# Table of Contents

Algorithms for the full model

# Finite difference discretization

Explicit upwind scheme for $f_{e,i}$ :

$$\frac{f^{n+1}_{s,k,l} - f^{n}_{s,k,l}}{\Delta t} + \begin{pmatrix} v_l \\ c_s E^n_k \end{pmatrix}_+ D^-_{k,l} f^n_s + \begin{pmatrix} v_l \\ c_s E^n_k \end{pmatrix}_- D^+_{k,l} f^n_s = S^n_{s,k,l}$$

with $S_e = 0$, $S_i = \nu f_e$, and $D^{\pm}_{k,l} f := \pm \left( \frac{f_{k\pm1,l} - f_{k,l}}{\Delta x}, \frac{f_{k,l\pm1} - f_{k,l}}{\Delta v} \right)^t$.

# Finite difference discretization

Explicit upwind scheme for $f_{e,i}$ :

$$\frac{f_{s,k,l}^{n+1} - f_{s,k,l}^{n}}{\Delta t} + \begin{pmatrix} v_l \\ c_s E_k^n \end{pmatrix}_+ D_{k,l}^{-} f_s^n + \begin{pmatrix} v_l \\ c_s E_k^n \end{pmatrix}_- D_{k,l}^{+} f_s^n = S_{s,k,l}^{n}$$

with $S_e = 0$, $S_i = \nu f_e$, and $D_{k,l}^{\pm} f \coloneqq \pm \left( \frac{f_{k\pm1,l} - f_{k,l}}{\Delta x}, \frac{f_{k,l\pm1} - f_{k,l}}{\Delta v} \right)^t$.

Integration for the electric field $E$, assuming $E(0) = 0$ :

$$E_k^{n+1} = \frac{1}{\lambda^2} \sum_{\kappa} \frac{1}{\Delta x} \left( n_{i,\kappa}^{n} - n_{e,\kappa}^{n} \right)$$

## Finite difference discretization

Explicit upwind scheme for $f_{e,i}$ :

$$\frac{f_{s,k,l}^{n+1} - f_{s,k,l}^{n}}{\Delta t} + \begin{pmatrix} v_l \\ c_s E_k^n \end{pmatrix}_+ D_{k,l}^- f_s^n + \begin{pmatrix} v_l \\ c_s E_k^n \end{pmatrix}_- D_{k,l}^+ f_s^n = S_{s,k,l}^n$$

with $S_e = 0$, $S_i = \nu f_e$, and $D_{k,l}^{\pm} f := \pm \left( \frac{f_{k\pm1,l} - f_{k,l}}{\Delta x}, \frac{f_{k,l\pm1} - f_{k,l}}{\Delta v} \right)^t$.
Integration for the electric field $E$, assuming $E(0) = 0$ :

$$E_k^{n+1} = \frac{1}{\lambda^2} \sum_{\kappa} \frac{1}{\Delta x} \left( n_{i,\kappa}^n - n_{e,\kappa}^n \right)$$

- CFL condition to ensure the stability of the explicit scheme.

# Finite difference discretization

Explicit upwind scheme for $f_{e,i}$ :

$$\frac{f^{n+1}_{s,k,l} - f^n_{s,k,l}}{\Delta t} + \begin{pmatrix} v_l \\ c_s E^n_k \end{pmatrix}_+ D^-_{k,l} f^n_s + \begin{pmatrix} v_l \\ c_s E^n_k \end{pmatrix}_- D^+_{k,l} f^n_s = S^n_{s,k,l}$$

with $S_e = 0$, $S_i = \nu f_e$, and $D^{\pm}_{k,l} f \coloneqq \pm \left( \frac{f_{k\pm1,l} - f_{k,l}}{\Delta x}, \frac{f_{k,l\pm1} - f_{k,l}}{\Delta v} \right)^t$.
Integration for the electric field $E$, assuming $E(0) = 0$ :

$$E^{n+1}_k = \frac{1}{\lambda^2} \sum_{\kappa} \frac{1}{\Delta x} \left( n^n_{i,\kappa} - n^n_{e,\kappa} \right)$$

- CFL condition to ensure the stability of the explicit scheme.
- First-order approximation, diffusive.

## Semi-Lagrangian scheme

Strang splitting :

$$\frac{\Delta t}{2} \quad \begin{cases} \partial_t f_s + v \partial_x f_s = 0 & \text{Linear advection at constant speed} \\ \lambda^2 \partial_x E = n_i - n_e & \text{Resolution by (numerical) integration} \\ \partial_t f_i = \nu f_e & \text{Pointwise ODE} \end{cases}$$

# Semi-Lagrangian scheme

Strang splitting :

$$\frac{\Delta t}{2} \quad \begin{cases} \partial_t f_s + v\partial_x f_s = 0 & \text{Linear advection at constant speed} \\ \lambda^2 \partial_x E = n_i - n_e & \text{Resolution by (numerical) integration} \\ \partial_t f_i = \nu f_e & \text{Pointwise ODE} \end{cases}$$

$$\Delta t \qquad \partial_t f_s + c_s E_s \partial_v f_s = 0 \quad \text{Again, advection at constant speed}$$

## Semi-Lagrangian scheme

Strang splitting :

$$\frac{\Delta t}{2} \quad \begin{cases} \partial_t f_s + v \partial_x f_s = 0 & \text{Linear advection at constant speed} \\ \lambda^2 \partial_x E = n_i - n_e & \text{Resolution by (numerical) integration} \\ \partial_t f_i = \nu f_e & \text{Pointwise ODE} \end{cases}$$

$$\Delta t \qquad \partial_t f_s + c_s E_s \partial_v f_s = 0 \quad \text{Again, advection at constant speed}$$

$$\frac{\Delta t}{2} \quad \begin{cases} \partial_t f_i = \nu f_e & \text{Pointwise ODE} \\ \lambda^2 \partial_x E = n_i - n_e & \text{Resolution by (numerical) integration} \\ \partial_t f_s + v \partial_x f_s = 0 & \text{Linear advection at constant speed.} \end{cases}$$

## Semi-Lagrangian scheme

Strang splitting :

$$\frac{\Delta t}{2} \quad \begin{cases} \partial_t f_s + v \partial_x f_s = 0 & \text{Linear advection at constant speed} \\ \lambda^2 \partial_x E = n_i - n_e & \text{Resolution by (numerical) integration} \\ \partial_t f_i = \nu f_e & \text{Pointwise ODE} \end{cases}$$

$$\Delta t \qquad \partial_t f_s + c_s E_s \partial_v f_s = 0 \quad \text{Again, advection at constant speed}$$

$$\frac{\Delta t}{2} \quad \begin{cases} \partial_t f_i = \nu f_e & \text{Pointwise ODE} \\ \lambda^2 \partial_x E = n_i - n_e & \text{Resolution by (numerical) integration} \\ \partial_t f_s + v \partial_x f_s = 0 & \text{Linear advection at constant speed.} \end{cases}$$

- The analytical solution to each step is known.

## Semi-Lagrangian scheme

Strang splitting :

$$\frac{\Delta t}{2} \quad \begin{cases} \partial_t f_s + v\partial_x f_s = 0 & \text{Linear advection at constant speed} \\ \lambda^2 \partial_x E = n_i - n_e & \text{Resolution by (numerical) integration} \\ \partial_t f_i = \nu f_e & \text{Pointwise ODE} \end{cases}$$

$$\Delta t \qquad \partial_t f_s + c_s E_s \partial_v f_s = 0 \quad \text{Again, advection at constant speed}$$

$$\frac{\Delta t}{2} \quad \begin{cases} \partial_t f_i = \nu f_e & \text{Pointwise ODE} \\ \lambda^2 \partial_x E = n_i - n_e & \text{Resolution by (numerical) integration} \\ \partial_t f_s + v\partial_x f_s = 0 & \text{Linear advection at constant speed.} \end{cases}$$

- The analytical solution to each step is known.
- Use of the 1D solver to enforce boundary conditions.

# Fixed-point algorithm

Equilibrium state :

$$\begin{cases} v\partial_x f_s - c_s \partial_x \varphi_s \partial_v f_s = S_s, \quad \frac{d}{d\tau}[f_s(x_s(\tau), v_s(\tau))] = S_s(x_s(\tau), v_s(\tau)) \\ -\lambda^2 \partial_{xx}^2 \varphi = n_i - n_e \end{cases}$$

# Fixed-point algorithm

Equilibrium state :

$$\begin{cases} v\partial_x f_s - c_s \partial_x \varphi_s \partial_v f_s = S_s, & \frac{d}{d\tau}[f_s(x_s(\tau), v_s(\tau))] = S_s(x_s(\tau), v_s(\tau)) \\ -\lambda^2 \partial^2_{xx}\varphi = n_i - n_e \end{cases}$$

Suppose $\varphi^k$ is known. Iteration of

$$f_e^{k+1}(x, v) \quad := \quad f_{e,b}(x_e(-\tau), v_e(-\tau))$$

# Fixed-point algorithm

Equilibrium state :

$$\begin{cases} v\partial_x f_s - c_s \partial_x \varphi_s \partial_v f_s = S_s, & \frac{d}{d\tau}[f_s(x_s(\tau), v_s(\tau))] = S_s(x_s(\tau), v_s(\tau)) \\ -\lambda^2 \partial^2_{xx}\varphi = n_i - n_e \end{cases}$$

Suppose $\varphi^k$ is known. Iteration of

$$\begin{aligned} f_e^{k+1}(x,v) &:= f_{e,b}(x_e(-\tau), v_e(-\tau)) \\ f_i^{k+1}(x,v) &:= f_{i,b}(x_i(-\tau), v_i(-\tau)) + \int_{-\tau}^{0} \nu f_e^{k+1}(x_i(s), v_i(s)) ds \end{aligned}$$

# Fixed-point algorithm

Equilibrium state :

$$\begin{cases} v\partial_x f_s - c_s \partial_x \varphi_s \partial_v f_s = S_s, & \frac{d}{d\tau}[f_s(x_s(\tau), v_s(\tau))] = S_s(x_s(\tau), v_s(\tau)) \\ -\lambda^2 \partial^2_{xx}\varphi = n_i - n_e \end{cases}$$

Suppose $\varphi^k$ is known. Iteration of

$$\begin{aligned} f_e^{k+1}(x,v) &:= f_{e,b}(x_e(-\tau), v_e(-\tau)) \\ f_i^{k+1}(x,v) &:= f_{i,b}(x_i(-\tau), v_i(-\tau)) + \int_{-\tau}^{0} \nu f_e^{k+1}(x_i(s), v_i(s))ds \\ -\lambda^2 \partial^2_{xx}\varphi^{k+1} &:= \int_v [f_i^{k+1}(\cdot, v) - f_e^{k+1}(\cdot, v)]dv \end{aligned}$$

Merci.