_{\text{measured}} < D)$, are shown in figure 11 for matching pairs of $f(\epsilon)$. Typically, the null hypothesis of the KS test, that the two empirical distribution functions to be compared are drawn from the same distribution function, is *rejected* if the measured KS statistic D_{measured} satisfies $P(D_{\text{measured}} < D) \leq \alpha$, with α typically chosen to be ≤ 0.05 - here we increase α to 0.1 for rigor. For calibration, figure 11a first shows the KS statistic for the GC and FO LOCUST simulations. In this case $P \approx 1$, meaning these distributions comfortably agree as expected. Contrast this to the next KS statistic, where the collision operator has been truncated, and the test clearly fails. Next we see that the LOCUST FO simulation easily agrees with the ASCOT equivalent, with $P \approx 1$ when the Coulomb logarithm is matched. The equivalent GC comparison fails solely due to the high-energy tail effect, since the equivalent measurement for the NUBEAM deposition in figure 11b now passes. Finally, we also observe that NUBEAM decisively matches LOCUST.

4.3. Synthetic diagnostics

To validate that the comparisons for MAST are realistic, the distribution functions calculated from the NUBEAM deposition by LOCUST and NUBEAM were fed into FIDASIM to generate synthetic FIDA measurements. The predicted and total measured signals for this spectral range are shown in figure 12a. Signals within a 660.5-661.5nm gate are shown radially resolved in figure 12b, along with each predicted signal from FIDASIM.

Within the core plasma, signals from LOCUST and TRANSP are within the smallest error bars of each other so as to be indistinguishable by the FIDA diagnostic. However, despite the gate, the presence of background Bremsstrahlung emission is still observed outside of $R_{\text{major}} = 1.25$m and the LCFS in figure 12b. Current error estimates do not take background light into account, or even the lack of a time-dependent background plasma, but if errors increased globally by $\sim$ 70% then the reduced $\chi^2 \sim 1$ within $R_{\text{major}} = 1.25$m.

5. Summary

The LOCUST code has been described. This includes discussions on the topics of the underlying kinetic physics model, software and hardware implementation, and algorithm design - as well as their mutual influence. In short, assumptions which allow for the independent tracking of fast-ion markers enable the adoption of massively parallel SIMT hardware - in this case GPGPUs.

LOCUST has been shown to compare well with popular fast-ion codes at a fundamental level over a range of realistic test environments. In the correct circumstances, it was shown that the predictions of all codes converge to within the assumptions of their respective physics models: in this case, the accuracy of the collision operator, choice of Coulomb logarithm and FLR model. These comparisons have been validated against experiment. In achieving this, credibility has been added to the conclusions of other parallel benchmarking activities, which may consider physics not present here [43].

Most importantly, these results demonstrate a readiness for the routine use of LOCUST in physics exploration, virtual engineering and plant design, and sophisticated integrated modelling workflows via the IMAS platform.

Moving forwards, IMAS itself will become increasingly important for the standardisation of verification and validation activities, as codes adapt to studying high-performance devices whereby problems grow increasingly sensitive. Indeed, the prediction [46] that computational tools will need to continually evolve into

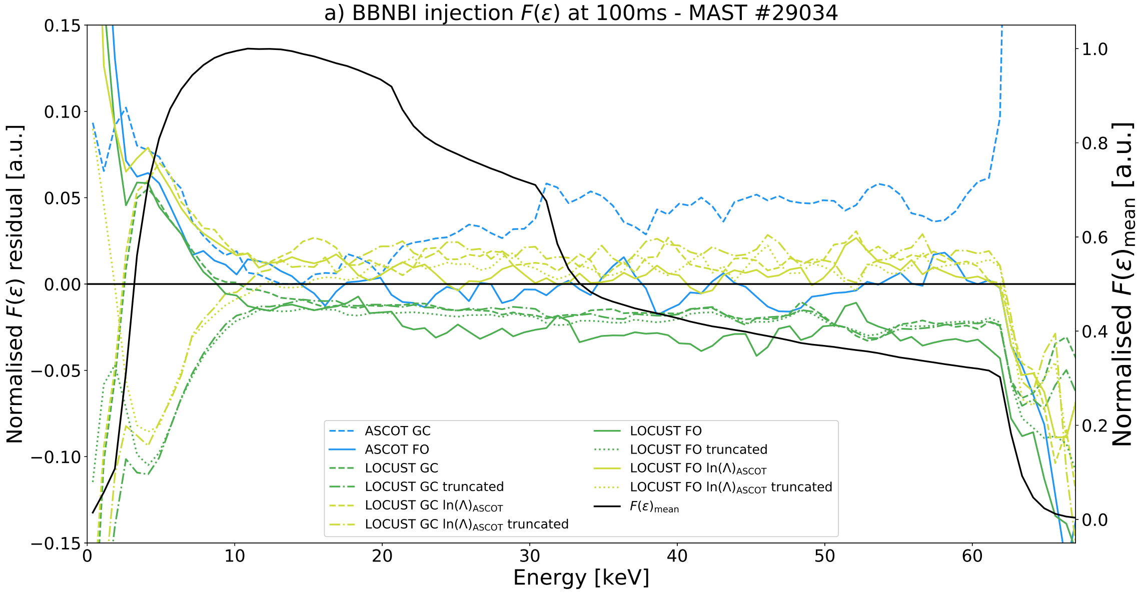

Figure 9. The fast-ion density $f(\epsilon)$ after 100 milliseconds, integrated over all dimensions except energy, is averaged across all simulations, normalised and shown in black as $f(\epsilon)_{\mathrm{mean}}$. The residual differences $(f(\epsilon) - f_{\mathrm{mean}})/f_{\mathrm{mean}}$ for each simulation are shown on the density residual axis. Simulations using GC and FO tracking (solid and dashed respectively); truncated and high-order collision operators (truncated and non-truncated labels respectively); and `LOCUST` $\log(\Lambda)$ and `ASCOT` $\log(\Lambda)$ (dark green and light green respectively), are all shown, with combinations of these linestyles representing the corresponding combinations of options. The near-symmetrical splitting of predictions at lower energies is caused by the collision operator truncation whilst systematic differences can be mainly attributed to the $\log(\Lambda)$ used.

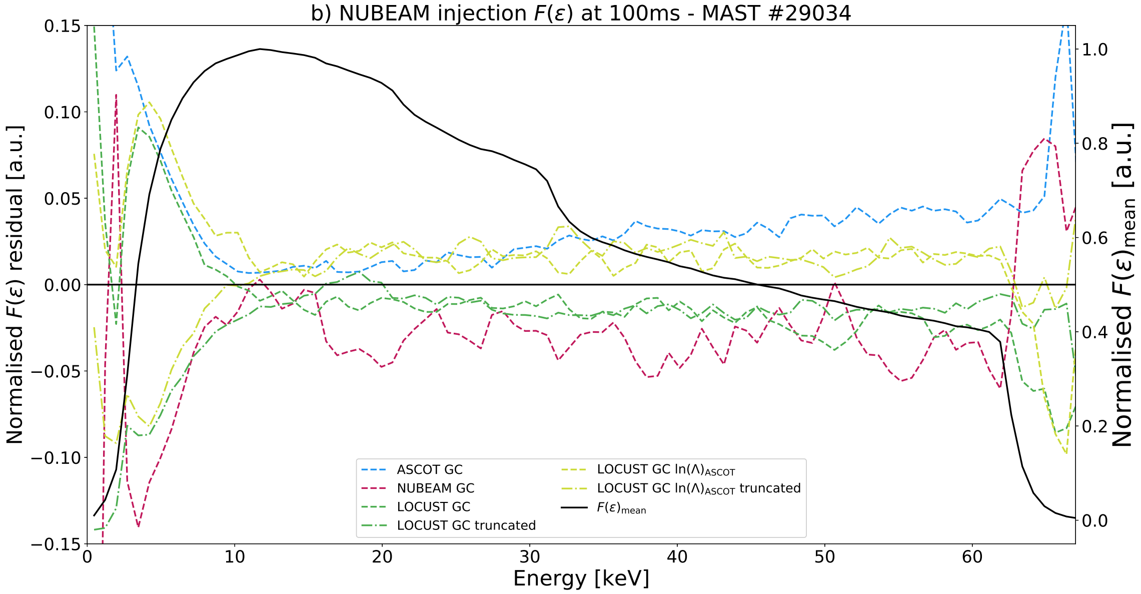

Figure 10. Equivalent to figure 9 but using deposition from `NUBEAM`, meaning only GC simulations can be performed. The overall trend is similar to figure 9 but the `ASCOT` tail is reduced.

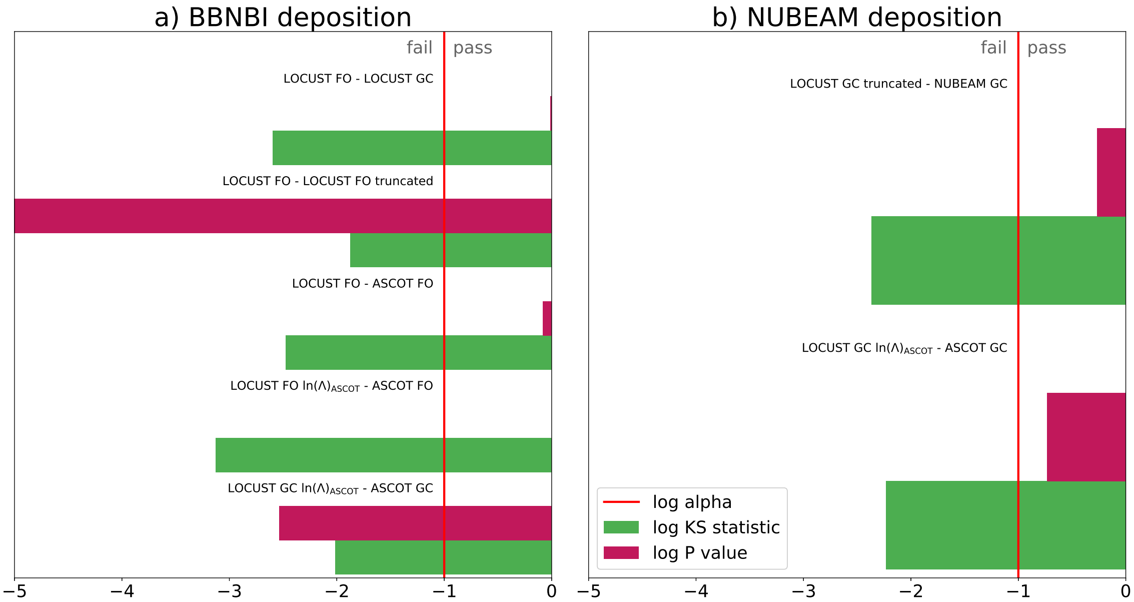

Figure 11. Kolmogorov-Smirnov test statistics (in green) for combinations of distribution functions shown in figures 9 and 10. The associated $P{=}P(D_{\mathrm{measured}} < D)$ values are shown in red, with the arbitrary test pass-fail boundary α shown vertically in bright red. To pass the test, P must be lower than α, as demonstrated by comparing GC and FO `LOCUST` simulations.

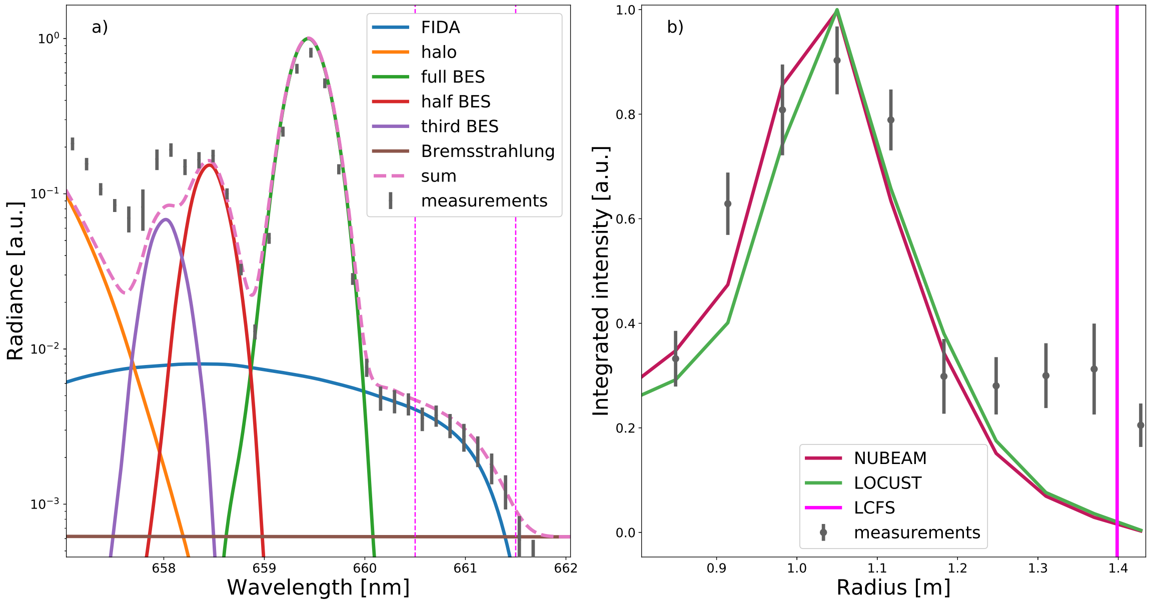

Figure 12.
a) Theoretical and measured signal intensities in MAST shot #29034 at 300ms for one channel. The wavelength filter denoted by vertical dashed lines encompasses 660.5-661.5nm and is used to integrate all signals across all channels to produce plot b). Some signals are still left out near 660nm to avoid integration of beam emission in other channels.
b) Radially resolved FIDA signal measured in MAST and as produced by `FIDASIM` for `NUBEAM` and `LOCUST`. An FLR correction was applied post-simulation.

more integrated workflows is still actively being realised [47]. As such, verification capabilities should evolve accordingly via continuous testing. Fortunately, the modelling community currently has an opportunity to enable this type of software lifecycle framework in the form of `IMAS`. With this capability, the modelling community will be empowered to quickly and confidently adapt to computationally challenging studies in the future.

6. Acknowledgements

This project was undertaken on the Viking Cluster, which is a high performance compute facility provided by the University of York. We are grateful for computational support from the University of York High Performance Computing service, Viking and the Research Computing team. Additional thanks are also owed to Stuart Henderson, David Keeling, Marina Gorelenkova, James Buchanan, Clive Michael and the team at Aalto University.

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 and 2019-2020 under grant agreement No 633053.

The views and opinions expressed herein do not necessarily reflect those of the European Commission. The views and opinions expressed herein do not necessarily reflect those of the ITER Organization.

References

[1] SD Pinches, IT Chapman, Ph W Lauber, HJC Oliver, SE Sharapov, K Shinohara, and K Tani. Energetic ions in iter plasmas. *Physics of Plasmas*, 22(2):021807, 2015.

[2] Konsta Särkimäki. Efficient and rigorous evaluation of fast particle losses in non-axisymmetric tokamak plasmas. *Nuclear Fusion*, 60(3):036002, 2020.

[3] MD Boyer, S Kaye, and K Erickson. Real-time capable modeling of neutral beam injection on nstx-u using neural networks. *Nuclear Fusion*, 59(5):056008, 2019.

[4] M Weiland, R Bilato, R Dux, B Geiger, A Lebschy, F Felici, R Fischer, D Rittich, M van Zeeland, Eurofusion MST1 Team, et al. Rabbit: Real-time simulation of the nbi fast-ion distribution. *Nuclear Fusion*, 58(8):082032, 2018.

[5] Sara Hooker. The hardware lottery. *arXiv preprint arXiv:2009.06489*, 2020.

[6] Andreas Klöckner, Nicolas Pinto, Yunsup Lee, B. Catanzaro, Paul Ivanov, and Ahmed Fasih. PyCUDA and PyOpenCL: A Scripting-Based Approach to GPU Run-Time Code Generation. *Parallel Computing*, 38(3):157–174, 2012.

[7] Siu Kwan Lam, Antoine Pitrou, and Stanley Seibert. Numba: A llvm-based python jit compiler. In *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, pages 1–6, 2015.

[8] TJ Martin. Computer programs for electromagnetic field and charged particle calculations, 1973.

[9] Gian Luca Delzanno et al. On particle movers in cylindrical geometry for particle-in-cell simulations. *Journal of Computational Physics*, 253:259–277, 2013.

[10] Krasymyr Tretiak and Daniel Ruprecht. An arbitrary order time-stepping algorithm for tracking particles in inhomogeneous magnetic fields. *Journal of Computational Physics: X*, 4:100036, 2019.

[11] Robert G Littlejohn. Variational principles of guiding centre motion. *Journal of Plasma Physics*, 29(1):111–125, 1983.

[12] Erwin Fehlberg. *Low-order classical Runge-Kutta formulas with stepsize control and their application to some heat transfer problems.* National aeronautics and space administration, 1969.

[13] Jeff R Cash and Alan H Karp. A variable order runge-kutta method for initial value problems with rapidly varying right-hand sides. *ACM Transactions on Mathematical Software (TOMS)*, 16(3):201–222, 1990.

[14] John R Dormand and Peter J Prince. A family of embedded runge-kutta formulae. *Journal of computational and applied mathematics*, 6(1):19–26, 1980.

[15] David Goeken and Olin Johnson. Fifth-order runge-kutta with higher order derivative approximations. *Electronic Journal of Differential Equations*, 2(November):1–9, 1999.

[16] J. P. Boris and R. A. Shanny. *Proceedings: Fourth Conference on Numerical Simulation of Plasmas, November 2, 3, 1970.* Naval Research Laboratory, 1972.

[17] E. Hirvijoki et al. Ascot: Solving the kinetic equation of minority particle species in tokamak plasmas. *Computer Physics Communications*, 185(4):1310–1321, 2014.

[18] A. Boozer et al. Monte carlo evaluation of transport coefficients. *The Physics of Fluids*, 24(5):851, 1981.

[19] A. Galeev and R.N. Sudan. *Basic plasma physics.* Number v. 1 in Handbook of plasma physics. North-Holland Pub., 1983.

[20] Subrahmanyan Chandrasekhar. Stochastic problems in physics and astronomy. *Reviews of modern physics*, 15(1):1, 1943.

[21] Alexei Pankin, Douglas McCune, Robert Andre, Glenn Bateman, and Arnold Kritz. The tokamak monte carlo fast ion module nubeam in the national transport code collaboration library. *Computer Physics Communications*, 159(3):157–184, 2004.

[22] K McGuire, R Goldston, M Bell, M Bitter, K Bol, K Brau, D Buchenauer, T Crowley, S Davis, F Dylla, et al. Study of high-beta magnetohydrodynamic modes and fast-ion losses in pdx. *Physical review letters*, 50(12):891, 1983.

[23] Todd A Wareing, John M McGhee, and Jim E Morel. Attila. a 3-d unstructured tetrahedral-mesh s n code, 1997.

[24] Hinrich Lütjens, Anders Bondeson, and Olivier Sauter. The chease code for toroidal mhd equilibria. *Computer physics communications*, 97(3):219–260, 1996.

[25] LL Lao, H St John, RD Stambaugh, AG Kellman, and W Pfeiffer. Reconstruction of current profile parameters and plasma shapes in tokamaks. *Nuclear fusion*, 25(11):1611, 1985.

[26] GTA Huysmans and O Czarny. Mhd stability in x-point geometry: simulation of elms. *Nuclear fusion*, 47(7):659, 2007.

[27] YQ Liu, Anders Bondeson, Carl-Magnus Fransson, Bengt Lennartson, and Claes Breitholtz. Feedback stabilization of nonaxisymmetric resistive wall modes in tokamaks. i. electromagnetic model. *Physics of Plasmas*, 7(9):3681–3690, 2000.

[28] Carl-Magnus Fransson, Bengt Lennartson, Claes Breitholtz, A Bondeson, and YQ Liu. Feedback stabilization of nonaxisymmetric resistive wall modes in tokamaks. ii. control analysis. *Physics of Plasmas*, 7(10):4143–4151,

2000.

[29] SC Jardin et al. Multiple timescale calculations of sawteeth and other global macroscopic dynamics of tokamak plasmas. *Computational Science & Discovery*, 5(1):014002, 2012.

[30] Otto Asunta, J Govenius, Robert Budny, M Gorelenkova, Giovanni Tardini, Taina Kurki-Suonio, Antti Salmi, Seppo Sipilä, et al. Modelling neutral beams in fusion devices: Beamlet-based model for fast particle simulations. *Computer Physics Communications*, 188:33–46, 2015.

[31] M Schneider, L-G Eriksson, I Jenkins, JF Artaud, V Basiuk, F Imbeaux, T Oikawa, JET-EFDA contributors, et al. Simulation of the neutral beam deposition within integrated tokamak modelling frameworks. *Nuclear Fusion*, 51(6):063019, 2011.

[32] Benedikt Geiger, Luke Stagner, William W Heidbrink, Ralph Dux, Rainer Fischer, Yutaka Fujiwara, AV Garcia, Asger Schou Jacobsen, A Jansen van Vuuren, Alexander N Karpushov, et al. Progress in modelling fast-ion d-alpha spectra and neutral particle analyzer fluxes using fidasim. *Plasma Physics and Controlled Fusion*, 62(10):105008, 2020.

[33] Franck Assous, Pierre Degond, and Jacques Segre. A particle-tracking method for 3d electromagnetic pic codes on unstructured meshes. *Computer physics communications*, 72(2-3):105–114, 1992.

[34] Makoto Matsumoto and Takuji Nishimura. Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 8(1):3–30, 1998.

[35] MA Van Zeeland et al. Modulation of prompt fast-ion loss by applied n= 2 fields in the diii-d tokamak. *Plasma Physics and Controlled Fusion*, 56(1):015009, 2013.

[36] Jari Varje, Konsta Särkimäki, Joona Kontula, Patrik Ollus, Taina Kurki-Suonio, Antti Snicker, Eero Hirvijoki, and Simppa Äkäslompolo. High-performance orbit-following code ascot5 for monte carlo simulations in fusion plasmas, 2019.

[37] Joshua Breslau, Marina Gorelenkova, Francesca Poli, Jai Sachdev, Xingqiu Yuan, and USDOE Office of Science. Transp, 6 2018.

[38] O. Meneghini, S.P. Smith, L.L. Lao, O. Izacard, Q. Ren, J.M. Park, J. Candy, Z. Wang, C.J. Luna, V.A. Izzo, B.A. Grierson, P.B. Snyder, C. Holland, J. Penna, G. Lu, P. Raum, A. McCubbin, D.M. Orlov, E.A. Belli, N.M. Ferraro, R. Prater, T.H. Osborne, A.D. Turnbull, and G.M. Staebler. Integrated modeling applications for tokamak experiments with omfit. *Nuclear Fusion*, 55(8):083008, 2015.

[39] Michael A Van Zeeland et al. Fast ion transport during applied 3d magnetic perturbations on diii-d. *Nuclear Fusion*, 55(7):073028, 2015.

[40] P Helander, RJ Akers, and L-G Eriksson. On neutral-beam injection counter to the plasma current. *Physics of plasmas*, 12(11):112503, 2005.

[41] JR Harrison, RJ Akers, SY Allan, JS Allcock, JO Allen, L Appel, Marian Barnes, N Ben Ayed, W Boeglin, C Bowman, et al. Overview of new mast physics in anticipation of first results from mast upgrade. *Nuclear Fusion*, 59(11):112011, 2019.

[42] CA Michael, N Conway, B Crowley, O Jones, WW Heidbrink, S Pinches, E Braeken, R Akers, C Challis, M Turnyanskiy, et al. Dual view fida measurements on mast. *Plasma Physics and Controlled Fusion*, 55(9):095007, 2013.

[43] Andrea Sperduti, Marco Cecconello, Sean Conroy, and Antti Snicker. Neutron rate estimates in mast based on gyro-orbit modelling of fast ions. *Nuclear Fusion*, 2020.

[44] Owen Jones. *Experimental fast-ion transport studies on the Mega-Amp Spherical Tokamak*. PhD thesis, Durham University, 2015.

[45] Andrey Kolmogorov. Sulla determinazione empirica di una lgge di distribuzione. *Inst. Ital. Attuari, Giorn.*, 4:83–91, 1933.

[46] Alain Becoulet, Per Strand, H Wilson, M Romanelli, L-G Eriksson, et al. The way towards thermonuclear fusion simulators. *Computer physics communications*, 177(1-2):55–59, 2007.

[47] P. Rodriguez-Fernandez, N. T. Howard, M. J. Greenwald, A. J. Creely, J. W. Hughes, J. C. Wright, C. Holland, Y. Lin, and F. Sciortino. Predictions of core plasma performance for the sparc tokamak. *Journal of Plasma Physics*, 86(5):865860503, 2020.