TGYRO iteration scheme notes

J. Candy

A. Transport equations

We restrict attention in this note to the steady-state transport problem. In this case, the transport equations for ion and electron energy flux are written

$$\frac{1}{V'(r)}\frac{\partial}{\partial r}\left[V'(r)Q_i(r)\right] = (1-\lambda)S_\alpha + S^{ei}_{\rm exch} + S^{i}_{\rm aux}\ , \tag{1}$$

$$\frac{1}{V'(r)}\frac{\partial}{\partial r}\left[V'(r)Q_e(r)\right] = \lambda S_\alpha - S^{ei}_{\rm exch} + S^{e}_{\rm aux} - S_{\rm rad}\ . \tag{2}$$

Here, the energy fluxes are the taken to be the sum of neoclassical transport (Hinton-Hazeltine analytic theory or NEO simulation) and turbulent transport (IFS-PPPL, QFM, TGLF or GYRO simulation):

$$Q_i = Q_i^{\rm Neo} + Q_i^{\rm Turb} \tag{3}$$

$$Q_e = Q_e^{\rm Neo} + Q_e^{\rm Turb} \tag{4}$$

At present we account for sources, radiation and exchange:

$$\begin{aligned}
S_\alpha &\to \text{Thermonuclear source} \\
S^{ei}_{\rm exch} &\to \text{Electron-ion energy exchange} \\
S^{i}_{\rm aux} &\to \text{Ion Auxilliary heating} \\
S^{e}_{\rm aux} &\to \text{Electron Auxilliary heating} \\
S_{\rm rad} &\to \text{Electron radiation} \\
\lambda &\to \text{Fraction of heating to electrons}
\end{aligned}$$

B. Some comments regarding units

In TGYRO, we have found it more convenient to use CGS units rather than employing some variant of the more popular dimensionless normalizations. Thus, we have

$$\text{Source}: \quad S \sim \frac{\text{erg}}{\text{cm}^3\,\text{s}} \tag{5}$$

$$\text{EnergyFlux}: \quad Q \sim \frac{\text{erg}}{\text{cm}^2\,\text{s}} \tag{6}$$

$$\text{Power}: \quad P \sim \frac{\text{erg}}{\text{s}} \to \int_0^r dx\, V'(x) S(x) \tag{7}$$

C. Solution strategy

Rather than solving the equations directly, we prefer to solve the volume-integrated form of the equation so that we can deal directly with the fluxes:

$$Q_i^T(r) \doteq \frac{1}{V'(r)} \int_0^r dx\, V'(x) \left[(1-\lambda)S_\alpha + S_{\text{exch}}^{ei} + S_{\text{aux}}^{i}\right] \tag{8}$$

$$Q_e^T(r) \doteq \frac{1}{V'(r)} \int_0^r dx\, V'(x) \left[\lambda S_\alpha - S_{\text{exch}}^{ei} + S_{\text{aux}}^{e} - S_{\text{rad}}\right] \tag{9}$$

The result is a curious system which depends on both the temperatures and the temperature gradients:

$$Q_i(z_i, z_e, T_i, T_e) - Q_i^T(T_i, T_e) = 0 \tag{10}$$

$$Q_e(z_i, z_e, T_i, T_e) - Q_e^T(T_i, T_e) = 0 \tag{11}$$

where

$$z_i \doteq -\frac{a}{T_i}\frac{\partial T_i}{\partial r} \quad \text{and} \quad z_e \doteq -\frac{a}{T_e}\frac{\partial T_e}{\partial r} \tag{12}$$

It is important to note the connection between profiles and gradients. Specifically, if we enforce the following pedestal boundary conditions at $r = r_*$:

$$T_\sigma(r_*) = T_\sigma^* \ . \tag{13}$$

Then the gradients z_σ uniquely determine the temperature profiles, T_σ:

$$T_\sigma(r) = T_\sigma^* \exp\left(\int_r^{r_*} dx\, z_\sigma(x)\right) \ . \tag{14}$$

D. Formulation on a discrete grid

On a discrete grid $\{r_j\}$, the temperature profile can be approximately determined using the trapezoidal rule

$$T_\sigma(r_{j-1}) = T_\sigma(r_j) \exp\left\{ \left[\frac{z_\sigma(r_j) + z_\sigma(r_{j-1})}{2} \right] [r_j - r_{j-1}] \right\} \ . \tag{15}$$

To put the problem into discrete form, we define a vector of independent variables (gradients) and functions (fluxes):

$$z_{\sigma,j} = z_\sigma(r_j) \ , \tag{16}$$

$$Q_{\sigma,j} = Q_\sigma(r_j) \ , \tag{17}$$

$$Q^T_{\sigma,j} = Q^T_\sigma(r_j) \ . \tag{18}$$

Then, the equations to be solved are

$$\widehat{Q}_{\sigma,j} = \widehat{Q}^T_{\sigma,j} \ . \tag{19}$$

where a hat denotes gyroBohm normalization:

$$\widehat{Q} \doteq \frac{Q}{Q_{\rm GB}} \quad \text{where} \quad Q_{\rm GB} = n_e T_e c_s (\rho_s/a)^2 \ . \tag{20}$$

The goal is to apply Newton's method in a way which is as accurate as possible while still minimizing evaluation of the expensive functions $Q_{\sigma,j}$. Operationally, we make the key assumption that the transport fluxes depend only locally on the gradients (which is approximately true when quantities are normalzied to the gyroBohm unit of flux), so that the Jacobian associated with $Q_{\sigma,j}$ is block diagonal:

$$\widehat{Q}_{\sigma,j}(z^0) - \widehat{Q}^T_{\sigma,j}(z^0) + \frac{\partial \widehat{Q}_{\sigma,j}}{\partial z_{\sigma',j}} \delta z_{\sigma',j} - \frac{\partial \widehat{Q}^T_{\sigma,j}}{\partial z_{\sigma',j'}} \delta z_{\sigma',j'} = 0 \ . \tag{21}$$

Above, we have used the shorthand $z \doteq \{z_{\sigma,j}\}$ and $z^0 \doteq \{z^0_{\sigma,j}\}$. This can be written in terms of Jacobian matrices as

$$\widehat{\mathcal{J}}_{\sigma\sigma',jj'} \, \delta z_{\sigma',j'} = - \left[\widehat{Q}_{\sigma,j}(z^0) - \widehat{Q}^T_{\sigma,j}(z^0) \right] \eta_{\sigma,j} \ , \tag{22}$$

where

$$\widehat{\mathcal{J}}_{\sigma\sigma',jj'} \doteq \mathcal{J}_{\sigma\sigma',jj} \delta_{jj'} - \mathcal{J}^T_{\sigma\sigma',jj'} \ , \tag{23}$$

and the quantity $z^1 = z^0 + \delta z$ is the Newton update for the vector z In Eq. (22), we have introduced a *relaxation parameter* $\eta_{\sigma,j}$. Note that this method generalizes to an arbitrary number of gradients and fluxes per gridpoint. In the case of three radial gridpoints, $\{r_1, r_2, r_3\}$, the Jacobian matrices have the explicit forms

$$\mathcal{J}_{\sigma\sigma',jj'} = \begin{pmatrix} \frac{\partial \widehat{Q}_{i,1}}{\partial z_{i,1}} & \frac{\partial \widehat{Q}_{i,1}}{\partial z_{e,1}} & 0 & 0 & 0 & 0 \\ \frac{\partial \widehat{Q}_{e,1}}{\partial z_{i,1}} & \frac{\partial \widehat{Q}_{e,1}}{\partial z_{e,1}} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\partial \widehat{Q}_{i,2}}{\partial z_{i,2}} & \frac{\partial \widehat{Q}_{i,2}}{\partial z_{e,2}} & 0 & 0 \\ 0 & 0 & \frac{\partial \widehat{Q}_{e,2}}{\partial z_{i,2}} & \frac{\partial \widehat{Q}_{e,2}}{\partial z_{e,2}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{\partial \widehat{Q}_{i,3}}{\partial z_{i,3}} & \frac{\partial \widehat{Q}_{i,3}}{\partial z_{e,3}} \\ 0 & 0 & 0 & 0 & \frac{\partial \widehat{Q}_{e,3}}{\partial z_{i,3}} & \frac{\partial \widehat{Q}_{e,3}}{\partial z_{e,3}} \end{pmatrix} \tag{24}$$

$$\mathcal{J}^T_{\sigma\sigma',jj'} = \begin{pmatrix} \frac{\partial \widehat{Q}^T_{i,1}}{\partial z_{i,1}} & \frac{\partial \widehat{Q}^T_{i,1}}{\partial z_{e,1}} & \frac{\partial \widehat{Q}^T_{i,1}}{\partial z_{i,2}} & \frac{\partial \widehat{Q}^T_{i,1}}{\partial z_{e,2}} & \frac{\partial \widehat{Q}^T_{i,1}}{\partial z_{i,3}} & \frac{\partial \widehat{Q}^T_{i,1}}{\partial z_{e,3}} \\ \frac{\partial \widehat{Q}^T_{e,1}}{\partial z_{i,1}} & \frac{\partial \widehat{Q}^T_{e,1}}{\partial z_{e,1}} & \frac{\partial \widehat{Q}^T_{e,1}}{\partial z_{i,2}} & \frac{\partial \widehat{Q}^T_{e,1}}{\partial z_{e,2}} & \frac{\partial \widehat{Q}^T_{e,1}}{\partial z_{i,3}} & \frac{\partial \widehat{Q}^T_{e,1}}{\partial z_{e,3}} \\ \frac{\partial \widehat{Q}^T_{i,2}}{\partial z_{i,1}} & \frac{\partial \widehat{Q}^T_{i,2}}{\partial z_{e,1}} & \frac{\partial \widehat{Q}^T_{i,2}}{\partial z_{i,2}} & \frac{\partial \widehat{Q}^T_{i,2}}{\partial z_{e,2}} & \frac{\partial \widehat{Q}^T_{i,2}}{\partial z_{i,3}} & \frac{\partial \widehat{Q}^T_{i,2}}{\partial z_{e,3}} \\ \frac{\partial \widehat{Q}^T_{e,2}}{\partial z_{i,1}} & \frac{\partial \widehat{Q}^T_{e,2}}{\partial z_{e,1}} & \frac{\partial \widehat{Q}^T_{e,2}}{\partial z_{i,2}} & \frac{\partial \widehat{Q}^T_{e,2}}{\partial z_{e,2}} & \frac{\partial \widehat{Q}^T_{e,2}}{\partial z_{i,3}} & \frac{\partial \widehat{Q}^T_{e,2}}{\partial z_{e,3}} \\ \frac{\partial \widehat{Q}^T_{i,3}}{\partial z_{i,1}} & \frac{\partial \widehat{Q}^T_{i,3}}{\partial z_{e,1}} & \frac{\partial \widehat{Q}^T_{i,3}}{\partial z_{i,2}} & \frac{\partial \widehat{Q}^T_{i,3}}{\partial z_{e,2}} & \frac{\partial \widehat{Q}^T_{i,3}}{\partial z_{i,3}} & \frac{\partial \widehat{Q}^T_{i,3}}{\partial z_{e,3}} \\ \frac{\partial \widehat{Q}^T_{e,3}}{\partial z_{i,1}} & \frac{\partial \widehat{Q}^T_{e,3}}{\partial z_{e,1}} & \frac{\partial \widehat{Q}^T_{e,3}}{\partial z_{i,2}} & \frac{\partial \widehat{Q}^T_{e,3}}{\partial z_{e,2}} & \frac{\partial \widehat{Q}^T_{e,3}}{\partial z_{i,3}} & \frac{\partial \widehat{Q}^T_{e,3}}{\partial z_{e,3}} \end{pmatrix} \tag{25}$$

An important quantity to measure after a Newton iteration is the residual

$$R^1_{\sigma,j} = \frac{\left[\widehat{Q}_{\sigma,j}(z^1) - \widehat{Q}^T_{\sigma,j}(z^1)\right]^2}{\left[\widehat{Q}_{\sigma,j}(z^1)\right]^2 + \left[\widehat{Q}^T_{\sigma,j}(z^1)\right]^2} \tag{26}$$

If, after a Newton step, any $R^1_{\sigma,j} > R^0_{\sigma,j}$ is not reduced, some strategy must be adopted to modify the gradient vector z^1 and/or the target. This strategy is under development. Note that there are two distinct iterations:

1. A Newton iteration, which is rapidly convergent given that one is close to a root and the $\widehat{Q}$ are smooth functions,

2. A fixed-point iteration following the Newton iteration, because the weak profile variation of $\widehat{Q}$ was ignored

If the temperature dependence of $\widehat{Q}$ was included, there would be no fixed-point iteration component.

E. Numerical implementation

1. Computation of the Jacobian

We approximate the derivatives in the Jacobian matrix using a forward difference approximation

$$\frac{\partial \widehat{Q}_{\sigma,j}}{\partial z_{\sigma',j'}} \simeq \frac{\widehat{Q}_{\sigma,j}(z_{\sigma',j'} + \Delta z) - \widehat{Q}_{\sigma,j}(z_{\sigma',j'})}{\Delta z} \tag{27}$$

A desireable feature of this approximation is that the iteration scheme, Eq. (22) if it converges, will converge to the exact root of the original equations without any influence of the finite-difference truncation error.

2. Iteration logic

If iteration proceeds such that all residuals $R_{\sigma,j}$ decrease at each iteration, then $\eta_{\sigma,j} = 1$ and no additional logic is required. However, in practice, not all residuals will decrease. In this case, we implement a residual management strategy. Here is the complete logic:

for all σ, j **do** $\qquad \triangleright$ Capped Newton step

$\quad \delta z_{\sigma,j} \leftarrow -\left(\widehat{\mathcal{J}}^{-1}\right)_{\sigma\sigma',jj'} \left[\widehat{Q}_{\sigma',j'}(z^0) - \widehat{Q}^T_{\sigma',j'}(z^0)\right] \eta_{\sigma',j'}$

$\quad$ **if** $|\delta z_{\sigma,j}| > \Delta z_{\max}$ **then**

$\qquad \delta z_{\sigma,j} \leftarrow \mathrm{sgn}(\delta z_{\sigma,j})\, \Delta z_{\max}$

$\qquad z^1_{\sigma,j} \leftarrow z^0_{\sigma,j} + \delta z_{\sigma,j}$

$\quad$ **end if**

end for

```
for all σ, j do                                  ▷ Manage increasing residual
    if R^1_{σ,j} > R^0_{σ,j} then
        z^1_{σ,j} ← z^0_{σ,j}
        η_{σ,j} ← η_{σ,j}/C_η
        if η_{σ,j} < 1/C_η^3 then
            η_{σ,j} ← 0.75 C_η
        end if
    else
        η_{σ,j} ← 1
    end if
end for
```

Then, the residuals R^1 are recomputed and the Newton stepping continues.