`gs2` notes

T. Tatsuno

March 22, 2006

1 Open Questions

List of my questions.

1. I think `add_nonlinear_terms` only evaluates Poisson bracket and puts it in `g1`. Where is it added? $\rightarrow$ It's added at `set_source` in `dist_fn.f90` by Euler or 2nd order Adams-Bashforth scheme, but why is the factor 0.5 multiplied? $\rightarrow$ Is this factor 0.5 because `set_source` < `get_source_term` < `invert_rhs` < `timeadv` is called twice in `advance_implicit`?

2. What does `istep_last` do in `add_nl`? $\rightarrow$ It's for the check if time step has advanced from the last call.

3. Can `load_kx_apar` in `nonlinear_terms.f90` be simplified if `vpa(*,2,*)` $=$ $-$`vpa(*,1,*)`? So as `load_ky_apar`.

4. Is it okay to use the same `timeadv` for the first and the second implicit time stepping in `advance_implicit`? I'm assuming it uses the scheme described in Kotschenreuther paper Sec.3.5.

5. At `invert_rhs_1` in `dist_fn.f90`, the 'homogeneous' solution `g1` is obtained only by multiplying `r`. Is `g1` at step n supposed to be zero? $\rightarrow$ yes they are. This is what the 'homogeneous' solution mean in Kotschenreuther paper.

6. At `invert_rhs_1` in `dist_fn.f90`, is it okay to use `r` and `ainv` defined for $\hat{v}_\parallel > 0$ in the calculation of `gnew` for `vpar` < 0?

7. In `get_source_term`, there appear a lot of cross terms like `vpac` $*$ $[J_0($`ig`$) + J_0($`ig+1`$)]$ or something like that. Is this what is meant as a numerical scheme?

8. On the one hand, density, temperature, and their gradients are basically field quantities. And on the other hand, `spec%dens`, `spec%temp`, `spec%fprim` and `spec%tprim` are just real values. How can I understand it? Are they just normalization factors and don't evolve in time?

9. In `get_source_term`?, when you use finite `bakdif`, do you not need to implement it in the source term either?

10. Do we need to convert `total` into 1d array `work` in `integrate_species` before `allreduce`? Can we use `reshape` function rather than multiple `do`-loops?

11. Can I remove some of the logical variable `alloc` by using `if (.not.allocated(...)) ...`?

12. For `nesuper` $= 1$, `xsup` and `wsup` can be made correct by setting them unity

13. Why `gbdrift` and `gbdrift0`? Which is ∇B-drift? Same with `cvdrift` and `cvdrift0`?

14. What are `gds2`, `gds21`, and `gds22`?

15. Can we include an equilibrium flow in distribution function?

16. How is g obtained from a given $A_\parallel$ profile in `recon10.in`?

17. How is perturbation in $A_\parallel$ given in `recon10a.in`?

18. Is collision operator fixed? Can I have the fixed one?

19. At `init_kt_grids` in `kt_grids.f90`, is it ok to make `tnorm` optional even if `norm_option` $=$ `bd`? Or add error output in case it is missed?

20. Is `idfit_eq` in `theta_grid_knobs` working?

21. Does `alloc` at `allocate_arrays` in `fields.f90` need `save` attribute? Same with `first` of `init_integrations` in `le_grids.f90`.

22. In `dist_fn.f90`, `adiabatic_option_zero` is not used.

23. Do variables need `save` attribute when defined in the common area (i.e., before `contains` statement) of each modules?

24. Do `fapar`, `faperp`, and `fphi` need default values at `read_parameters` in `run_parameters.f90`?

25. pitch angle grid: `xx(1:ng2)` and `wx(1:ng2)` are the Legendre zeros and weights rescaled to $(1,0)$ and in descending order.

$$\texttt{al(1:ng2)} = \frac{1}{B_{\max}}(1 - \texttt{xx(1:ng2)}^2) \tag{1}$$

$$\begin{aligned} w_l(\theta,\lambda) &= 2w_x(\lambda)\sqrt{\frac{B(\theta)}{B_{\max}}\frac{\frac{1}{B_{\max}} - \texttt{al}(\lambda)}{\frac{1}{B(\theta)} - \texttt{al}(\lambda)}} \\ &= 2w_x(\lambda)\frac{B(\theta)}{B_{\max}}\sqrt{\frac{1-\lambda B_{\max}}{1-\lambda B(\theta)}} \end{aligned} \tag{2}$$

On the other hand, the integral we need is

$$\begin{aligned} I &= B(\theta)\int_0^{1/B_{\max}} \frac{1}{\sqrt{1-\lambda B(\theta)}}\, d\lambda \\ &= B(\theta)\int_0^{1/B_{\max}} \sqrt{\frac{1-\lambda B_{\max}}{1-\lambda B(\theta)}}\frac{1}{\sqrt{1-\lambda B_{\max}}}\, d\lambda \\ &= \frac{2B(\theta)}{B_{\max}}\int_0^1 \sqrt{\frac{1-\lambda(X)B_{\max}}{1-\lambda(X)B(\theta)}}\, dX \end{aligned} \tag{3}$$

We are having a weird factor with square root. Is this a quadrature? I mean, if there is additional weight, that has to be removed. Otherwise we have to use a proper function for that weight!? I don't think this can give exponential convergence even if the factor does not have branch point.

I wonder if Chebyshev is better by defining $X = \sqrt{\lambda B(\theta)}$ or anything like that. Does Candy & Waltz use different scheme?

Maybe this is okay. We need to integrate something like

$$I' = B(\theta)\int_0^{1/B_{\max}} \frac{f(\lambda)}{\sqrt{1-\lambda B(\theta)}}\, d\lambda \tag{4}$$

Then, by regarding the weight unity and taking the factor as part of the integrand by muitiplying it with $f(\lambda(X))$, we may use Legendre zeros and weights. Since weights are always multiplied with the integrand, the code takes care of the multiplication of the factor by multiplying it onto weights instead of $f(\lambda(X))$.

Then, the next question is: Does the branch point matters at $x = 1$?

26. In root finding at `setegrid` in `le_grids.f90`, why cheking every other nodes instead of adjacent ones?

27. When is `xgrid_v` used at all?

28. There are `g*_lo%ulim_proc` and `g*_lo%ulim_alloc`. The definition of `ulim_alloc` is

    `ulim_alloc` = `max(llim_proc,ulim_proc)`

    Isn't it always same with `ulim_proc`? Or, can `blocksize` be zero? If so, shouldn't we use `ulim_alloc` any time? What is the difference?

29. Does the argument of Bessel function only include k_y at `init_bessel` in `dist_fn.f90`? Why? Don't need k_x since it has to be $k_\perp v_\perp/\Omega$?

30. What's the meaning of the value `upar0` = 0.436 in `recon10a.in`?

31. logical variable `list` (in `gs2.f90`) determines the run type: if the command line argument is '`?*.list`' (judged at `run_type` in `file_utils.f90`) then `list` = `.true.`: make multiple runs changing parameters?

32. What are `lavg` and `tavg` doing at `time_message` in `gs2_reinit.f90`?

33. Why is `cbuff` necessary? In parent process, `run_name` still points `arun_name`, but in child processes, it points `cbuff`. `cbuff` = `arun_name` only at parent process. `arun_name` is undefined at child processes.

34. What is θ used for `ntheta` and `nperiod`? In toroidal or poloidal direction?

35. What is `alpmhd`? It is neither defined at `read_parameters` in `theta_grid_params` nor in `gs2_template.in` file.

36. Is `itor` working? What is it for then?

37. Are ∇B and curvature drift same in most cases except when ∇p is large?

38. What's the difference between `theta` and `theta0`? Why is `theta0` 2D?

39. What are `pure` and `elemental` functions?

2 Closed Questions

1. `akx_out` is the wave number used in most of the output including the one in `gs2_io.f90`, but it is defined in `kt_grids.mod` as

$$\texttt{akx\_out} = \texttt{akx}/\sqrt{2} \tag{5}$$

   which I guess corresponds to the conversion of the wave number in BD normalization to MTK one. On the other hand, in the output file `recon10?.out.nc`, we see that `kx(1)` $= 1/6$, which has to be the value of `akx`! Why? $\rightarrow$ You are missing the multiplication of $\sqrt{2}$ on wave numbers at `get_grids` in `kt_grids.mod`. This converts all `akx` in the code into MTK normalization.

2. Why `ostride` and `odist` are both zero at `transform_x5d` in `gs2_transforms.f90`? $\rightarrow$ since `xf_` and `xb_fft` do in-place transforms.

3. What is `ainv` in `dist_fn.f90`? $\rightarrow$ inverse of the coefficient of f_{i+1}^{n+1} in Kotschenreuther paper (14).

4. In the beginning of `fields_explicit.f90`, it says NOT UP TO DATE ... DO NOT USE. Should we maintain it? $\rightarrow$ maybe not.

Some notes.

1. `theta_grid.f90` consists of six modules of `theta_grid`, `theta_grid_params`, `theta_grid_gridgen`, `theta_grid_salpha`, `theta_grid_eik`, and `theta_grid_file`.

2. `DOS` mode input file doesn't work with `ingen` and probably with `gs2`, either.

3 Modifications I want to make

1. Can `get_unused_unit` in `file_utils.f90` be simplified by `opened` specifier?

2. Can I use `spread` for defining `theta0` at `range_get_grids` in `kt_grids.f90`?

3. Can I replace multiple `do` loops by `forall`? $\rightarrow$ don't do that. `forall` statement is still inefficient.

4. At `efitin` in `eeq.f90`, inquire existance of `eqfile`

5. Is it difficult to use different energy grids for different species? We may not be able to make simulation with vastly different temperatures between species.

4 Variables

By default, I don't mean the value in the input file, but that used in the actual calculation.

scalars

name	type	default value	description
`nx`	int	0	number of grid points in x real space
`ny`	int	0	number of grid points in y real space
`ntheta0`	int	`2*((nx-1)/3)+1`	number of valid modes in x
`naky`	int	`(ny-1)/3+1`	half number of valid modes in y
`ntgrid`	int		half number of grid point in z (finite difference)?
`negrid`	int	10	total number of energy grid
`ngauss`	int	5	half number of λ grid points
`ng2`	int	`ngauss` $* 2$	
`nlambda`	int	`ng2` $+$ `nbset` (`eps` > 0)	number of grid points in $\lambda = \mu/E$
		`ng2` (otherwise)	
`y0`	real	2.0	box length in y by multiple of 2π
`ly`	real	$2\pi *$ `y0`	box length in y
`rtwist`	real	1.0	l_y/l_x aspect ratio
`lmax`	int	`nlambda` $- 1$ (`eps` $> 0.$)	maximum value of λ?
		`nlambda` (otherwise)	
`shat`	real		$\hat{s} = \frac{r}{q}\frac{dq}{dr}$
`igomega`	int	0	`ig` to output in 2d

1d arrays

name	type	dim	description
`akx`	real	`ntheta0`	$\hat{k}_{x\mathrm{M}}$ wavenumbers in x (reversed in the middle)
`aky`	real	`naky`	$\hat{k}_{y\mathrm{M}}$ wavenumbers in y in MTK normalization
`al`	real	`nlambda`	pitch-angle grid $\lambda = \mu/E$ (weights: `wl` in 2d)

2d arrays

name	type	dim	description
`theta0`	real	`ntheta0` $\times$ `naky`	`akx(i)/(aky(2:)*shat)`, `theta0(:,1)` = 0. ??
`akr`	real	$ntg \times$ `ntheta0`	`akx(it)*sqrt(abs(gds22(:)))/abs(shat)` ??
`e`	real	`negrid` $\times$ `nspec`	energy grid
`w`	real	`negrid` $\times$ `nspec`	energy weights
`wl`	real	$ntg \times$ `nlambda`	pitch-angle weights
`vperp2`	real	$ntg \times$ `glo`	$\check{v}_\perp^2$
`anon`	real	`negrid` $\times$ `nspec`	equals unity unless `slowing_down_species`
`aj0`	real	$ntg \times$ `glo`	Bessel function J_0
`aj1`	real	$ntg \times$ `glo`	Bessel function J_1

where ntg = `-ntgrid:ntgrid`.

3d arrays

name	type	dim	description
`vpa`	real	$ntg \times 2 \times$ `glo`	$\check{v}_\parallel$ or zero (for non-passing zone)
`phi`	complex	$ntg \times$ `ntheta0` $\times$ `naky`	electrostatic field
`apar`	complex	$ntg \times$ `ntheta0` $\times$ `naky`	$A_\parallel$
`aperp`	complex	$ntg \times$ `ntheta0` $\times$ `naky`	$A_\perp$
`g`	complex	$ntg \times 2 \times$ `glo`	distribution function

5 Equations found

When one chooses `s-alpha` equilibrium option, toroidal magnetic field profile `bmag` is defined at `salpha_get_grids` as

$$\texttt{bmag} = 1 - \epsilon\cos\theta - \alpha_1\cos(3\theta) \qquad \text{for } \texttt{model\_option} = \texttt{'alpha1'} \tag{6}$$

$$\texttt{bmag} = 1 - \epsilon\cos\theta \qquad \text{for } \texttt{model\_option} = \texttt{'b2'} \tag{7}$$

$$\texttt{bmag} = \frac{1}{1+\epsilon\cos\theta} \qquad \text{otherwise} \tag{8}$$

where $\epsilon(=$ `eps`$) = r/R$, r and R denote minor radius of interest and major radius, respectively. Thus θ is found to be poloidal angle.

$\boxed{\texttt{init\_vpar}}$

For untrapped particle

$$\texttt{vpa(ig)} = \sigma\sqrt{E(1-\lambda B(\texttt{ig}))} \tag{9}$$

$$\texttt{vpac(ig)} = \frac{1}{2}[\texttt{vpa(ig)} + \texttt{vpa(ig+1)}] \tag{10}$$

$$\tag{11}$$

and for nonpassing zone

$$\texttt{vpa} = 0 \tag{12}$$

$$\texttt{vpac} = \sigma \tag{13}$$

where $\sigma = \pm 1$ denotes the coordinate for the sign of $v_\parallel$. And then `vpar` is defined by

$$\texttt{vpar(ig)} = \frac{Z}{\sqrt{mT}}\texttt{tunits}\frac{\Delta t}{2\Delta\theta}\frac{1}{2}[\texttt{gradpar(ig)} + \texttt{gradpar(ig+1)}] * \texttt{vpac(ig)} \tag{14}$$

where `gradpar(:)` $=$ `kp` in `salpha` option.

$\boxed{\texttt{init\_wdrift}}$

For $k_y = 0$,

$$\texttt{wdrift} = \frac{k_x E}{\texttt{shat}}\frac{\texttt{delt}}{2}\left(v_{\text{curv0}}[1 - \lambda B(\theta)] + \frac{1}{2}v_{\nabla B0}\lambda B(\theta)\right) \tag{15}$$

$$= \omega_{\text{d}} * \texttt{delt}? \tag{16}$$

and for $k_y \neq 0$,

$$\texttt{wdrift} = \left[(v_{\text{curv}} + \theta_0 v_{\text{curv0}})(1 - \lambda B(\theta)) + (v_{\nabla B} + \theta_0 v_{\nabla B0})\frac{1}{2}\lambda B(\theta)\right] E * \texttt{delt} * \texttt{wunits} \tag{17}$$

where from `adjust_time_norm` in `run_parameters.f90`,

$$\texttt{wunits} = \begin{cases} 1 & (\texttt{wstar\_units} = \texttt{.true.}) \\ k_y/2 & (\texttt{wstar\_units} = \texttt{.false.}) \end{cases} \tag{18}$$

$$\texttt{tunits} = \begin{cases} 2/k_y & (\texttt{wstar\_units} = \texttt{.true.}\ \&\ k_y \neq 0) \\ 1 & (\texttt{wstar\_units} = \texttt{.false.}) \end{cases} \tag{19}$$

$$\texttt{funits} = \begin{cases} 1 & (\texttt{wstar\_units} = \texttt{.true.}) \\ \texttt{tnorm} & (\texttt{wstar\_units} = \texttt{.false.}) \end{cases} \tag{20}$$

$$\texttt{woutunits} = \begin{cases} k_y/\sqrt{2} & (\texttt{wstar\_units} = \texttt{.true.}) \\ \texttt{tnorm} & (\texttt{wstar\_units} = \texttt{.false.}) \end{cases} \tag{21}$$

$\boxed{\texttt{init\_wstar}}$

$$\texttt{wstar} = \texttt{delt} * \texttt{wunits} * [\texttt{fprim} + \texttt{tprim} * (E - 1.5)] \tag{22}$$

$\boxed{\texttt{init\_bessel}}$

$$\texttt{kperp2} = \begin{cases} \dfrac{\texttt{gds22}}{(\texttt{shat})^2}k_x^2 & (k_y = 0) \\ (\texttt{gds2} + 2\theta_0\texttt{gds21} + \theta_0^2\texttt{gds22})\, k_y^2 & (k_y \neq 0) \end{cases} \tag{23}$$

with the argument

$$\texttt{arg} = \frac{\sqrt{mT}}{|Z|}\sqrt{\texttt{kperp2}\frac{\lambda E}{B}}, \tag{24}$$

we define the Bessel functions

$$\texttt{aj0} = J_0(\texttt{arg}), \quad \texttt{aj1} = J_1(\texttt{arg}) \tag{25}$$

with a formula taken from Abramovitz & Stegun (page 369, 9.4). By the way, in `recon10a.in`,

$$\texttt{gds2} = 1, \quad \texttt{gds21} = -(\texttt{shat})^2\theta, \quad \texttt{gds22} = (\texttt{shat})^2, \tag{26}$$

and

$$\theta_0 = \begin{cases} \frac{k_x}{k_y \texttt{shat}} & (k_y \neq 0) \\ 0 & (k_y = 0) \end{cases} . \tag{27}$$

Therefore,

$$\texttt{kperp2} = \begin{cases} k_x^2 & (k_y = 0) \\ k_x^2 + k_y^2 - 2k_x k_y \theta * \texttt{shat} & (k_y \neq 0) \end{cases} \tag{28}$$

`shat` is small ($= 10^{-6}$), but `kperp2` is not exactly $k_\perp^2$ for $\theta \neq 0$. Is this okay?

$\boxed{\texttt{init\_invert\_rhs}}$

$$\begin{aligned}
\texttt{ainv}(ntg,\texttt{glo}) &= \frac{1}{1+\texttt{bd}+(1-\texttt{fexp})\frac{T}{Z}[i\texttt{wd}(1+\texttt{bd})+2\texttt{vp}]} && (29)\\
&\sim \left[1+(1-f_{\text{exp}})\Delta t\left(i\omega_{\text{d}}+\frac{2\hat{v}_{\|}}{\Delta\theta}\right)\right]^{-1} && (30)\\
&= (\text{coeff. of } f_{i+1}^{n+1} \text{ in the lhs})^{-1} = (D_4 \text{ plus } \texttt{bd} \text{ factor})^{-1} && (31)\\
\texttt{r}(ntg,\texttt{glo}) &= \frac{1-\texttt{bd}+(1-\texttt{fexp})\frac{T}{Z}[i\texttt{wd}(1-\texttt{bd})-2\texttt{vp}]}{1+\texttt{bd}+(1-\texttt{fexp})\frac{T}{Z}[i\texttt{wd}(1+\texttt{bd})+2\texttt{vp}]} && (32)\\
&= \left[1+(1-f_{\text{exp}})\Delta t\left(i\omega_{\text{d}}-\frac{2\hat{v}_{\|}}{\Delta\theta}\right)\right]*\texttt{ainv} && (33)\\
&= (\text{coeff. of } f_{i}^{n+1} \text{ in the lhs})*\texttt{ainv} = (D_3 \text{ plus } \texttt{bd} \text{ factor})*\texttt{ainv} && (34)\\
\texttt{a}(ntg,\texttt{glo}) &= 1+\texttt{bd}+\texttt{fexp}\frac{T}{Z}\left[-i\texttt{wd}(1+\texttt{bd})-2\texttt{vp}\right] && (35)\\
&= 1-f_{\text{exp}}\Delta t\left(i\omega_{\text{d}}+\frac{2}{\Delta\theta}\hat{v}_{\|}\right)?(\text{haven't checked } \omega_{\text{d}} \text{ factor}) && (36)\\
&= D_2 \text{ (coeff. of } f_i^{n+1} \text{ term) plus } \texttt{bd} \text{ factor} && (37)\\
\texttt{b}(ntg,\texttt{glo}) &= 1-\texttt{bd}+\texttt{fexp}\frac{T}{Z}\left[-i\texttt{wd}(1-\texttt{bd})+2\texttt{vp}\right] && (38)\\
&= 1-f_{\text{exp}}\Delta t\left(i\omega_{\text{d}}-\frac{2}{\Delta\theta}\hat{v}_{\|}\right)?(\text{haven't checked } \omega_{\text{d}} \text{ factor}) && (39)\\
&= D_1 \text{ (coeff. of } f_i^{n} \text{ term) plus } \texttt{bd} \text{ factor} && (40)
\end{aligned}$$

where `wd` = `wdrift` ($= 0$ in reconnection) and `vp` = `vpar`$(ntg, 1,$ `glo`$)$. Note that they are all defined for positive $v_{\|}$.

`fexp` is a complex number. What's the meaning of the imaginary part? The meaning of the real part of `fexp` and `bd` (= `bakdif`) is explained later (in `get_source_term`?).

init_fieldeq

$$\texttt{gamtot} = \sum_s \frac{nZ_s^2}{T_s} \int\int (1 - J_0^2) * \texttt{anon}\, d\lambda\, dE + \texttt{kperp2} * \texttt{poisfac} \tag{41}$$

$$\texttt{gamtot1} = \sum_s nZ_s \int\int 2v_\perp^2 J_0 J_1 * \texttt{anon}\, d\lambda\, dE \tag{42}$$

$$\texttt{gamtot2} = \sum_s nT \int\int 2v_\perp^4 J_1^2 * \texttt{anon}\, d\lambda\, dE \tag{43}$$

Of course these integrations are done with proper weights and Jacobians described elsewhere.

$\boxed{\texttt{invert\_rhs}}$

In `dist_fn.f90`. Add source term

$$\texttt{sourcefac} = \begin{cases} s_0 \exp[(-i\omega_0 + \gamma_0)t] & (t > t_0) \\ \dfrac{1}{2}\left(1 - \cos\dfrac{\pi t}{t_0}\right) \exp[(-i\omega_0 + \gamma_0)t] & (t \le t_0) \end{cases}, \tag{44}$$

where s_0 (`source0`), ω_0 (`omega0`), γ_0 (`gamma0`), and t_0 (`t0`) are given parameters specified in `source_knobs`.

$\boxed{\texttt{get\_source\_term}}$

Writing $f_\phi = \texttt{fphi}$, $f_{\exp} = \texttt{fexp}$ ($= 1 - \delta$ in Kotschenreuther paper)

$$\texttt{phigavg} = f_\phi J_0 \left[f_{\exp}\phi^n + (1 - f_{\exp})\phi^{n+1}\right] + f_{A_\perp} \frac{T}{Z} v_\perp^2 J_1 \left[f_{\exp} A_\perp^n + (1 - f_{\exp}) A_\perp^{n+1}\right] \tag{45}$$

$$\texttt{apargavg} = f_{A_\parallel} J_0 \left[f_{\exp} A_\parallel^n + (1 - f_{\exp}) A_\parallel^{n+1}\right] \tag{46}$$

$$\texttt{ufac} = 2 * \texttt{uprim} + \frac{\sqrt{\pi}}{4} E^{3/2} * \texttt{uprim2} \tag{47}$$

The following is for reconnection problem:

$$\texttt{source(ig)} = -2\texttt{vpar(ig)}\,\phi_m - \frac{Z}{\sqrt{mT}} \texttt{vpac(ig)} \frac{J_0(\texttt{ig}) + J_0(\texttt{ig}+1)}{2} A_{\parallel m} \tag{48}$$

where

$$\phi_m = \texttt{phigavg}(\texttt{ig}+1) - \texttt{phigavg}(\texttt{ig}) \sim \Delta\theta \frac{\partial (J_0 \phi)}{\partial \theta} \tag{49}$$

$$\begin{aligned} A_{\|m} &= A_\|^{n+1}(\texttt{ig}+1) + A_\|^{n+1}(\texttt{ig}) - A_\|^{n}(\texttt{ig}+1) - A_\|^{n}(\texttt{ig}) \\ &\sim 2\Delta t \frac{\partial A_\|}{\partial t} \end{aligned} \tag{50}$$

$$\texttt{phigavg} = J_0(\texttt{ig}) \left[f_{\rm exp} \phi^n(\texttt{ig}) + (1-f_{\rm exp})\phi^{n+1}(\texttt{ig}) \right] \tag{51}$$

$$\texttt{vpar(ig)} = \frac{Z}{\sqrt{mT}} \frac{\Delta t}{\Delta \theta} \texttt{kp} * \frac{\check{v}_\|(\texttt{ig}) + \check{v}_\|(\texttt{ig}+1)}{2} \sim \frac{\hat{Z}}{\hat{T}} \frac{\Delta t}{\Delta\theta} \hat{v}_\| \tag{52}$$

$$\texttt{vpac(ig)} = \frac{\check{v}_\|(\texttt{ig}) + \check{v}_\|(\texttt{ig}+1)}{2} \tag{53}$$

Thus,

$$\texttt{source(ig)} \sim -\frac{\hat{Z}}{\hat{T}} \hat{v}_\| \left[\frac{\partial (J_0\phi)}{\partial\theta} + J_0 \frac{\partial A_\|}{\partial t} \right] (2\Delta t) \tag{54}$$

where ϕ and $A_\|$ are evaluated from both time steps of n and $n+1$. This is the expression you get at `set_source`. Is the sign of the first term in $[\ldots]$ okay??

Moreover, if `nonlin` = `.true.`, then add nonlinear terms

$$\texttt{source} = (54) + \frac{1}{2} \frac{\texttt{delt}}{\texttt{tnorm}} \times (\text{nonlinear terms}), \tag{55}$$

in Euler scheme at the first time step and in second order Adams-Bashforth scheme for the rest. `tnorm` $= \sqrt{2}$ in reconnection runs, and `delt` is multiplied by `tnorm` in `init_run_parameters`. So the factor `delt/tnorm` corresponds to the real `delt` specified in the input file. The Δt (= `delt`) in the linear terms is $\sqrt{2}$ times larger than that. The precise form of the nonlinear terms is described in `add_nl`.

Next, we go back to `get_source_term` and around the place where `Do matrix multiplications`... For $\sigma = 1$

$$\texttt{b}(ig, iglo) * g(ig, 1, iglo) + \texttt{a}(ig, iglo) * g(ig+1, 1, iglo) \tag{56}$$

is added to `source(ig)`, which corresponds to the $\texttt{g}^n$ terms arising from the finite difference form of the lhs:

$$\begin{aligned} \frac{\partial f}{\partial t} + i\omega_{\rm d} f + \hat{v}_\| \frac{\partial f}{\partial \theta} &\sim \frac{1}{2\Delta t} \left[(f_i^{n+1} + f_{i+1}^{n+1}) - (f_i^n + f_{i+1}^n) \right] \\ &+ \frac{i\omega_{\rm d}}{2} \left[(1-f_{\rm exp})(f_i^{n+1} + f_{i+1}^{n+1}) + f_{\rm exp}(f_i^n + f_{i+1}^n) \right] \\ &+ \frac{\hat{v}_\|}{\Delta\theta} \left[(1-f_{\rm exp})(f_{i+1}^{n+1} - f_i^{n+1}) + f_{\rm exp}(f_{i+1}^n - f_i^n) \right] \end{aligned} \tag{57}$$

where $\omega_{\rm d} = 0$ in the reconnection problem. For $\sigma = -1$, the sign change of $\hat{v}_\parallel$ is taken care of by multiplying a and b oppositely on $g(ig)$ and $g(ig+1)$, respectively, because the definition of `a` and `b` uses `vpar`$(ntg, 1,$ `glo`$)$ which is the positive part of $\hat{v}_\parallel$.

Okay, let's think about `bakdif` now. It is introduced in order to make $\hat{v}_\parallel \partial_\theta f$ term an upwind difference scheme. As is described in (57), everything is evaluated at grid point $i+1/2$ in θ. Instead of changing the finite differencing of $\partial_\theta f$, we shift the grid point for other terms to be evaluated a little bit forward. Then, the scheme is going to be upwind finite difference.

Let's work on the terms appearing in (57), and we write $\beta =$ `bakdif` for simplicity. Any term evaluated at $i+1/2$ is expressed as follows:

$$f_{i+1/2} = \frac{1}{2}(f_{i+1} + f_i). \tag{58}$$

By shifting it forward, we may write it as

$$f_{i+(1+\beta)/2} = \frac{1}{2}[(1+\beta)f_{i+1} + (1-\beta)f_i], \tag{59}$$

where $0 \le \beta \le 1$ and $\beta = 0$ corresponds to second order centered difference scheme (may β be larger than unity?).

Thus, for the terms in (57), they are finite differenced as

$$\begin{aligned}
&\left(\frac{\partial f}{\partial t} + i\omega_{\rm d} f\right)_{i+(1+\beta)/2} + \left(\hat{v}_\parallel \frac{\partial f}{\partial \theta}\right)_{i+1/2} \\
&\sim \frac{1}{2}\left[(1+\beta)\frac{f_{i+1}^{n+1} - f_{i+1}^n}{\Delta t} + (1-\beta)\frac{f_i^{n+1} - f_i^n}{\Delta t}\right] \\
&+\frac{i\omega_{\rm d}}{2}\left\{(1-f_{\rm exp})\left[(1+\beta)f_{i+1}^{n+1} + (1-\beta)f_i^{n+1}\right] + f_{\rm exp}\left[(1+\beta)f_{i+1}^n + (1-\beta)f_i^n\right]\right\} \\
&+\frac{\hat{v}_\parallel}{\Delta\theta}\left[(1-f_{\rm exp})(f_{i+1}^{n+1} - f_i^{n+1}) + f_{\rm exp}(f_{i+1}^n - f_i^n)\right] \\
&= \frac{1}{2\Delta t}\left[\frac{1}{\texttt{ainv}} f_{i+1}^{n+1} + \frac{\texttt{r}}{\texttt{ainv}} f_i^{n+1} - \texttt{a} f_{i+1}^n - \texttt{b} f_i^n\right].
\end{aligned} \tag{60}$$

Here comes the question. When you use finite `bakdif`, do you not need to implement it in the source term either?

$\boxed{\texttt{invert\_rhs\_1}}$

Is it okay to use `r` and `ainv` defined for $\hat{v}_\parallel > 0$ in the calculation of `gnew` for `vpar` < 0?

$\boxed{\texttt{getan}}$

This is in `dist_fn.f90`.

$$\texttt{antot} = \sum_s nZ \int\int J_0 * \texttt{gnew}\, d\lambda\, dE \tag{61}$$

$$\begin{aligned} \texttt{antota} &= \sum_s 2 * \texttt{beta} * nZ \sqrt{\frac{T}{m}} \int\int J_0 \check{v}_\parallel * \texttt{gnew}\, d\lambda\, dE \\ &= \sum_s 2 * \texttt{beta} * nZ \int\int J_0 \hat{v}_\parallel * \texttt{gnew}\, d\lambda\, dE \end{aligned} \tag{62}$$

$$\texttt{antotp} = \sum_s nT \int\int J_1 \check{v}_\perp^2 * \texttt{gnew}\, d\lambda\, dE \tag{63}$$

$\boxed{\texttt{getfieldeq1}}$

$\boxed{\texttt{add\_nl}}$

If the time step (`istep`) is advanced from the last call (`istep_last`),

$$\texttt{g2} = \texttt{g1} \tag{64}$$

$$\texttt{g1} = ik_x \left[J_0 \left(f_\phi \phi - \hat{v}_\parallel f_{A_\parallel} A_\parallel^n \right) + \frac{2m}{Z} J_1 \hat{v}_\perp^2 f_{A_\perp} A_\perp^n \right] \tag{65}$$

$$\texttt{ba} = \mathcal{F}\left(\texttt{g1}\right) \tag{66}$$

$$\texttt{g1} = \frac{Z}{T} \left[ik_y \left(J_0 f_\phi \phi + \frac{2m}{Z} J_1 \hat{v}_\perp^2 f_{A_\perp} A_\perp^n \right) \right] + ik_y \texttt{g} \tag{67}$$

$$\texttt{gb} = \mathcal{F}\left(\texttt{g1}\right) \tag{68}$$

$$\texttt{bracket} = \texttt{ba} * \texttt{gb} * \texttt{kxfac} \tag{69}$$

$$\texttt{g1} = ik_y \left[J_0 \left(f_\phi \phi - \hat{v}_\parallel f_{A_\parallel} A_\parallel^n \right) + \frac{2m}{Z} J_1 \hat{v}_\perp^2 f_{A_\perp} A_\perp^n \right] \tag{70}$$

$$\texttt{ba} = \mathcal{F}\left(\texttt{g1}\right) \tag{71}$$

$$\texttt{g1} = \frac{Z}{T} \left[ik_x \left(J_0 f_\phi \phi + \frac{2m}{Z} J_1 \hat{v}_\perp^2 f_{A_\perp} A_\perp^n \right) \right] + ik_x \texttt{g} \tag{72}$$

$$\texttt{gb} = \mathcal{F}\left(\texttt{g1}\right) \tag{73}$$

$$\texttt{bracket} = \texttt{bracket} - \texttt{ba} * \texttt{gb} * \texttt{kxfac} \tag{74}$$

$$\texttt{g1} = \mathcal{F}\left(\texttt{bracket}\right) \tag{75}$$

where $\texttt{kxfac} = 1$ when $\texttt{equilibrium\_option} = \texttt{s-alpha}$. Why do $(Z/T)\,[...]$ terms appear in (67) and (72)?

6 Normalization

Energy is normalized by

$$\check{E}_{\mathrm{s}} = \frac{E_{\mathrm{s}}}{m_{\mathrm{s}} v_{\mathrm{ts}}^2/2} \tag{76}$$

where

$$v_{\mathrm{ts}} = \begin{cases} \sqrt{2T_{\mathrm{s}}/m_{\mathrm{s}}} & \texttt{norm\_option} = \texttt{with\_root\_2} \\ \sqrt{T_{\mathrm{s}}/m_{\mathrm{s}}} & \texttt{norm\_option} = \texttt{no\_root\_2} \end{cases} . \tag{77}$$

Note that energy normalization is done in terms of the thermal velocity for each specy.

In the following, we first write everything in the definition of v_{t} with $\sqrt{2}$, and in case which is different without $\sqrt{2}$, it's shown in the bracket. The energy in relation to temperature

$$\check{E}_{\mathrm{s}} = \frac{E_{\mathrm{s}}}{T_{\mathrm{s}}} \quad \left(\check{E}_{\mathrm{s}} = \frac{2E_{\mathrm{s}}}{T_{\mathrm{s}}} \right) \tag{78}$$

and in the velocity

$$\check{E}_{\mathrm{s}} = \check{v}_{\mathrm{s}}^2 \tag{79}$$

Is the energy variable normalized with respect to the temperature for each species, while $k_\perp$ is normalized in terms of a representative $v_{\mathrm{t}*}$ for which we choose $T = 1$? Then, the argument of the Bessel function can be understood:

$$\begin{aligned} \texttt{arg}_{\mathrm{s}} &= \frac{\sqrt{\hat{m}_{\mathrm{s}} \hat{T}_{\mathrm{s}}}}{Z_{\mathrm{s}}} \frac{\hat{k}_\perp \check{v}_{\perp \mathrm{s}}}{\hat{B}} \\ &= \frac{\sqrt{\hat{m}_{\mathrm{s}} \hat{T}_{\mathrm{s}}}}{Z_{\mathrm{s}}} \frac{k_\perp \rho_* v_{\perp \mathrm{s}}/v_{\mathrm{ts}}}{B/B_*} \\ &= \frac{\sqrt{m_{\mathrm{s}} T_{\mathrm{s}}}}{q_{\mathrm{s}}} \frac{q_*}{\sqrt{m_* T_*}} \frac{k_\perp v_{\perp \mathrm{s}}}{B/B_*} \frac{v_{\mathrm{t}*}}{\Omega_*} \sqrt{\frac{m_{\mathrm{s}}}{2T_{\mathrm{s}}}} \\ &= \frac{k_\perp v_{\perp \mathrm{s}}}{\Omega_{\mathrm{s}}} \end{aligned} \tag{80}$$

On the other hand, they cancel in the pitch-angle variable λ

$$\check{\lambda} = \frac{\check{v}_\perp^2}{\check{v}^2 \hat{B}} = \frac{\hat{v}_\perp^2}{\hat{v}^2 \hat{B}} = \hat{\lambda} \tag{81}$$

Here is a formula to transfer $\hat{v}$ to $\check{v}$:

$$\hat{v}_{\mathrm{s}} = \check{v}_{\mathrm{s}} \sqrt{\frac{\hat{T}_{\mathrm{s}}}{\hat{m}_{\mathrm{s}}}} \tag{82}$$

7 collisions

Rough sketch is given in Greg's and David's memo. Here is shown the expression of the function H_{ee} appearing in Greg's note.

Define the function

$$H_{ee}(E) = \frac{1}{\sqrt{\pi E}} e^{-E} + \left(1 - \frac{1}{2E}\right) \mathrm{erf}\left(\sqrt{E}\right) \tag{83}$$

with this erf(·) part given in an awful polynomial including 16-th power in `gs2`. I don't know the source of that formula.

For electrons,

$$\texttt{vnew} = \frac{\texttt{vnewk}}{\check{v}^3}(\texttt{zeff} + H_{ee}) * 0.5 * \texttt{tunits} \tag{84}$$

where `vnewk` and `zeff` are the input variables in `species_parameters` and `parameters` namelists, respectively. `zeff` term represents electron-ion collision under an approximation of replacing ion distribution function by a delta-function valid when $v_{\mathrm{th,i}} \ll v_{\mathrm{th,e}}$. H_{ee} term represents the like-particle collisions with a Maxwellian background. For ions, `zeff` term is omitted. When `const_v` flag in `collisions_knobs` is on, the whole `vnew` is evaluated for the thermal velocity ($\check{v} = 1$) for both electrons and ions. There is another array `vnew4` in `collisions.f90`, but it is unused.

8 Ascii output files

Everything turned on by the flag `write_ascii`.

`(runname).moments` Controlled by `write_final_moments` and each column means

$$\theta \quad k_y \quad k_x \quad \texttt{ntot} \quad \texttt{dens} \quad u_\parallel \quad T_\parallel \quad T_\perp \quad \theta - \theta_0 \quad \texttt{is}$$

middle 5 normalized by `phi0`

`(runname).mom2` Controlled by `write_final_moments` and each column means

$$\theta \quad k_y \quad k_x \quad \texttt{ntot} \quad \texttt{dens} \quad u_\parallel \quad T_\parallel \quad T_\perp \quad \theta - \theta_0 \quad \texttt{is}$$

`(runname).fields` Controlled by `write_final_fields` and each column means

$$\theta \quad k_y \quad k_x \quad \phi_\mathrm{r} \quad \phi_\mathrm{i} \quad A_{\|,\mathrm{r}} \quad A_{\|,\mathrm{i}} \quad A_{\perp,\mathrm{r}} \quad A_{\perp,\mathrm{i}} \quad \theta-\theta_0 \quad |\phi|$$

9 Netcdf file and `gs2.pro`

`phi2` in `(runname).out` is a volume average.

`phi0`, `apar0`, `aperp0` written out in the subroutine `nc_loop` are all 3D arrays including time at the slice with `ig` = `igomega` (θ = `igomega` $* 2\pi$) where `igomega` is an input variable in `gs2_diagnostics_knobs` namelist.

`phi` and `apar` are 3D arrays of the electrostatic field and parallel vector potential at the last timestep written out by the subroutine `nc_final_-fields`. Their arguments are (k_y, k_x, θ, ri) as seen in the `ncdump` command, but in `gs2` and IDL routine, they are accessed as (ri, θ, k_x, k_y).

Variables `md` and `nd` are valid number of modes after truncation by 2/3-rule in k_y and k_x direction, respectively. `malias`, and `nalias` are the full number of modes, or the number of grid points in y and x directions. Making the connection to `gs2` variables, we obtain the following correspondence:

$$\texttt{md} = \texttt{naky}, \quad \texttt{nd} = \texttt{ntheta0} \tag{85}$$

$$\texttt{malias} = \texttt{ny}, \quad \texttt{nalias} = \texttt{nx}, \tag{86}$$

where the left hand sides are the variables in `gs2.pro` and the rhs are those in `gs2`. `malias` and `nalias` were defined in `gs2.pro` as

$$\texttt{malias} = 3 * \texttt{md}, \quad \texttt{nalias} = 3 * \texttt{nd}/2 + 1, \tag{87}$$

but I changed them to the followings:

$$\texttt{malias} = (\texttt{md} - 1) * 3 + 1, \tag{88}$$

$$\texttt{malias} = \texttt{malias} + (\texttt{malias} \mod 2) \tag{89}$$

$$\texttt{nalias} = (\texttt{nd} - 1)/2 * 3 + 1 \tag{90}$$

$$\texttt{nalias} = \texttt{nalias} + (\texttt{nalias} \mod 2) \tag{91}$$

which are the exact inverse of the aliasing expressions found in `gs2` when `nx` and `ny` are exact powers of 2 and larger than 2.

Here is a list of changes I made on `gs2.pro`.

1. recovered exact number of grid points as explained above
2. added one more grid in both x and y directions to take care of the periodicity
3. added `phi`, `apar`, and `apar_1` in the 'Field Plot' section. They are the 2D real-space values of each quantity at the final step. So, if the run stops in the linear phase, they give the eigenfunctions. They only work with the axes of 'x,y', and `apar_1` is obtained by eliminating the equilibrium component out of `apar`. The value of θ is controlled by 'Active l' slidebar in the right.
4. added `phi`, `apar`, and `apar_1` in the 'Line Plot' section. They are the 1D real-space values of the above. The plane you slice in y is determined by 'Active M' slidebar.

10 Bug report

1. Tar ball `src.08.17.04.tgz` is broken. First `make distclean` before compilation: fixed.
2. At `DEPENDENCIES` section in `Makefile`, `ingen.o` must also depend on `text_options.o`, `constants.o`, and `theta_grid.o`: fixed
3. At `MODULE DECLARATIONS` section in `Makefile`, `LINKS` should include `file_utils.f90`: fixed. Also at `DIRECTIVES` section, there are multiple declaration of `file_utils.o` (maybe case dependent?), but is the second line needed in the first declaration? If needed, `file_utils.f90` may also need to be added in dependency.