

Draft 0

Definition for RF solver component

1. Functionality

The minimal functionality for the RF solver component is to:

Provide the RF source terms to all other modeling components. Most generally this is the velocity-space (quasi-linear) diffusion operator induced by the RF, $\tilde{\mathbf{Q}}(\mathbf{x}, \mathbf{v})$, but it also includes the local power deposition, $P_j(\mathbf{x})$ and may include other macroscopic plasma responses such as local RF induced fluid force $\mathbf{F}(\mathbf{x})$. Outside of the RF component the wave electric field itself, $\mathbf{E}(\mathbf{x})$, is only needed for synthetic diagnostics and possibly for computation of the quasilinear operator, if this is to be done outside the RF component.

In order to provide this information any RF component will have to:

Provide an antenna model – That is it must provide an adequate representation of the spectrum of waves excited in the plasma by the launching structure

Calculate the propagation of the waves through the plasma and their absorption

Relate the wave solution $\mathbf{E}(\mathbf{x})$ to the quasilinear operator and various averages or moments thereof

1.A Equations

RF processes in fusion plasmas are governed by the Maxwell-Boltzmann system of equations. The plasma state is described by a distribution function $f_s(\mathbf{r}, \mathbf{v}, t)$ representing the density of species s in a six dimensional phase space of position and velocity. This function evolves in time according to the Boltzmann equation by convection in the 6-D phase space while under forces exerted by the electric and magnetic fields, \mathbf{E} and \mathbf{B} , respectively. For ICRF applications, the wave time scale is the fastest time scale in the system. Thus, the wave fields and particle distribution function can be separated into a time-average, or slowly varying part, $(\mathbf{E}_0, \mathbf{B}_0, f_s^0)$, and a time harmonic, or rapidly oscillating part, $[\mathbf{E}(\mathbf{r})e^{-i\omega t}, \mathbf{B}(\mathbf{r})e^{-i\omega t}, f_s^1(\mathbf{r}, \mathbf{v})e^{-i\omega t}]$ where ω is the frequency of the wave. The time-harmonic wave fields are small compared to the equilibrium fields, and the equations can be linearized with respect to these amplitudes. Solving the linearized Boltzmann equation gives the rapidly varying part of the distribution function $f_s^1(\mathbf{r}, \mathbf{v})$ in terms of the equilibrium part f_s^0 .

For the rapidly oscillating, time harmonic wave fields, Maxwell's equations reduce to a generalization of the Helmholtz wave equation,

$$-\nabla \times \nabla \times \mathbf{E} + \frac{\omega^2}{c^2} \left(\mathbf{E} + \frac{i}{\omega \epsilon_0} \mathbf{J}_p \right) = -i\omega \mu_0 \mathbf{J}_{ant}, \quad (1)$$

where \mathbf{J}_{ant} is an externally driven antenna current, localized near the plasma edge, that acts as a source for the waves. The fluctuating plasma current \mathbf{J}_p can be derived directly from the rapidly varying part of the distribution function $f_s^1(\mathbf{r}, \mathbf{v})$. In general, \mathbf{J}_p is a non-local, integral operator on the wave electric field,

$$\mathbf{J}_p(\mathbf{r}, t) = \sum_s \int d\mathbf{r}' \int_{-\infty}^t dt' \sigma(f_s^0(E), \mathbf{r}, \mathbf{r}', t, t') \mathbf{E}(\mathbf{r}', t'), \quad (2)$$

where $\sigma(f_s^0, \mathbf{r}, \mathbf{r}', t, t')$ is the “plasma conductivity kernel.”

Several approximations are necessary to obtain practical solutions. These approximations are code specific. Probably most significant is the necessity to approximate the non-local conductivity kernel. Common approximate forms in use include the Stix model, the Finite Larmor radius (FLR) model, and the cold plasma model.

Various approximations are also used in specifying the antenna current distribution, $\mathbf{J}_{ant}(\mathbf{x})$. So far we are not able to calculate the self-consistent current distribution, including the response to the plasma fields and all of the 3D structure of the antenna itself. Sometimes the wave excitation is modeled by a non-zero equivalent field at the conducting wall calculated by a separate 3D antenna model.

A very important approximate solution method involves geometric optics (ray tracing), which does not treat the wave field at all but calculates the trajectories of power flow by a Hamiltonian method and calculates power absorption along the trajectory by various models. In this case the antenna model is replaced by an initial distribution of rays originating in the plasma near the antenna.

The response of the plasma distribution function, $f_0(\mathbf{x}, \mathbf{v}, t)$, on timescales slower than the RF period is obtained from a time-averaged form of Eq (1), referred to as the quasilinear Fokker-Planck equation,

$$\frac{df_0}{dt} = Q(\mathbf{E}^1, f_0) + C(f_0) + S(\mathbf{x}, \mathbf{v}) \quad (3)$$

Here, the time derivative includes particle drift motion and, the RF quasilinear operator, $Q(\mathbf{E}^1, f_0)$ [23], is quadratic in the RF field $\mathbf{E}(\mathbf{x}, t)$ and describes wave-induced velocity-space diffusion of f_0 . In the absence of sources, the collision operator relaxes f_0 to a local Maxwellian distribution. However, quasilinear diffusion, particle or energy sources, or gradients in macroscopic plasma quantities drive f_0 away from Maxwellian.

The general form of the divergence of the quasilinear flux is

$$\nabla \cdot \bar{\mathbf{Q}} \cdot \nabla_{\mathbf{v}} f_0 = \left\langle \left(\mathbf{E}_1^* + \mathbf{v} \times \mathbf{B}_1^* \right) \cdot \nabla_{\mathbf{v}} f_1 \right\rangle_t$$

where the average is over a time long compared to a gyro-period or wave period but short compared to the time scale for evolution of the distribution function. This is always further averaged – bounce averaged, orbit averaged or flux surface averaged.

To obtain self-consistency between the wave fields and the particle distribution function, a procedure of iteration of Eqs. (1-3) is necessary. In this procedure, four different physical

models are iteratively coupled: (1) the plasma conductivity for non-Maxwellian distribution functions, (2) a wave solver incorporating this non-Maxwellian conductivity, (3) the quasi-linear operator that drives the non-thermal distribution, and (4) a Fokker-Planck solver to advance the distribution function. Solution of the Fokker-Planck equation is not considered part of the RF component, but is itself a separate component. For the Fast MHD campaign a separate iteration between the RF component and the distribution function solver component will likely be necessary to maintain self-consistency.

Following Stix [4], the normalized plasma conductivity (i.e. susceptibility) for an arbitrary non-relativistic species s can be expressed as a velocity space integral of the perturbed distribution function over velocity components perpendicular and parallel to the magnetic field. The integral over parallel velocities is singular at cyclotron resonance, and is therefore evaluated using the Plemelj relation.

The power absorbed by the plasma P_{RF} is quadratic in the wave electric field and can be expressed locally as a double summation over Fourier wave numbers \mathbf{k}_1 and \mathbf{k}_2 [5]. This sum can be extremely time-consuming to evaluate, requiring four nested do loops in 2-D and six nested loops in 3-D. Even for Maxwellians, these summations takes an order of magnitude more time to calculated than the wave solution itself. For non-Maxwellians, the time is totally prohibitive. A more efficient way to calculate the power absorption emerges when the velocity space integrals are brought outside of the sum over \mathbf{k}_1 and \mathbf{k}_2 . In this case, P_{RF} can be expressed as a product of sums rather than as nested sums. Because the sums over \mathbf{k}_1 and \mathbf{k}_2 are separated, there is an enormous savings in computation time, and P_{RF} can be evaluated for non-Maxwellians in approximately the same time as required to calculate the plasma current.

The flux surface average heating rate can be written in terms of the bounce-averaged distribution function f_0 and quasi-linear diffusion coefficients. The power absorption obtained in this way is compared to that obtained from the flux surface average of P_{RF} to provide a cross-check on the consistency of the quasi-linear diffusion coefficients calculated by the wave solver.

1.B Codes for initial implementation of the component

AORSA2D

The AORSA code [1] solves general integral form of Eqs.(1-2), with no restriction on wavelength relative to orbit size and no limit on the number of cyclotron harmonics. In addition, AORSA has been generalized to treat non-thermal (i.e. non-Maxwellian) plasma components. However there is a ‘local’ approximation in the characteristic integral describing the parallel unperturbed motion of the particle, yielding conventional plasma dispersion functions for the parallel response. Thus the conductivity kernel is effectively the Stix model with full Bessel functions and non-Maxwellian distributions.

Fourier modes in cylindrical coordinates, (R, Z, φ), the major radius, vertical height, and toroidal angle, respectively, are used as basis functions for solutions in 1D, 2D, or 3D.

$$\mathbf{E}(\mathbf{x}) = \sum_{n,m,l} \mathbf{E}_{n,m,l} e^{i(nx+my+l\phi)} \quad (4)$$

The method of collocation is used to write the Fourier expanded wave equation at each point on a multi-dimensional spatial grid, and because of the integral nature of the plasma response, are dense. The resulting set of linear equations is then transformed from Fourier space back to configuration space and solved for the components of the RF wave field using ScaLAPACK libraries with parallelism implemented through MPI. This method avoids complicated convolutions associated with calculating the plasma current, and at the same time, includes cyclotron harmonics of arbitrarily high order. The fast Fourier transformation is accomplished by using the publicly available FFTPACK, developed at NCAR

For Maxwellian velocity distributions, the plasma conductivity can be expressed in terms of analytic functions, while for non-Maxwellians, a 2D numerical integral must be performed at each collocation point for each Fourier mode. The parallel ScaLAPACK library is used to factor this matrix. Depending on the dimensionality and resolution, there are from 10^3 to $<10^6$ complex unknowns. The electric field solution is then used to calculate the 4D (for 2D spatial resolution) RF quasilinear operators.

(a) Programming languages, libraries, and other software used:

The all orders spectral algorithm AORSA [1] is a Fortran-based parallel processing code developed under SciDAC at the Oak Ridge National Laboratory. Fixed source Fortran 77 style code shares space with newer Fortran 90 constructs. In particular, some routines are defined within MODULES, memory is dynamically managed using ALLOCATE, and KIND adds flexibility for data typing, etc. A modern build system manages ports to several high performance computing platforms.

The code base consists of approximately 28,000 lines of executable instructions, 5,500 data declarations, and 8,000 comment lines, contained in about 45 files. The compiled executable ranges in size from 1.3 Mbytes on Cheetah (AIX, power4, xlf compilers), to 6.1 Mbytes on Ram (Linux, IA64, Intel compilers), to 14.7 Mbytes on Jaguar (Linux, AMD Opteron, Cray programming environment). This range is a function of, among other things, operating system requirements, the optimization strategies and resources employed by the compiler, and the use of dynamically loaded shared object libraries. (Cray provides statically linked libraries, which explains the larger compiled executables.)

(b) Parallel programming system used (e.g. MPI, OpenMP, ...)

Communication among parallel processes is centered around the solution of a double precision complex-valued system of linear equations. Each grid point generates three complex linear equations. However, only those grid points within the real physical space (as opposed to Fourier space) are included in the calculation. This reduces the size of the system by about 30%. Thus, for an $N \times N$ grid in 2-D, AORSA generates a matrix of size approximately $0.70 \cdot (3 \cdot N^2)$. For example, a medium size ITER problem (128x128) requires the solution of a double complex valued linear system of size 34,692. The larger ITER

problem (256x256) required to resolve the mode-converted waves requires solution of a linear system of size 124,587.

The linear system is solved using the publicly available ScaLAPACK library. In particular, the routine `pzgetrf` factors the matrix into upper and lower components, which are then used by the routine `pzgetrs` to compute the solution vector. ScaLAPACK also contains the parallel BLAS library, which includes two routines called directly by AORSA: `pzsca1` for scaling distributed vectors, and `pdznrn2` for computing distributed vector norms. The PBLAS library also provides a convenient mode for distributing the matrix into the required two-dimensional block cyclic format. Inter-process communication within ScaLAPACK is managed using the BLACS component library, which is also used by AORSA for distributing the solution vector within the AORSA code, as well as setting process barriers. The ScaLAPACK library is included as a highly optimized component of most high performance computing environments.

Parallelism external to the linear system solver requires little inter-process communication. The parallel processes operate on assigned sections of the problem domain, sharing their results in only a few places via BLACS reduction functionality.

Although AORSA uses the MPI parallel programming model, MPI functionality is not directly exposed. Thus ports to other present and future architectures that provide inter-process communication protocols is simply dependent upon the provision of a BLACS library, or with only slightly more work, a linear equation solver.

(c) Efficiency of the calculation:

AORSA demonstrates excellent strong scaling as well as very good computational performance on each processor on the XT3 at ORNL. For a small problem that includes the effect of non-Maxwellian distribution functions (128x128), the runtime is 79.8 minutes on 64 processors, 44.3 minutes on 128 processors, and 24.4 minutes on 256 processors. For the large Maxwellian problem required to study mode conversion in ITER (256x256), the runtime is 44.4 minutes on 1024 processors, 27.1 minutes on 2048 processors, and 23.3 minutes on 3072 processors. These times are expected to be meaningfully reduced as Cray focuses more effort on improving the performance of ScaLAPACK on the XT3. This process will be strongly affected by Cray's plans to reduce inter-process communication latencies and collect communication.

For the ITER mode conversion problem, the bulk of computation in AORSA is in the solution of the linear system. For example, the linear solution represents about 68% of the total computation time for a 1024 processor run. The efficiency of this part of the calculation on the XT3 is about 50% and expected to increase as Cray optimizes ScaLAPACK for the XT3. For an N-dimension system, the number of floating point additions and multiplications is $8*N^3 + 7*N^2$ for the linear solution. This represents significant use of the N^2 data, and explains the relatively high efficiency for this phase of the calculation.

(d) Data storage and transfer requirements

In addition to the small problem definition input file aorsa2d.in, AORSA reads in three data files: ZTABLE.TXT, cql3d.out, and eqdsk.out. For the 128x128 problem, the file sizes are listed in the following table:

<i>File name</i>	<i>Description</i>	<i>Size (Mbytes)</i>
cql3d.out	CQL3D output	29.3
eqdsk.out	Magnetic field geometry	1.6
ZTABLE.TXT	Z function table	10.6

Output is also fairly small, and all written by process 0.

<i>File name</i>	<i>Description</i>	<i>Size (Mbytes)</i>
movie_[ealpha,eb,wdot]	Movie output	.39-.79
out15	Printed output	.13
out38	Plotting output	14.0
out_cql3d.coef	CQL3D input	39.0

- [1] Jaeger, E. F., Berry, L. A., Myra, J. R., et al., *Phys. Rev. Lett.* **90**, 195001-1 (2003).
- [2] Kennel, C. F. and Engelmann, F., *Phys. Fluids* **9**, 2377 (1966).
- [3] Harvey, R. W., and McCoy, M. G., in *Proceedings of the IAEA Technical Committee Meeting on Advances in Simulation and Modeling of Thermonuclear Plasmas* (IAEA, Montreal, 1992), available through USDOC, NTIS No. DE9300962.
- [4] Stix, T. H., *Waves in Plasmas*, American Institute of Physics, New York, 1992.
- [5] Smithe, D. N., *Plasma Phys. Controlled Fusion* **31**, 1105 (1989).

TORIC

The perpendicular plasma conductivity in the 2D (axisymmetric tokamak geometry) TORIC [6,7] kernel is a Taylor expansion in the perpendicular wavelength and includes terms through second order in the (assumed) small parameter of perpendicular wavelength times the ion cyclotron radius. Flux coordinates are used to express the radial dependence in terms of a single variable with a finite-element basis set. Fourier modes are used to expand the dependence on poloidal angle. This poloidal dependence allows an algebraic expression for the plasma conductivity parallel to the equilibrium magnetic field. This formulation of the plasma conductivity describes interactions through the second cyclotron harmonic. ICRH at the first and second harmonic are typical applications, with lower hybrid applications possible at for very high resolution. The resulting linear equations are block tridiagonal. For

typical resolutions, both the block size and number of blocks are from a few hundred to several thousand. This matrix is factored using a parallel (or serial for small problems) algorithm that saves the LU factorization of each block to disk, and then generates a solution from this decomposition using a variation of the Thomas algorithm.

[6] J. C. Wright, P. T. Bonoli, M. Brambilla et al., Phys. Plasmas **11**, 2473 (2004).

[7] M. Brambilla and T. Krucken, Nucl. Fusion **28**, 1813 (1988).

GENRAY

GENRAY [8] is a general ray tracing code for the calculation of electromagnetic wave propagation and absorption in the geometrical optics approximation. It provides a solution of the ray tracing equations in general non-axisymmetric geometry, although work to date is with axisymmetric equilibria with added toroidal perturbations. Several alternative dispersion functions, D , are provided in order to ray trace for Electron Cyclotron (EC), Lower Hybrid (LH), and Ion Cyclotron Range of Frequencies (ICRF) waves. Current drive is calculated based on Maxwellian distribution functions. Results are coupled to the CQL3D code to provide input for calculation of the RF quasi-linear diffusion coefficients. The code requires solving six ordinary differential equations and will not pose a significant load on computer resources.

2. Interface

2.A Inputs required for all RF solvers

f_{rf} = wave frequency

$n_j(\mathbf{x})$ = density of all species j treated as Maxwellian, generally this presupposes knowledge of the flux surfaces of the magnetic equilibrium e.g. $\Psi(\mathbf{x})$

$T_j(\mathbf{x})$ = temperature of all species j treated as Maxwellian

$f_j(\mathbf{x}, \mathbf{v})$ = distribution function of all species j treated as non-Maxwellian

Selectors for different plasma models for each species – e.g cold, FLR, full Bessel, Maxwellian/non-Maxwellian

$\mathbf{B}(\mathbf{x})$ = Magnetic field vector

$\nabla n_j(\mathbf{x})$ = gradient density of all species j treated as Maxwellian

$\nabla T_j(\mathbf{x})$ = gradient temperature of all species j treated as Maxwellian

$\nabla f_j(\mathbf{x}, \mathbf{v})$ = gradient distribution function of all species j treated as non-Maxwellian

$\nabla \mathbf{B}(\mathbf{x})$ = gradient Magnetic field tensor

Antenna model (includes specification of total launched power)

Conducting wall geometry description \Leftrightarrow boundary conditions

2.B Outputs required from all RF solvers

$\bar{\mathbf{Q}}(\mathbf{E}, \mathbf{x}, \mathbf{v})$ = quasilinear operator or bounce/orbit averages of the operator

Moments and averages of this are also useful.

$P_j(\mathbf{x})$ = local power deposition density

$\langle P \rangle$ = flux surface averaged power deposition (Ψ)

$\langle\langle P \rangle\rangle$ = bounce averaged or orbit averaged power in a given velocity space element $(\varepsilon, \mu, \rho_0)$

$\mathbf{F}(\mathbf{x})$ = local RF force

$\langle j_{\parallel RF} \rangle$ = flux surface averaged driven current (Ψ)

2.C Interfaces provided

Initialize – read input files, set code specific parameters

Run – Solve RF fields. Return E, quasilinear operator, power deposition, etc

Save internal state for restart

Finalize – clean up data structures, flush communication pipes, etc

2.D Interfaces used – coupling to Plasma State component, direct coupling to other components

Plasma state – Require all of the inputs listed in 2.A

Possible direct connection to Fokker Planck solver to exchange $f_j(\mathbf{x}, \mathbf{v})$ and $\bar{\mathbf{Q}}(\mathbf{E}, \mathbf{x}, \mathbf{v})$ or $\mathbf{E}(\mathbf{x})$ for internal iteration to self-consistency.

3. Code specific data

AORSA2D

a) Specific inputs: grids, code specific controls, and parameters

Control parameters → output options, solver options

Mesh description

b) Code state data that is required to re restart:

I believe (DBB) the only internal state that might need to be saved for this component is associated with sub-iterations to convergence RF fields with Fokker Planck solutions – $f_j(\mathbf{x}, \mathbf{v})$ and $\bar{\mathbf{Q}}(\mathbf{E}, \mathbf{x}, \mathbf{v})$ or $\mathbf{E}(\mathbf{x})$

c) What type of spatial discretization is used for each field? (finite diff, finite element, spectral, finite volume, ...)? What are the dimensionalities (1D, 2D, 3D)? What are the coordinates (Cartesian, flux ...)? How is the poloidal angle defined? Are mappings applied to restructure the domain (e.g., to be able to use a regular mesh or a Collela-Burger AMR mesh)?

d) Other code outputs (that might be useful for diagnostics or visualization for example) that doesn't specifically require inter-component interaction

e) Where the required inputs (both generic and code specific) are going to come from: Plasma State or other components, external file, ... (Any inputs that are strictly local to the component code can and should be input through the code's current mechanisms e.g. file I/O usually).

Most of the generic inputs will come from the Plasma State component.

Remaining inputs probably from a code specific namelist file

An issue for the RF group is to specify the antenna model in a way that is as generic as possible (antenna structures don't change often, but operational parameters do) while recognizing that each code has a different approach to including antenna currents. For example ray tracing codes require an initial distribution of rays, rather than a \mathbf{J}_{ant} current distribution.

Answer questions above for TORIC and GENRAY and finish answers for AORSA

4. Expectations about input data. e.g., that equilibrium has been reached, or that numerically conservation has been maintained.

RF codes require the $\nabla\mathbf{B}(\mathbf{x})$ tensor. AORSA requires 2nd derivatives of \mathbf{B} . These need to be accurate and smooth.

RF codes require real and velocity space derivatives of $f_j(\mathbf{x}, \mathbf{v})$. These need to be accurate and smooth.

5. Shared infrastructure components such as global time-keeper, interpolators from one representation to another, flux surface average routines, storage of the component's state for restart.

RF codes require accurately interpolated representations of the generic inputs (2.A) on different spatial grids – Cartesian, flux coordinate

6) Physics analysis/development needs, possible mathematical or algorithmic problems, opportunities for improvement, additional opportunities for parallelism that are not presently realized.

Improved treatment of the non-local conductivity operator – parallel gradient effects, collisional and other decorrelation

More efficient field representation allowing for non-uniform meshes

Iterative field solutions for slowly evolving plasma state

Develop methods to treat velocity space unstable distributions – consider time dependent solutions

Better coupling between 3D antenna models and wave solvers – coupling of full wave codes to antenna codes, theory based model for initialization of rays in ray tracing codes

7) Summary of computational needs this component will have when used for the fast MHD campaign. Including computations, memory, amount of time to carry out its designated task in the framework, and kind of parallelism (threaded or distributed memory or both).

Some of this material is listed above for AORSA. But we need to also provide it for TORIC and GENRAY and we need to do more arithmetic to understand the requirements when these codes are applied to SWIM simulations.