Piggyback documentation

Modified Registry (Registry.EM_COMMON → Registry.EM_COMMON.PG)

New variables:

state	real	t_pg	ikjb	dyn_em	2	i0rhusdf=(bdy_interp:dt)	"t_pg"	"perturbation potential temperature (theta-t0) pg"	"K"
i1	real	t_tend_pg	ikj	dyn_em	1	-	-	-	-
i1	real	t_tendf_pg	ikj	dyn_em	1	-	-	-	-
i1	real	t_2save_pg	ikj	dyn_em	1	-	-	-	-
state	real	t_save_pg	ikj	dyn_em	1	-	-	"t_save piggyback"	-
state	real	t_base_pg	k	dyn_em	1	-	"t_base_pg"	"base state t in idealized cases pg"	"K"
state	real	qv_pg	ikjftb	scalar	1	i0rhusdf=(bdy_interp:dt)	"qvapor_pg"	"water vapor mixing ratio pg"	"kg kg-1"
state	real	qc_pg	ikjftb	scalar	1	i0rhusdf=(bdy_interp:dt)	"qcloud_pg"	"cloud water mixing ratio pg"	"kg kg-1"
state	real	qr_pg	ikjftb	scalar	1	i0rhusdf=(bdy_interp:dt)	"qrain_pg"	"rain water mixing ratio pg"	"kg kg-1"
state	real	qi_pg	ikjftb	scalar	1	i0rhusdf=(bdy_interp:dt)	"qice_pg"	"ice mixing ratio pg"	"kg kg-1"
state	real	qs_pg	ikjftb	scalar	1	i0rhusdf=(bdy_interp:dt)	"qsnow_pg"	"snow mixing ratio pg"	"kg kg-1"
state	real	qg_pg	ikjftb	scalar	1	i0rhusdf=(bdy_interp:dt)	"qgraup_pg"	"graupel mixing ratio pg"	"kg kg-1"

state	real	qh_pg	ikjftb	scalar	1	i0rhusdf=(bdy_interp:dt)	"qhail_pg"	"hail mixing ratio pg"	"kg kg-1"
state	real	qndrop_pg	ikjftb	scalar	1	i0rhusdf=(bdy_interp:dt)	"qndrop_pg"	"droplet number mixing ratio pg"	"# kg(-1)"
State	real	qni_pg	ikjftb	scalar	1	i0rhusdf=(bdy_interp:dt)	"qnice_pg"	"ice number concentration pg"	"# kg(-1)"
state	real	qt_pg	ikjftb	scalar	1	i0rhusdf=(bdy_interp:dt)	"cwm_pg"	"total condensate mixing ratio pg"	"kg kg-1"
State	Real	qns_pg	ikjftb	scalar	1	i0rhusdf=(bdy_interp:dt)	"qnsnow_pg"	"snow number concentration pg"	"# kg(-1)"
State	Real	qnr_pg	ikjftb	scalar	1	i0rhusdf=(bdy_interp:dt)	"QNRAIN_PG"	"Rain Number concentration pg"	"# kg(-1)"
State	Real	qng_pg	ikjftb	scalar	1	i0rhusdf=(bdy_interp:dt)	"QNGRAUPEL_PG"	"Graupel Number concentration pg"	"# kg(-1)"
State	Real	qnh_pg	ikjftb	scalar	1	i0rhusdf=(bdy_interp:dt)	"QNHAIL_PG"	"Hail Number concentration pg"	"# kg(-1)"
State	Real	qnn_pg	ikjftb	scalar	1	i0rhusdf=(bdy_interp:dt)	"QNCCN_PG"	"CCN Number concentration pg"	"# kg(-1)"
State	Real	qnc_pg	ikjftb	scalar	1	i0rhusdf=(bdy_interp:dt)	"QNCLOUD_PG"	"cloud water Number concentration pg"	"# kg(-1)"
State	Real	qvolg_pg	ikjftb	scalar	1	i0rhusdf=(bdy_interp:dt)	"QVGRAUPEL_PG"	"Graupel Particle Volume pg"	"m(3) kg(-1)"
State	Real	qvolh_pg	ikjftb	scalar	1	i0rhusdf=(bdy_interp:dt)	"QVHAIL_PG"	"Hail Particle Volume pg"	"m(3) kg(-1)"
State	Real	dfi_qv_pg	ikjftb	dfi_scalar	1	rusdf=(bdy_interp:dt)	"DFI_QVAPOR_PG"	"Water vapor mixing ratio pg"	"kg kg-1"
State	Real	dfi_qc_pg	ikjftb	dfi_scalar	1	rusdf=(bdy_interp:dt)	"DFI_QCLOUD_PG"	"Cloud water mixing ratio pg"	"kg kg-1"
State	Real	dfi_qr_pg	ikjftb	dfi_scalar	1	rusdf=(bdy_interp:dt)	"DFI_QRAIN_PG"	"Rain water mixing ratio pg"	"kg kg-1"
State	Real	dfi_qi_pg	ikjftb	dfi_scalar	1	rusdf=(bdy_interp:dt)	"DFI_QICE_PG"	"Ice mixing ratio pg"	"kg kg-1"

State	Real	dfi_qs_pg	ikjftb	dfi_scalar	1	rusdf=(bdy_interp:dt)	"DFI_QSNOW_PG"	"Snow mixing ratio pg"	"kg kg-1"
State	Real	dfi_qg_pg	ikjftb	dfi_scalar	1	rusdf=(bdy_interp:dt)	"DFI_QGRAUP_PG"	"Graupel mixing ratio pg"	"kg kg-1"
State	Real	dfi_qh_pg	ikjftb	dfi_scalar	1	rusdf=(bdy_interp:dt)	"DFI_QHAIL_PG"	"Hail mixing ratio pg"	"kg kg-1"
State	Real	dfi_qndrop_pg	ikjftb	dfi_scalar	1	rusdf=(bdy_interp:dt)	"DFI_QNDROP_PG"	"Droplet number mixing ratio pg"	"# kg-1"
State	Real	dfi_qni_pg	ikjftb	dfi_scalar	1	rusdf=(bdy_interp:dt)	"DFI_QNICE_PG"	"Ice Number concentration pg"	"# kg-1"
State	Real	dfi_qt_pg	ikjftb	dfi_scalar	1	rusdf=(bdy_interp:dt)	"DFI_CWM_PG"	"Total condensate mixing ratio pg"	"kg kg-1"
State	Real	dfi_qns_pg	ikjftb	dfi_scalar	1	rusdf=(bdy_interp:dt)	"DFI_QNSNOW_PG"	"Snow Number concentration pg"	"# kg-1"
State	Real	dfi_qnr_pg	ikjftb	dfi_scalar	1	rusdf=(bdy_interp:dt)	"DFI_QNRAIN_PG"	"Rain Number concentration pg"	"# kg-1"
State	Real	dfi_qng_pg	ikjftb	dfi_scalar	1	rusdf=(bdy_interp:dt)	"DFI_QNGRAUPEL_PG"	"Graupel Number concentration pg"	"# kg-1"
State	Real	dfi_qnh_pg	ikjftb	dfi_scalar	1	rusdf=(bdy_interp:dt)	"DFI_QNHAIL_PG"	"Hail Number concentration pg"	"# kg-1"
State	Real	dfi_qnn_pg	ikjftb	dfi_scalar	1	rusdf=(bdy_interp:dt)	"DFI_QNCC_PG"	"CNN Number concentration pg"	"# kg-1"
State	Real	dfi_qnc_pg	ikjftb	dfi_scalar	1	rusdf=(bdy_interp:dt)	"DFI_QNCLOUD_PG"	"Cloud Number concentration pg"	
State	Real	dfi_qvolg_pg	ikjftb	dfi_scalar	1	rusdf=(bdy_interp:dt)	"DFI_QVGRAUPEL_PG"	"Graupel Particle Volume pg"	"m(3) kg(-1)"
State	Real	dfi_qvolh_pg	ikjftb	dfi_scalar	1	rusdf=(bdy_interp:dt)	"DFI_QVHAIL_PG"	"Hail Particle Volume pg"	"m(3) kg(-1)"
State	Real	dfi_tten_rad_pg	ikj	dfi_ misc	1	ir	"RAD_TTEN_DFI_PG"	"Radar Pot. Temp. Tendency Pg"	"K s-1"
State	Real	h_diabatic_pg	ikj	misc	1	rdu	"h_diabatic_pg"	"Microphysics Latent Heating PG"	"K s-1"

State	Real	qv_base_pg	k	misc	1	ir	"qv_base_pg"	"Base State Qv In Idealized Cases Pg"	""
State	Real	refl_10cm_pg	ikj	dyn_em	1	rhdu	"refl_10cm pg"	"Radar Reflectivity (Lamda = 10 Cm) PG"	"dBZ"
i1	real	th_phy_pg	ikj	misc	1	-			
state	real	t_phy_pg	ikj	misc	1	r	"T_PHY PG"	"Temperature Pg"	"K"
State	Real	dfi_QVG_PG	ij	misc	1	r	"QVG_dfi_PG"	"Water Vapor Mixing Ratio At The Surface PG"	"kg kg-1"
State	Real	RAINCPG	ij	misc	1	rhdu	"RAINCPG"	"Accumulated Total Cumulus Precipitation Pg"	"mm"
State	Real	RAINNCPG	ij	misc	1	rhdu	"RAINNCPG"	"Accumulated Total Grid Scale Precipitation Pg"	"mm"
State	Real	RAINSHPG	ij	misc	1	rhdu	"RAISHPG"	"Accumulated Shallow Cumulus Precipitation Pg"	"mm"
State	Real	SNOWNCPG	ij	misc	1	rhdu	"SNOWNCPG"	"Accumulated Total Grid Scale Snow And Ice Pg"	"mm"
State	Real	GRAUPELNCPG	ij	misc	1	rhdu	"GRAUPELNCPG"	"Accumulated Total Grid Scale Graupel Pg"	"mm"
State	Real	HAILNCPG	ij	misc	1	rhdu	"HAILNCPG"	"Accumulated Total Grid Scale Hail Pg"	"mm"
State	Real	RTHFTENPG	ikj	misc	1	r	"RTHFTEN PG"	"Total Advective Potential Temperature Tendency Pg"	"K s-1"
State	Real	RQVFTENPG	ikj	misc	1	r	"RQVFTEN PG"	"Total Advective Moisture Tendency Pg"	"kg kg-1 s-1"

Microphysics package:

package: thompson; mp_physics==8; moist:qv,qc,qr,qi,qs,qg;scalar:qni,qnr,qv_pg,qc_pg,qr_pg,qi_pg,qs_pg,qg_pg,qni_pg,qnr_pg

package; thompson_dfi; mp_physics_dfi==8; dfi_moist:dfi_qv,dfi_qc,dfi_qr,dfi_qi,dfi_qs,dfi_qg;dfi_scalar:dfi_qni,dfi_qnr,dfi_qv_pg, dfi_qc_pg,dfi_qr_pg,dfi_qi_pg,dfi_qs_pg,dfi_qg_pg,dfi_qni_pg,dfi_qnr_pg

Modification in HALO files:

halo: HALO_EM_INIT_2; dyn_em 48: t_1,t_2,mu_1,mu_2,tke_1,tke_2,ww,phb,t_pg_1,t_pg_2

halo: HALO_EM_INIT_4; dyn_em 48: pb,h_diabatic,msftx,msfty,msfux,msfuy,msfvx,msfvy,msfvx_inv,f,e,sina,cosa,ht,potevp, snopcx,soiltb,xlat,xlong,xlat_u,xlat_v,xlong_u,xlong_v,clat,h_diabatic_pg

halo: HALO_EM_B; dyn_em 4: ph_2,al,p,t_1,t_save,u_save,v_save,mu_1,mu_2,mudf,php,alt,pb,t_pg_1,t_save_pg

halo: HALO_EM_D2_3; dyn_em 24: u_2,v_2,w_2,t_2,ph_2,t_pg_2;24:moist,chem,tracer,scalar;4:mu_2,al

halo: HALO_EM_D2_5; dyn_em 48: u_2,v_2,w_2,t_2,ph_2,t_pg_2;24:moist,chem,tracer,scalar;4:mu_2,al

halo: HALO_EM_D3_3; dyn_em 24: u_1,u_2,v_1,v_2,w_1,w_2,t_1,t_2,ph_1,ph_2,tke_1,tke_2,moist,chem,tracer,scalar,t_pg_1, t_pg_2;4:mu_1,mu_2

halo: HALO_EM_D3_5; dyn_em 48: u_1,u_2,v_1,v_2,w_1,w_2,t_1,t_2,ph_1,ph_2,tke_1,tke_2,moist,chem,tracer,scalar,t_pg_1, t_pg_2;4:mu_1,mu_2

halo: HALO_EM_E_3; dyn_em 24: u_1,u_2,v_1,v_2,w_1,w_2,t_1,t_2,ph_1,ph_2,tke_1,tke_2,t_pg_1,t_pg_2;4:mu_1,mu_2

halo: HALO_EM_E_5; dyn_em 48: u_1,u_2,v_1,v_2,w_1,w_2,t_1,t_2,ph_1,ph_2,tke_1,tke_2,t_pg_1,t_pg_2;4:mu_1,mu_2

halo: HALO_CUP_G3_IN; dyn_em 24: RTHFTEN,RQVFTEN,w_2,t_phy,t_phy_pg,RTHFTENPG,RQVFTENPG

halo: HALO_EM_COUPLE_B; dyn_em 48: ph_1,ph_2,w_1,w_2,t_1,t_2,u_1,u_2,v_1,v_2,moist,chem,tracer,scalar,t_pg_1,t_pg_2

period: PERIOD_EM_COUPLE_B; dyn_em 3: ph_1,ph_2,w_1,w_2,t_1,t_2,u_1,u_2,v_1,v_2,moist,chem,tracer,scalar,t_pg_1,t_pg_2

For moving nests:

#halo: em_shift_halo_y; dyn_em 48: imask_nostag,imask_xstag,imask_ystag,imask_xystag,u_2,v_2,t_2,t_pg_2

#halo: em_shift_halo_x; dyn_em 48: imask_nostag,imask_xstag,imask_ystag,imask_xystag,u_2,v_2,t_2,t_pg_2

Periodic Boundary Communications

period: PERIOD_BDY_EM_INIT; dyn_em 3: u_1,u_2,v_1,v_2,w_1,w_2,t_1,t_2,ph_1,ph_2,t_init,phb,ph0,php,pb,al,alt,alb,mu_1,mu_2,mub,mu0,ht,msftx,msfty,msfux,msfuy,msfvx,msfvy,msfvx_inv,sina,cosa,e,f,t_pg_1,t_pg_2

period: PERIOD_BDY_EM_B; dyn_em 2: ru_tend,rv_tend,ph_2,al,p,t_1,t_save,u_save,v_save,mu_1,mu_2,mudf,php,alt,pb,t_pg_1,t_save_pg

period: PERIOD_BDY_EM_C; dyn_em 2: u_2,u_save,v_2,v_save,t_2,t_save,muv,msfvx,msfvy,muu,msfux,msfuy,msfvx_inv,t_pg_2,t_save_pg

period: PERIOD_BDY_EM_D; dyn_em 3: u_2,v_2,w_2,t_2,ph_2,mu_2,tke_2,t_pg_2

period: PERIOD_BDY_EM_D3; dyn_em 3: u_1,u_2,v_1,v_2,w_1,w_2,t_1,t_2,ph_1,ph_2,tke_1,tke_2,mu_1,mu_2,t_pg_1,t_pg_2

#swap: SWAP_ETAMP_NEW; dyn_em 1: dz8w,p_phy,pi_phy,rho,th_phy,moist,F_ICE_PHY,F_RAIN_PHY,F_RIMEF_PHY,RAINNC,RAINNCV,SR,LOWLYR,th_phy_pg,RAINNCPG

#swap: SWAP_WSM3; dyn_em 1: th_phy,moist,w_2,rho,pi_phy,p_phy,dz8w,rainnc,rainncv,th_phy_pg,rainncpg

Modified modules:

1. start_em.F → start_em.F.PG

Line 464: grid%t_pg_1(i,k,j)=grid%t_2(i,k,j)

Line 1090: CALL set_physical_bc3d(grid%t_pg_1 , 't' , config_flags , &
ids , ide , jds , jde , kds , kde , &
ims , ime , jms , jme , kms , kme , &
its , ite , jts , jte , kts , kte , &
its , ite , jts , jte , kts , kte)

2. solve_em.F → solve_em.F.PG

Line 415: CALL zero_bdytend (grid%u_btxs,grid%u_btxe,grid%u_btys,grid%u_btye, &
grid%v_btxs,grid%v_btxe,grid%v_btys,grid%v_btye, &
grid%ph_btxs,grid%ph_btxe,grid%ph_btys,grid%ph_btye, &
grid%t_btxs,grid%t_btxe,grid%t_btys,grid%t_btye, &
grid%t_pg_btxs,grid%t_pg_btxe,grid%t_pg_btys,grid%t_pg_btye, &
grid%w_btxs,grid%w_btxe,grid%w_btys,grid%w_btye, &
grid%mu_btxs,grid%mu_btxe,grid%mu_btys,grid%mu_btye, &
moist_btxs,moist_btxe, &
moist_btys,moist_btye, &
grid%spec_bdy_width,num_3d_m, &
ids,ide, jds,jde, kds,kde, &
ims,ime, jms,jme, kms,kme, &
ips,ipe, jps,jpe, kps,kpe, &
grid%i_start(ij), grid%i_end(ij), &
grid%j_start(ij), grid%j_end(ij), &
k_start, k_end)

Line 538: CALL rk_step_prep (config_flags, rk_step, &
grid%u_2, grid%v_2, grid%w_2, grid%t_2, **grid%t_pg_2**, grid%ph_2, grid%mu_2, moist, &
grid%ru, grid%rv, grid%rw, grid%ww, grid%php, grid%alt, grid%muu, grid%muv, &
grid%mub, grid%mut, grid%phb, grid%pb, grid%p, grid%al, grid%alb, &
cqu, cqv, cqw, &
grid%msfux, grid%msfuy, grid%msfvx, grid%msfvx_inv, &
grid%msfvy, grid%msftx, grid%msfty, &
grid%fnm, grid%fnp, grid%dnw, grid%rdx, grid%rdy, &
num_3d_m, &
ids, ide, jds, jde, kds, kde, &
ims, ime, jms, jme, kms, kme, &

grid%i_start(ij), grid%i_end(ij), &
grid%j_start(ij), grid%j_end(ij), &
k_start, k_end)

Line 683: CALL first_rk_step_part1 (grid, config_flags &
, moist , moist_tend &
, chem , chem_tend &
, tracer, tracer_tend &
, scalar , scalar_tend &
, fdda3d, fdda2d &
, ru_tendf, rv_tendf &
, rw_tendf, t_tendf, **t_tendf_pg** &
, ph_tendf, mu_tendf &
, tke_tend &
, config_flags%use_adaptive_time_step &
, curr_secs &
, psim , psih , wspd , gz1oz0 &
, br , chklowq &
, cu_act_flag , hol , th_phy, **th_phy_pg** &
, pi_phy , p_phy , grid%t_phy, **grid%t_phy_pg** &
, u_phy , v_phy &
, dz8w , p8w , t8w , rho_phy , rho &
, ids, ide, jds, jde, kds, kde &
, ims, ime, jms, jme, kms, kme &
, ips, ipe, jps, jpe, kps, kpe &
, imsx, imex, jmsx, jmex, kmsx, kmex &
, ipsx, ipex, jpsx, jpex, kpsx, kpex &
, imsy, imey, jmsy, jmey, kmsy, kmey &
, ipsy, ipey, jpsy, jpey, kpsy, kpey &
, k_start , k_end &
, f_flux)

Line 750: CALL first_rk_step_part2 (grid, config_flags &
, moist , moist_tend &
, chem , chem_tend &
, tracer, tracer_tend &
, scalar , scalar_tend &
, fdda3d, fdda2d &
, ru_tendf, rv_tendf &
, rw_tendf, t_tendf, **t_tendf_pg** &
, ph_tendf, mu_tendf &
, tke_tend &
, adapt_step_flag , curr_secs &
, psim , psih , wspd , gz1oz0 &
, br , chklowq &
, cu_act_flag , hol , th_phy, **th_phy_pg** &
, pi_phy , p_phy , grid%t_phy, **grid%t_phy_pg** &
, u_phy , v_phy &
, dz8w , p8w , t8w , rho_phy , rho &
, nba_mij, num_nba_mij & !JDM
, nba_rij, num_nba_rij & !JDM
, ids, ide, jds, jde, kds, kde &
, ims, ime, jms, jme, kms, kme &
, ips, ipe, jps, jpe, kps, kpe &
, imsx, imex, jmsx, jmex, kmsx, kmex &
, ipsx, ipex, jpsx, jpex, kpsx, kpex &
, imsy, imey, jmsy, jmey, kmsy, kmey &
, ipsy, ipey, jpsy, jpey, kpsy, kpey &
, k_start , k_end)

Line 810: CALL rk_tendency (config_flags, rk_step &
,grid%ru_tend, grid%rv_tend, rw_tend, ph_tend, t_tend, **t_tend_pg** &

,ru_tendf, rv_tendf, rw_tendf, ph_tendf, t_tendf, **t_tendf_pg** &
,mu_tend, grid%u_save, grid%v_save, w_save, ph_save &
,grid%t_save, **grid%t_save_pg**, mu_save, grid%rthften, **grid%rthftenpg** &
,grid%ru, grid%rv, grid%rw, grid%ww &
,grid%u_2, grid%v_2, grid%w_2, grid%t_2, **grid%t_pg_2**, grid%ph_2 &
,grid%u_1, grid%v_1, grid%w_1, grid%t_1, **grid%t_pg_1**, grid%ph_1 &
,grid%h_diabatic, **grid%h_diabatic_pg**, grid%phb, grid%t_init &
,grid%mu_2, grid%mut, grid%muu, grid%muv, grid%mub &
,grid%al, grid%alt, grid%p, grid%pb, grid%php, cqu, cqv, cqw &
,grid%u_base, grid%v_base, grid%t_base, **grid%t_base_pg**, grid%qv_base, **grid%qv_base_pg**, grid%z_base &
,grid%msfux,grid%msfuy, grid%msfvx, grid%msfvx_inv &
,grid%msfvy, grid%msftx,grid%msfty, grid%clat, grid%f, grid%e, grid%sina, grid%cosa &
,grid%fnm, grid%fnp, grid%rdn, grid%rdnw &
,grid%dt, grid%rdx, grid%rdy, grid%khdif, grid%kvdif, grid%xkmh, grid%xkhh &
,grid%diff_6th_opt, grid%diff_6th_factor &
,config_flags%momentum_adv_opt &
,grid%dampcoef,grid%zdamp,config_flags%damp_opt,config_flags%rad_nudge &
,grid%cf1, grid%cf2, grid%cf3, grid%cfn, grid%cfn1, num_3d_m &
,config_flags%non_hydrostatic, config_flags%top_lid &
,grid%u_frame, grid%v_frame &
,ids, ide, jds, jde, kds, kde &
,ims, ime, jms, jme, kms, kme &
,grid%i_start(ij), grid%i_end(ij) &
,grid%j_start(ij), grid%j_end(ij) &
,k_start, k_end &
,max_vert_cfl_tmp(ij), max_horiz_cfl_tmp(ij) &
,dz8w,grid%xtime) !-- Temporary add by G. Thompson

Line 868: CALL relax_bdy_dry (config_flags, &

grid%u_save, grid%v_save, ph_save, grid%t_save, **grid%t_save_pg**, &
w_save, mu_tend, &
grid%ru, grid%rv, grid%ph_2, grid%t_2, **grid%t_pg_2**, &
grid%w_2, grid%mu_2, grid%mut, &
grid%u_bxs,grid%u_bxe,grid%u_bys,grid%u_bye, &
grid%v_bxs,grid%v_bxe,grid%v_bys,grid%v_bye, &
grid%ph_bxs,grid%ph_bxe,grid%ph_bys,grid%ph_bye, &
grid%t_bxs,grid%t_bxe,grid%t_bys,grid%t_bye, &
grid%t_pg_bxs,grid%t_pg_bxe,grid%t_pg_bys,grid%t_pg_bye, &
grid%w_bxs,grid%w_bxe,grid%w_bys,grid%w_bye, &
grid%mu_bxs,grid%mu_bxe,grid%mu_bys,grid%mu_bye, &
grid%u_btxs,grid%u_btxe,grid%u_btys,grid%u_btye, &
grid%v_btxs,grid%v_btxe,grid%v_btys,grid%v_btye, &
grid%ph_btxs,grid%ph_btxe,grid%ph_btys,grid%ph_btye, &
grid%t_btxs,grid%t_btxe,grid%t_btys,grid%t_btye, &
grid%t_pg_btxs,grid%t_pg_btxe,grid%t_pg_btys,grid%t_pg_btye, &
grid%w_btxs,grid%w_btxe,grid%w_btys,grid%w_btye, &
grid%mu_btxs,grid%mu_btxe,grid%mu_btys,grid%mu_btye, &
config_flags%spec_bdy_width, grid%spec_zone, grid%relax_zone, &
grid%dtbc, grid%fcx, grid%gcx, &
ids,ide, jds,jde, kds,kde, &
ims,ime, jms,jme, kms,kme, &
ips,ipe, jps,jpe, kps,kpe, &
grid%i_start(ij), grid%i_end(ij), &
grid%j_start(ij), grid%j_end(ij), &
k_start, k_end)

Line 896: CALL rk_addtend_dry(grid%ru_tend, grid%rv_tend, rw_tend, ph_tend, t_tend, **t_tend_pg**, &
ru_tendf, rv_tendf, rw_tendf, ph_tendf, t_tendf, **t_tendf_pg**, &
grid%u_save, grid%v_save, w_save, ph_save, grid%t_save, **grid%t_save_pg**,&

mu_tend, mu_tendf, rk_step, &
grid%h_diabatic, **grid%h_diabatic_pg**, grid%mut, grid%msftx, &
grid%msfty, grid%msfux,grid%msfuy, &
grid%msfvx, grid%msfvx_inv, grid%msfvy, &
ids,ide, jds,jde, kds,kde, &
ims,ime, jms,jme, kms,kme, &
ips,ipe, jps,jpe, kps,kpe, &
grid%i_start(ij), grid%i_end(ij), &
grid%j_start(ij), grid%j_end(ij), &
k_start, k_end)

Line 912: CALL spec_bdy_dry (config_flags, &
grid%ru_tend, grid%rv_tend, ph_tend, t_tend, **t_tend_pg**, &
rw_tend, mu_tend, &
grid%u_bxs,grid%u_bxe,grid%u_bys,grid%u_bye, &
grid%v_bxs,grid%v_bxe,grid%v_bys,grid%v_bye, &
grid%ph_bxs,grid%ph_bxe,grid%ph_bys,grid%ph_bye, &
grid%t_bxs,grid%t_bxe,grid%t_bys,grid%t_bye, &
grid%t_pg_bxs,grid%t_pg_bxe,grid%t_pg_bys,grid%t_pg_bye, &
grid%w_bxs,grid%w_bxe,grid%w_bys,grid%w_bye, &
grid%mu_bxs,grid%mu_bxe,grid%mu_bys,grid%mu_bye, &
grid%u_btxs,grid%u_btxe,grid%u_btys,grid%u_btye, &
grid%v_btxs,grid%v_btxe,grid%v_btys,grid%v_btye, &
grid%ph_btxs,grid%ph_btxe,grid%ph_btys,grid%ph_btye, &
grid%t_btxs,grid%t_btxe,grid%t_btys,grid%t_btye, &
grid%t_pg_btxs,grid%t_pg_btxe,grid%t_pg_btys,grid%t_pg_btye, &
grid%w_btxs,grid%w_btxe,grid%w_btys,grid%w_btye, &
grid%mu_btxs,grid%mu_btxe,grid%mu_btys,grid%mu_btye, &
config_flags%spec_bdy_width, grid%spec_zone, &
ids,ide, jds,jde, kds,kde, & ! domain dims
ims,ime, jms,jme, kms,kme, & ! memory dims

```
                ips,ipe, jps,jpe, kps,kpe,  & ! patch  dims
                grid%i_start(ij), grid%i_end(ij),          &
                grid%j_start(ij), grid%j_end(ij),          &
                k_start, k_end                             )
```

Line 979:

```
CALL small_step_prep(grid%u_1,grid%u_2,grid%v_1,grid%v_2,grid%w_1,grid%w_2,   &
                grid%t_1, grid%t_pg_1, grid%t_2, grid%t_pg_2, grid%ph_1,grid%ph_2,         &
                grid%mub, grid%mu_1, grid%mu_2,                    &
                grid%muu, muus, grid%muv, muvs,                    &
                grid%mut, grid%muts, grid%mudf,                    &
                grid%u_save, grid%v_save, w_save,                  &
                grid%t_save, grid%t_save_pg, ph_save, mu_save,                 &
                grid%ww, ww1,                                      &
                grid%dnw, c2a, grid%pb, grid%p, grid%alt,          &
                grid%msfux,grid%msfuy, grid%msfvx, grid%msfvx_inv,      &
                grid%msfvy, grid%msftx,grid%msfty,                 &
                grid%rdx, grid%rdy, rk_step,                       &
                ids, ide, jds, jde, kds, kde,                      &
                ims, ime, jms, jme, kms, kme,                      &
                grid%i_start(ij), grid%i_end(ij),                  &
                grid%j_start(ij), grid%j_end(ij),                  &
                k_start    , k_end                                 )
```

Line 1116:

```
CALL set_physical_bc3d( grid%t_pg_1, 'p', config_flags,         &
                  ids, ide, jds, jde, kds, kde,         &
                  ims, ime, jms, jme, kms, kme,         &
                  ips, ipe, jps, jpe, kps, kpe,         &
                  grid%i_start(ij), grid%i_end(ij),     &
                  grid%j_start(ij), grid%j_end(ij),     &
                  k_start    , k_end                    )
```

Line 1132: CALL set_physical_bc3d(**grid%t_save_pg**, 't', config_flags, &
ids, ide, jds, jde, kds, kde, &
ims, ime, jms, jme, kms, kme, &
ips, ipe, jps, jpe, kps, kpe, &
grid%i_start(ij), grid%i_end(ij), &
grid%j_start(ij), grid%j_end(ij), &
k_start , k_end)

Line 1290: CALL advance_mu_t(grid%ww, ww1, grid%u_2, grid%u_save, grid%v_2, grid%v_save, &
grid%mu_2, grid%mut, muave, grid%muts, grid%muu, grid%muv, &
grid%mudf, grid%ru_m, grid%rv_m, grid%ww_m, &
grid%t_2, **grid%t_pg_2**, grid%t_save, **grid%t_save_pg**, t_2save, **t_2save_pg**, t_tend,&
t_tend_pg, mu_tend, &
grid%rdx, grid%rdy, dts_rk, grid%epssm, &
grid%dnw, grid%fnm, grid%fnp, grid%rdnw, &
grid%msfux,grid%msfuy, grid%msfvx, grid%msfvx_inv, &
grid%msfvy, grid%msftx,grid%msfty, &
iteration, config_flags, &
ids, ide, jds, jde, kds, kde, &
ims, ime, jms, jme, kms, kme, &
grid%i_start(ij), grid%i_end(ij), &
grid%j_start(ij), grid%j_end(ij), &
k_start , k_end)

Line 1361: CALL spec_bdyupdate(**grid%t_pg_2, t_tend_pg**, dts_rk, &
't' , config_flags, &
grid%spec_zone, &
ids,ide, jds,jde, kds,kde, &
ims,ime, jms,jme, kms,kme, &

ips,ipe, jps,jpe, kps,kpe, &
grid%i_start(ij), grid%i_end(ij),&
grid%j_start(ij), grid%j_end(ij),&
k_start , k_end)

Line 1403: CALL advance_w(grid%w_2, rw_tend, grid%ww, w_save, &
grid%u_2, grid%v_2, &
grid%mu_2, grid%mut, muave, grid%muts, &
t_2save, **t_2save_pg**, grid%t_2, **grid%t_pg_2**, grid%t_save, **grid%t_save_pg**, &
grid%ph_2, ph_save, grid%phb, ph_tend, &
grid%ht, c2a, cqw, grid%alt, grid%alb, &
a, alpha, gamma, &
grid%rdx, grid%rdy, dts_rk, t0, grid%epssm, &
grid%dnw, grid%fnm, grid%fnp, grid%rdnw, &
grid%rdn, grid%cf1, grid%cf2, grid%cf3, &
grid%msftx, grid%msfty, &
config_flags, config_flags%top_lid, &
ids,ide, jds,jde, kds,kde, &
ims,ime, jms,jme, kms,kme, &
grid%i_start(ij), grid%i_end(ij), &
grid%j_start(ij), grid%j_end(ij), &
k_start , k_end)

Line 1634: CALL small_step_finish(grid%u_2, grid%u_1, grid%v_2, grid%v_1, grid%w_2, grid%w_1, &
grid%t_2, **grid%t_pg_2**, grid%t_1, **grid%t_pg_1**, grid%ph_2, grid%ph_1, grid%ww, ww1, &
grid%mu_2, grid%mu_1, &
grid%mut, grid%muts, grid%muu, muus, grid%muv, muvs, &
grid%u_save, grid%v_save, w_save, &
grid%t_save, **grid%t_save_pg**, ph_save, mu_save, &
grid%msfux,grid%msfuy, grid%msfvx,grid%msfvy, grid%msftx,grid%msfty, &

grid%h_diabatic, **grid%h_diabatic_pg**, &
number_of_small_timesteps,dts_rk, &
rk_step, rk_order, &
ids, ide, jds, jde, kds, kde, &
ims, ime, jms, jme, kms, kme, &
grid%i_start(ij), grid%i_end(ij), &
grid%j_start(ij), grid%j_end(ij), &
k_start , k_end)

Line 2104: CALL rk_scalar_tend (im, im, config_flags, tenddec, &
rk_step, dt_rk, &
grid%ru_m, grid%rv_m, grid%ww_m, &
grid%muts, grid%mub, grid%mu_1, &
grid%alt, &
moist_old(ims,kms,jms,im), &
moist(ims,kms,jms,im), &
moist_tend(ims,kms,jms,im), &
advect_tend,h_tendency,z_tendency,grid%rqvften,**grid%rqvftenpg**, &
grid%qv_base, **grid%qv_base_pg**, .true., grid%fnm, grid%fnp, &
grid%msfux,grid%msfuy, grid%msfvx, grid%msfvx_inv,&
grid%msfvy, grid%msftx,grid%msfty, &
grid%rdx, grid%rdy, grid%rdn, grid%rdnw, grid%khdif, &
grid%kvdif, grid%xkhh, &
grid%diff_6th_opt, grid%diff_6th_factor, &
config_flags%moist_adv_opt, &
ids, ide, jds, jde, kds, kde, &
ims, ime, jms, jme, kms, kme, &
grid%i_start(ij), grid%i_end(ij), &
grid%j_start(ij), grid%j_end(ij), &
k_start , k_end)

Line 2235: CALL rk_scalar_tend (1, 1, config_flags, tenddec, &
rk_step, dt_rk, &
grid%ru_m, grid%rv_m, grid%ww_m, &
grid%muts, grid%mub, grid%mu_1, &
grid%alt, &
grid%tke_1, &
grid%tke_2, &
tke_tend(ims,kms,jms), &
advect_tend,h_tendency,z_tendency,grid%rqvften, **grid%rqvftenpg**, &
grid%qv_base, **grid%qv_base_pg**, .false., grid%fnm, grid%fnp, &
grid%msfux,grid%msfuy, grid%msfvx, grid%msfvx_inv, &
grid%msfvy, grid%msftx,grid%msfty, &
grid%rdx, grid%rdy, grid%rdn, grid%rdnw, grid%khdif, &
grid%kvdif, grid%xkhh, &
grid%diff_6th_opt, grid%diff_6th_factor, &
config_flags%tke_adv_opt, &
ids, ide, jds, jde, kds, kde, &
ims, ime, jms, jme, kms, kme, &
grid%i_start(ij), grid%i_end(ij), &
grid%j_start(ij), grid%j_end(ij), &
k_start , k_end)

Line 2322: CALL rk_scalar_tend (ic, ic, config_flags, tenddec, &
rk_step, dt_rk, &
grid%ru_m, grid%rv_m, grid%ww_m, &
grid%muts, grid%mub, grid%mu_1, &
grid%alt, &
chem_old(ims,kms,jms,ic), &
chem(ims,kms,jms,ic), &
chem_tend(ims,kms,jms,ic), &
advect_tend,h_tendency,z_tendency,grid%rqvften,**grid%rqvftenpg**, &

grid%qv_base, **grid%qv_base_pg**, .false., grid%fnm, grid%fnp, &
grid%msfux,grid%msfuy, grid%msfvx, grid%msfvx_inv, &
grid%msfvy, grid%msftx,grid%msfty, &
grid%rdx, grid%rdy, grid%rdn, grid%rdnw, &
grid%khdif, grid%kvdif, grid%xkhh, &
grid%diff_6th_opt, grid%diff_6th_factor, &
config_flags%chem_adv_opt, &
ids, ide, jds, jde, kds, kde, &
ims, ime, jms, jme, kms, kme, &
grid%i_start(ij), grid%i_end(ij), &
grid%j_start(ij), grid%j_end(ij), &
k_start , k_end)

Line 2448: CALL rk_scalar_tend (ic, ic, config_flags, tenddec, &
rk_step, dt_rk, &
grid%ru_m, grid%rv_m, grid%ww_m, &
grid%muts, grid%mub, grid%mu_1, &
grid%alt, &
tracer_old(ims,kms,jms,ic), &
tracer(ims,kms,jms,ic), &
tracer_tend(ims,kms,jms,ic), &
advect_tend,h_tendency,z_tendency,grid%rqvften, **grid%rqvftenpg**, &
grid%qv_base, **grid%qv_base_pg**, .false., grid%fnm, grid%fnp, &
grid%msfux,grid%msfuy, grid%msfvx, grid%msfvx_inv, &
grid%msfvy, grid%msftx,grid%msfty, &
grid%rdx, grid%rdy, grid%rdn, grid%rdnw, &
grid%khdif, grid%kvdif, grid%xkhh, &
grid%diff_6th_opt, grid%diff_6th_factor, &
config_flags%tracer_adv_opt, &
ids, ide, jds, jde, kds, kde, &
ims, ime, jms, jme, kms, kme, &

grid%i_start(ij), grid%i_end(ij), &
grid%j_start(ij), grid%j_end(ij), &
k_start , k_end)

Line 2583: CALL rk_scalar_tend (is, is, config_flags, tenddec, &
rk_step, dt_rk, &
grid%ru_m, grid%rv_m, grid%ww_m, &
grid%muts, grid%mub, grid%mu_1, &
grid%alt, &
scalar_old(ims,kms,jms,is), &
scalar(ims,kms,jms,is), &
scalar_tend(ims,kms,jms,is), &
advect_tend,h_tendency,z_tendency,grid%rqvften, **grid%rqvftenpg**, &
grid%qv_base, **grid%qv_base_pg**, .false., grid%fnm, grid%fnp, &
grid%msfux,grid%msfuy, grid%msfvx, grid%msfvx_inv, &
grid%msfvy, grid%msftx,grid%msfty, &
grid%rdx, grid%rdy, grid%rdn, grid%rdnw, &
grid%khdif, grid%kvdif, grid%xkhh, &
grid%diff_6th_opt, grid%diff_6th_factor, &
config_flags%scalar_adv_opt, &
ids, ide, jds, jde, kds, kde, &
ims, ime, jms, jme, kms, kme, &
grid%i_start(ij), grid%i_end(ij), &
grid%j_start(ij), grid%j_end(ij), &
k_start , k_end)

Line 2945: CALL rk_phys_bc_dry_2(config_flags, &
grid%u_2, grid%v_2, grid%w_2, &
grid%t_2, **grid%t_pg_2**, grid%ph_2, grid%mu_2, &
ids, ide, jds, jde, kds, kde, &
ims, ime, jms, jme, kms, kme, &

ips, ipe, jps, jpe, kps, kpe, &
grid%i_start(ij), grid%i_end(ij), &
grid%j_start(ij), grid%j_end(ij), &
k_start , k_end)

Line 3277: CALL moist_physics_prep_em(&
grid%t_2, **grid%t_pg_2**, grid%t_1, **grid%t_pg_1**, t0, rho, &
grid%al, grid%alb, grid%p, p8w, p0, grid%pb, &
grid%ph_2, grid%phb, th_phy, **th_phy_pg**, pi_phy, p_phy, &
grid%z, grid%z_at_w, dz8w, &
dtm, grid%h_diabatic, **grid%h_diabatic_pg**, &
config_flags,grid%fnm, grid%fnp, &
ids, ide, jds, jde, kds, kde, &
ims, ime, jms, jme, kms, kme, &
its, ite, jts, jte, &
k_start , k_end)

Line 3436: **diag_piggy = .TRUE.**

Line 3445: second call of → CALL microphysics_driver(….)

Line 3615: CALL moist_physics_finish_em(grid%t_2, **grid%t_pg_2**, grid%t_1, **grid%t_pg_1**, t0, grid%muts, th_phy, **th_phy_pg**, grid%h_diabatic, **grid%h_diabatic_pg**,dtm, config_flags, &
#if (WRF_DFI_RADAR == 1)
grid%dfi_tten_rad,**grid%dfi_tten_rad_pg**, grid%dfi_stage, &
#endif
ids, ide, jds, jde, kds, kde, &
ims, ime, jms, jme, kms, kme, &
its, ite, jts, jte, &
k_start , k_end)

Line 3950: CALL set_phys_bc_dry_2(config_flags, &
grid%u_1, grid%u_2, grid%v_1, grid%v_2, grid%w_1, grid%w_2, &
grid%t_1, **grid%t_pg_1**, grid%t_2,**grid%t_pg_2**, grid%ph_1, grid%ph_2, grid%mu_1, grid%mu_2, &
ids, ide, jds, jde, kds, kde, &
ims, ime, jms, jme, kms, kme, &
ips, ipe, jps, jpe, kps, kpe, &
grid%i_start(ij), grid%i_end(ij), &
grid%j_start(ij), grid%j_end(ij), &
k_start , k_end)

3. module_small_step_em.F → module_small_step_em.F.PG

Line 18: SUBROUTINE small_step_prep(u_1, u_2, v_1, v_2, w_1, w_2, &
t_1,**t_pg_1**,t_2,**t_pg_2**, ph_1, ph_2, &
mub, mu_1, mu_2, &
muu, muus, muv, muvs, &
mut, muts, mudf, &
u_save, v_save, w_save, &
t_save,**t_save_pg**, ph_save, mu_save, &
ww, ww_save, &
dnw, c2a, pb, p, alt, &
msfux, msfuy, msfvx, &
msfvx_inv, &
msfvy, msftx, msfty, &
rdx, rdy, &
rk_step, &
ids,ide, jds,jde, kds,kde, &
ims,ime, jms,jme, kms,kme, &
its,ite, jts,jte, kts,kte)

Line 270: t_save_pg(i,k,j) = t_pg_2(i,k,j)
t_pg_2(i,k,j) = muts(i,j)*t_pg_1(i,k,j)-mut(i,j)*t_pg_2(i,k,j)

Line 304: SUBROUTINE small_step_finish(u_2, u_1, v_2, v_1, w_2, w_1, &
t_2, **t_pg_2**,t_1,**t_pg_1**, ph_2, ph_1, ww, ww1, &
mu_2, mu_1, &
mut, muts, muu, muus, muv, muvs, &
u_save, v_save, w_save, &
t_save,**t_save_pg**,ph_save, mu_save, &
msfux, msfuy, msfvx, msfvy, &
msftx, msfty, &
h_diabatic,**h_diabatic_pg**, &

number_of_small_timesteps,dts, &
rk_step, rk_order, &
ids,ide, jds,jde, kds,kde, &
ims,ime, jms,jme, kms,kme, &
its,ite, jts,jte, kts,kte)

Line 420: t_pg_2(i,k,j) = (t_pg_2(i,k,j) + t_save_pg(i,k,j)*mut(i,j))/muts(i,j)

Line 430: t_pg_2(i,k,j) = (t_pg_2(i,k,j) + t_save_pg(i,k,j)*mut(i,j))/muts(i,j)

Line 441: t_pg_2(i,k,j) = (t_pg_2(i,k,j) -dts*number_of_small_timesteps*mut(i,j)* h_diabatic_pg(i,k,j)+t_save_pg(i,k,j)*mut(i,j))/muts(i,j)

Line 1079: SUBROUTINE advance_mu_t(ww, ww_1, u, u_1, v, v_1, &
mu, mut, muave, muts, muu, muv, &
mudf, uam, vam, wwam, t,**t_pg**, t_1,**t_pg_1**, &
t_ave,**t_ave_pg**, ft,**ft_pg**, mu_tend, &
rdx, rdy, dts, epssm, &
dnw, fnm, fnp, rdnw, &
msfux, msfuy, msfvx, msfvx_inv, &
msfvy, msftx, msfty, &
step, config_flags, &
ids, ide, jds, jde, kds, kde, &
ims, ime, jms, jme, kms, kme, &
its, ite, jts, jte, kts, kte)

Line 1281: t_ave_pg(i,k,j) = t_pg(i,k,j)
t_pg (i,k,j) = t_pg(i,k,j) + msfty(i,j)*dts*ft_pg(i,k,j)

Line 1292: wdtnpg(i,1)=0.
wdtnpg(i,kde)=0.

Line 1300: wdtnpg(i,k)= ww(i,k,j)*(fnm(k)*t_pg_1(i,k ,j)+fnp(k)*t_pg_1(i,k-1,j))

Line 1321: t_pg(i,k,j) = t_pg(i,k,j) - dts*msfty(i,j)*(& ! multiplication by mx needed for RHS terms 1 & 2

msftx(i,j)*(&

.5*rdy (v(i,k,j+1)*(t_pg_1(i,k,j+1)+t_pg_1(i,k, j)) &

-v(i,k,j)*(t_pg_1(i,k, j)+t_pg_1(i,k,j-1))) &

+ .5*rdx* &

(u(i+1,k,j)*(t_pg_1(i+1,k,j)+t_pg_1(i ,k,j)) &

-u(i ,k,j)*(t_pg_1(i ,k,j)+t_pg_1(i-1,k,j)))) &

+ rdnw(k)*(wdtnpg(i,k+1)-wdtnpg(i,k)))

Line 1345: SUBROUTINE advance_w(w, rw_tend, ww, w_save, u, v, &

mu1, mut, muave, muts, &

t_2ave,**t_2ave_pg**, t_2,**t_pg_2**, t_1,**t_pg_1**, &

ph, ph_1, phb, ph_tend, &

ht, c2a, cqw, alt, alb, &

a, alpha, gamma, &

rdx, rdy, dts, t0, epssm, &

dnw, fnm, fnp, rdnw, rdn, &

cf1, cf2, cf3, msftx, msfty,&

config_flags, top_lid, &

ids,ide, jds,jde, kds,kde, & ! domain dims

ims,ime, jms,jme, kms,kme, & ! memory dims

its,ite, jts,jte, kts,kte) ! tile dims

Line 1505: t_2ave_pg(i,k,j)=.5*((1.+epssm)*t_pg_2(i,k,j +(1.-epssm)*t_2ave_pg(i,k,j))

t_2ave_pg(i,k,j)=(t_2ave_pg(i,k,j) + muave(i,j)*t0)/(muts(i,j)*(t0+t_pg_1(i,k,j)))

4. module_initialize_quarter_ss.F → module_initialize_quarter_ss.F.PG

Line 418: scalar(i,k,j,P_QV_PG) = moist(i,k,j,P_QV)
grid%t_pg_1(i,k,j) = grid%t_1(i,k,j)
grid%t_pg_2(i,k,j) = grid%t_1(i,k,j)

Line 506: grid%t_pg_1(i,k,j)=grid%t_pg_1(i,k,j)+delt*COS(.5*PI*RAD)**2
grid%t_pg_2(i,k,j)=grid%t_pg_1(i,k,j)

Line 664: grid%t_base_pg(k) = grid%t_pg_1(1,k,1)
grid%qv_base_pg(k) = scalar(1,k,1,P_QV_PG)

Line 673: CALL wrf_dm_bcast_real(grid%t_base_pg , kte)
CALL wrf_dm_bcast_real(grid%qv_base_pg , kte)

5. module_first_rk_step_part2.F → module_first_rk_step_part2.F.PG

Line 17: SUBROUTINE first_rk_step_part2 (grid , config_flags &
, moist , moist_tend &
, chem , chem_tend &
, tracer, tracer_tend &
, scalar , scalar_tend &
, fdda3d, fdda2d &
, ru_tendf, rv_tendf &
, rw_tendf, t_tendf, **t_tendf_pg** &
, ph_tendf, mu_tendf &
, tke_tend &
, adapt_step_flag , curr_secs &
, psim , psih , wspd , gz1oz0 , br , chklowq &
, cu_act_flag , hol , th_phy, **th_phy_pg** &
, pi_phy , p_phy , t_phy , **t_phy_pg**, u_phy , v_phy &
, dz8w , p8w , t8w , rho_phy , rho &
, nba_mij, n_nba_mij & !JDM
, nba_rij, n_nba_rij & !JDM
, ids, ide, jds, jde, kds, kde &
, ims, ime, jms, jme, kms, kme &
, ips, ipe, jps, jpe, kps, kpe &
, imsx,imex,jmsx,jmex,kmsx,kmex &
, ipsx,ipex,jpsx,jpex,kpsx,kpex &
, imsy,imey,jmsy,jmey,kmsy,kmey &
, ipsy,ipey,jpsy,jpey,kpsy,kpey &
, k_start , k_end)

Line 692: CALL damptop(grid%u_2, grid%v_2, **grid%t_pg_2**, &
grid%mut, grid%muu, grid%muv, &
pi_phy, &
t_tendf_pg, ru_tendf, rv_tendf, P2SI, &

ids, ide, jds, jde, kds, kde, &
ims, ime, jms, jme, kms, kme, &
grid%i_start(ij), grid%i_end(ij), &
grid%j_start(ij), grid%j_end(ij), &
k_start, k_end)

Line 754: CALL vertical_diffusion_2(ru_tendf, rv_tendf, rw_tendf, &
t_tendf, **t_tendf_pg**, tke_tend, &
moist_tend, num_moist, &
chem_tend, num_chem, &
scalar_tend, num_scalar, &
tracer_tend, num_tracer, &
grid%u_2, grid%v_2, &
grid%t_2,**grid%t_pg_2**,grid%u_base,grid%v_base,grid%t_base,**grid%t_base_pg**,&
grid%qv_base,**grid%qv_base_pg**, &
grid%mut,grid%tke_2,config_flags, &
grid%defor13,grid%defor23,grid%defor33, &
nba_mij, num_nba_mij, & !JDM
grid%div, moist, chem, scalar,tracer, &
grid%xkmv, grid%xkhv, config_flags%km_opt, &
grid%fnm, grid%fnp, grid%dn, grid%dnw, grid%rdz, grid%rdzw, &
grid%hfx, grid%qfx, grid%ustm, rho, &
ids, ide, jds, jde, kds, kde, &
ims, ime, jms, jme, kms, kme, &
grid%i_start(ij), grid%i_end(ij), &
grid%j_start(ij), grid%j_end(ij), &
k_start, k_end)

Line 787: CALL horizontal_diffusion_2(t_tendf, **t_tendf_pg**,ru_tendf, rv_tendf, rw_tendf, &
tke_tend, &
moist_tend, num_moist, &

```
chem_tend, num_chem,                    &
scalar_tend, num_scalar,                &
tracer_tend, num_tracer,                &
grid%t_2, grid%t_pg_2, th_phy,th_phy_pg,                    &
grid%mut, grid%tke_2, config_flags,             &
grid%defor11, grid%defor22, grid%defor12,           &
grid%defor13, grid%defor23,   &
nba_mij, num_nba_mij,         & !JDM
grid%div,                     &
moist, chem, scalar,tracer,             &
grid%msfux,grid%msfuy, grid%msfvx,grid%msfvy, grid%msftx,  &
grid%msfty, grid%xkmh, grid%xkhh, config_flags%km_opt,     &
grid%rdx, grid%rdy, grid%rdz, grid%rdzw,                &
grid%fnm, grid%fnp, grid%cf1, grid%cf2, grid%cf3,         &
grid%zx, grid%zy, grid%dn, grid%dnw,                    &
ids, ide, jds, jde, kds, kde,         &
ims, ime, jms, jme, kms, kme,          &
grid%i_start(ij), grid%i_end(ij),      &
grid%j_start(ij), grid%j_end(ij),      &
k_start    , k_end                  )
```

6. module_first_rk_step_part1.F → module_first_rk_step_part1.F.PG

Line 17: SUBROUTINE first_rk_step_part1 (grid , config_flags &
, moist , moist_tend &
, chem , chem_tend &
, tracer, tracer_tend &
, scalar , scalar_tend &
, fdda3d, fdda2d &
, ru_tendf, rv_tendf &
, rw_tendf, t_tendf, **t_tendf_pg** &
, ph_tendf, mu_tendf &
, tke_tend &
, adapt_step_flag , curr_secs &
, psim , psih , wspd , gz1oz0 , br , chklowq &
, cu_act_flag , hol , th_phy, **th_phy_pg** &
, pi_phy , p_phy , t_phy , **t_phy_pg**, u_phy , v_phy &
, dz8w , p8w , t8w , rho_phy , rho &
, ids, ide, jds, jde, kds, kde &
, ims, ime, jms, jme, kms, kme &
, ips, ipe, jps, jpe, kps, kpe &
, imsx,imex,jmsx,jmex,kmsx,kmex &
, ipsx,ipex,jpsx,jpex,kpsx,kpex &
, imsy,imey,jmsy,jmey,kmsy,kmey &
, ipsy,ipey,jpsy,jpey,kpsy,kpey &
, k_start , k_end &
, f_flux)

Line 166: CALL init_zero_tendency (ru_tendf, rv_tendf, rw_tendf, &
ph_tendf, t_tendf, **t_tendf_pg**, tke_tend, &
mu_tendf, &
moist_tend,chem_tend,scalar_tend, &

tracer_tend,num_tracer, &
num_moist,num_chem,num_scalar, &
rk_step, &
ids, ide, jds, jde, kds, kde, &
ims, ime, jms, jme, kms, kme, &
grid%i_start(ij), grid%i_end(ij), &
grid%j_start(ij), grid%j_end(ij), &
k_start, k_end)

Line 194: CALL phy_prep (config_flags, &
grid%mut, grid%muu, grid%muv, grid%u_2, &
grid%v_2, grid%p, grid%pb, grid%alt, &
grid%ph_2, grid%phb, grid%t_2, **grid%t_pg_2**, grid%tsk, moist, num_moist, &
rho,th_phy, **th_phy_pg**, p_phy, pi_phy, u_phy, v_phy, &
p8w, t_phy, **t_phy_pg**, t8w, grid%z, grid%z_at_w, dz8w, &
grid%p_hyd, grid%p_hyd_w, grid%dnw, &
grid%fnm, grid%fnp, grid%znw, grid%p_top, &
grid%rthraten, &
grid%rthblten, grid%rublten, grid%rvblten, &
grid%rqvblten, grid%rqcblten, grid%rqiblten, &
grid%rucuten, grid%rvcuten, grid%rthcuten, &
grid%rqvcuten, grid%rqccuten, grid%rqrcuten, &
grid%rqicuten, grid%rqscuten, &
grid%rushten, grid%rvshten, grid%rthshten, &
grid%rqvshten, grid%rqcshten, grid%rqrshten, &
grid%rqishten, grid%rqsshten, grid%rqgshten, &
grid%rthften, grid%rqvften, **grid%rthftenpg, grid%rqvftenpg**, &
grid%RUNDGDTEN, grid%RVNDGDTEN, grid%RTHNDGDTEN, &
grid%RPHNDGDTEN,grid%RQVNDGDTEN, grid%RMUNDGDTEN,&
ids, ide, jds, jde, kds, kde, &
ims, ime, jms, jme, kms, kme, &

```
grid%i_start(ij), grid%i_end(ij),          &
grid%j_start(ij), grid%j_end(ij),          &
k_start, k_end                             )
```

7. module_em.F → module_em.F.PG

Line 15: USE module_state_description, only: param_first_scalar, p_qr, p_qv, **p_qv_pg**, p_qc, p_qg, p_qi, p_qs, tiedtkescheme, heldsuarez, positivedef, gdscheme, g3scheme, kfetascheme, monotonic, wenopd_scalar, weno_scalar, weno_mom

Line 25: SUBROUTINE rk_step_prep (config_flags, rk_step, &
u, v, w, t, **t_pg**, ph, mu, &
moist, &
ru, rv, rw, ww, php, alt, &
muu, muv, &
mub, mut, phb, pb, p, al, alb, &
cqu, cqv, cqw, &
msfux, msfuy, &
msfvx, msfvx_inv, msfvy, &
msftx, msfty, &
fnm, fnp, dnw, rdx, rdy, &
n_moist, &
ids, ide, jds, jde, kds, kde, &
ims, ime, jms, jme, kms, kme, &
its, ite, jts, jte, kts, kte)

Line 177: SUBROUTINE rk_tendency (config_flags, rk_step, &
ru_tend, rv_tend, rw_tend, ph_tend, t_tend, **t_tend_pg**, &
ru_tendf, rv_tendf, rw_tendf, ph_tendf, t_tendf, **t_tendf_pg**, &
mu_tend, u_save, v_save, w_save, ph_save, &
t_save, **t_save_pg**, mu_save, RTHFTEN, **RTHFTENPG**, &
ru, rv, rw, ww, &
u, v, w, t, **t_pg**, ph, &
u_old, v_old, w_old, t_old, **t_pg_old**, ph_old, &
h_diabatic, **h_diabatic_pg**, phb,t_init, &

mu, mut, muu, muv, mub, &
al, alt, p, pb, php, cqu, cqv, cqw, &
u_base, v_base, t_base, **t_base_pg**, qv_base,**qv_base_pg**, z_base, &
msfux, msfuy, msfvx, msfvx_inv, &
msfvy, msftx, msfty, &
clat, f, e, sina, cosa, &
fnm, fnp, rdn, rdnw, &
dt, rdx, rdy, khdif, kvdif, xkmhd, xkhh, &
diff_6th_opt, diff_6th_factor, &
adv_opt, &
dampcoef,zdamp,damp_opt,rad_nudge, &
cf1, cf2, cf3, cfn, cfn1, n_moist, &
non_hydrostatic, top_lid, &
u_frame, v_frame, &
ids, ide, jds, jde, kds, kde, &
ims, ime, jms, jme, kms, kme, &
its, ite, jts, jte, kts, kte, &
max_vert_cfl, max_horiz_cfl, dz8w, xtime) !Arguments, dz82, xtime added by Morrison, M.

Line 378: CALL zero_tend **(t_tend_pg**, &
ids, ide, jds, jde, kds, kde, &
ims, ime, jms, jme, kms, kme, &
its, ite, jts, jte, kts, kte)

Line 413: CALL zero_tend (**t_save_pg**, &
ids, ide, jds, jde, kds, kde, &
ims, ime, jms, jme, kms, kme, &
its, ite, jts, jte, kts, kte)

Line 513: CALL advect_scalar (**t_pg, t_pg, t_tend_pg**, ru, rv, ww, &
mut, time_step, config_flags, &

msfux, msfuy, msfvx, msfvy, &
msftx, msfty, fnm, fnp, &
rdx, rdy, rdnw, &
ids, ide, jds, jde, kds, kde, &
ims, ime, jms, jme, kms, kme, &
its, ite, jts, jte, kts, kte)

Line 531: CALL set_tend(**RTHFTENPG, t_tend_pg**, msfty, &
ids, ide, jds, jde, kds, kde, &
ims, ime, jms, jme, kms, kme, &
its, ite, jts, jte, kts, kte)

Line 790: CALL horizontal_diffusion_3dmp ('m', **t_pg, t_tendf_pg**, mut, &
config_flags, t_init, &
msfux, msfuy, msfvx, msfvx_inv, &
msfvy, msftx, msfty, &
khdq , xkhh, rdx, rdy, &
ids, ide, jds, jde, kds, kde, &
ims, ime, jms, jme, kms, kme, &
its, ite, jts, jte, kts, kte)

Line 829: CALL vertical_diffusion_3dmp (**t_pg, t_tendf_pg**, config_flags, t_init, &
alt, mut, rdn, rdnw, kvdq , &
ids, ide, jds, jde, kds, kde, &
ims, ime, jms, jme, kms, kme, &
its, ite, jts, jte, kts, kte)

Line 872: CALL sixth_order_diffusion('m', **t_pg, t_tendf_pg**, mut, dt, &
config_flags, &
diff_6th_opt, diff_6th_factor, &
ids, ide, jds, jde, kds, kde, &

ims, ime, jms, jme, kms, kme, &
its, ite, jts, jte, kts, kte)

Line 884: CALL rk_rayleigh_damp(ru_tendf, rv_tendf, &
rw_tendf, t_tendf, **t_tendf_pg**, &
u, v, w, t, **t_pg**, t_init, &
mut, muu, muv, ph, phb, &
u_base, v_base, t_base, **t_base_pg**, z_base, &
dampcoef, zdamp, &
ids, ide, jds, jde, kds, kde, &
ims, ime, jms, jme, kms, kme, &
its, ite, jts, jte, kts, kte)

Line 894: CALL theta_relaxation(t_tendf, **t_tendf_pg**, t, **t_pg**, t_init, &
mut, ph, phb, &
t_base, **t_base_pg**, z_base, &
ids, ide, jds, jde, kds, kde, &
ims, ime, jms, jme, kms, kme, &
its, ite, jts, jte, kts, kte)

Line 907: SUBROUTINE rk_addtend_dry (ru_tend, rv_tend, rw_tend, ph_tend, t_tend, **t_tend_pg**, &
ru_tendf, rv_tendf, rw_tendf, ph_tendf, t_tendf, **t_tendf_pg**, &
u_save, v_save, w_save, ph_save, t_save, **t_save_pg**, &
mu_tend, mu_tendf, rk_step, &
h_diabatic, **h_diabatic_pg**, mut, msftx, msfty, msfux, msfuy, &
msfvx, msfvx_inv, msfvy, &
ids,ide, jds,jde, kds,kde, &
ims,ime, jms,jme, kms,kme, &
ips,ipe, jps,jpe, kps,kpe, &
its,ite, jts,jte, kts,kte)

Line 1033:

IF(rk_step == 1)t_tendf_pg(i,k,j) = t_tendf_pg(i,k,j) + t_save_pg(i,k,j)

! divide by my to couple theta

t_tend_pg(i,k,j) = t_tend_pg(i,k,j) + t_tendf_pg(i,k,j)/msfty(i,j) &

+ mut(i,j)*h_diabatic_pg(i,k,j)/msfty(i,j)

! divide by my to couple heating

Line 1061: SUBROUTINE rk_scalar_tend (scs, sce, config_flags, &

tenddec, &

rk_step, dt, &

ru, rv, ww, mut, mub, mu_old, &

alt, &

scalar_old, scalar, &

scalar_tends, advect_tend, &

h_tendency, z_tendency, &

RQVFTEN, **RQVFTENPG**, &

base, **base_pg**, moist_step, fnm, fnp, &

msfux, msfuy, msfvx, msfvx_inv, &

msfvy, msftx, msfty, &

rdx, rdy, rdn, rdnw, &

khdif, kvdif, xkmhd, &

diff_6th_opt, diff_6th_factor, &

adv_opt, &

ids, ide, jds, jde, kds, kde, &

ims, ime, jms, jme, kms, kme, &

its, ite, jts, jte, kts, kte)

Line 1262:

IF((config_flags%cu_physics == GDSCHEME .OR. config_flags%cu_physics == G3SCHEME .OR. &

```
      config_flags%cu_physics == KFETASCHEME .OR. &      ! new trigger in KF
      config_flags%cu_physics == TIEDTKESCHEME  ) &      ! Tiedtke
           .and. moist_step .and. ( im == P_QV_PG) ) THEN

CALL set_tend( RQVFTENPG, advect_tend, msfty,    &
               ids, ide, jds, jde, kds, kde,   &
               ims, ime, jms, jme, kms, kme,   &
               its, ite, jts, jte, kts, kte      )
ENDIF
```

Line 1659:
```
SUBROUTINE init_zero_tendency(ru_tendf, rv_tendf, rw_tendf, ph_tendf,  &
                  t_tendf, t_tendf_pg,tke_tendf, mu_tendf,          &
                  moist_tendf,chem_tendf,scalar_tendf,     &
                  tracer_tendf,n_tracer,                &
                  n_moist,n_chem,n_scalar,rk_step,        &
                  ids, ide, jds, jde, kds, kde,           &
                  ims, ime, jms, jme, kms, kme,           &
                  its, ite, jts, jte, kts, kte           )
```

Line 1735:
```
CALL zero_tend ( t_tendf_pg,                &
           ids, ide, jds, jde, kds, kde,    &
           ims, ime, jms, jme, kms, kme,    &
           its, ite, jts, jte, kts, kte     )
```

8. module_diffusion_em.F → module_diffusion_em.F.PG

Line 7: USE module_big_step_utilities_em, only: grid_config_rec_type, param_first_scalar, p_qv, p_qi, p_qc, **p_qv_pg**

Line 2257: SUBROUTINE horizontal_diffusion_2 (rt_tendf,**rt_tendf_pg**, ru_tendf, rv_tendf, rw_tendf, &

tke_tendf, &
moist_tendf, n_moist, &
chem_tendf, n_chem, &
scalar_tendf, n_scalar, &
tracer_tendf, n_tracer, &
thp,**thp_pg**, theta, **theta_pg**,mu, tke, config_flags, &
defor11, defor22, defor12, &
defor13, defor23, &
nba_mij, n_nba_mij, & !JDM
div, &
moist, chem, scalar,tracer, &
msfux, msfuy, msfvx, msfvy, &
msftx, msfty, xkmh, xkhh,km_opt, &
rdx, rdy, rdz, rdzw, fnm, fnp, &
cf1, cf2, cf3, zx, zy, dn, dnw, &
ids, ide, jds, jde, kds, kde, &
ims, ime, jms, jme, kms, kme, &
its, ite, jts, jte, kts, kte)

Line 2414: CALL horizontal_diffusion_s (**rt_tendf_pg**, mu, config_flags, **thp_pg**, &

msftx, msfty, msfux, msfuy, &
msfvx, msfvy, xkhh, rdx, rdy, &
fnm, fnp, cf1, cf2, cf3, &
zx, zy, rdz, rdzw, dnw, dn, &

.false., &
ids, ide, jds, jde, kds, kde, &
ims, ime, jms, jme, kms, kme, &
its, ite, jts, jte, kts, kte)

Line 3393: SUBROUTINE vertical_diffusion_2 (ru_tendf, rv_tendf, rw_tendf, rt_tendf,**rt_tendf_pg**, &
tke_tendf, moist_tendf, n_moist, &
chem_tendf, n_chem, &
scalar_tendf, n_scalar, &
tracer_tendf, n_tracer, &
u_2, v_2, &
thp,**thp_pg**,u_base,v_base,t_base,**t_base_pg**,qv_base,**qv_base_pg**,mu,tke, &
config_flags,defor13,defor23,defor33, &
nba_mij, n_nba_mij, & !JDM
div, &
moist,chem,scalar,tracer,xkmv,xkhv,km_opt,&
fnm, fnp, dn, dnw, rdz, rdzw, &
hfx, qfx, ust, rho, &
ids, ide, jds, jde, kds, kde, &
ims, ime, jms, jme, kms, kme, &
its, ite, jts, jte, kts, kte)

Line 3675-3698: Diffusion for piggybacking

Line 3716:

```
DO j = j_start, j_end
DO i = i_start, i_end
    cpm = cp * (1. + 0.8 * scalar(i,kts,j,P_QV_PG))
    hfx(i,j)=heat_flux*cp*rho(i,1,j)        ! provided for output only
    rt_tendf_pg(i,kts,j)=rt_tendf_pg(i,kts,j)+mu(i,j)*heat_flux*rdzw(i,kts,j)
```

```
ENDDO
ENDDO
```

Line 3736:

```
DO j = j_start, j_end
DO i = i_start, i_end
        cpm = cp * (1. + 0.8 * scalar(i,kts,j,P_QV_PG))
        heat_flux = hfx(i,j)/cpm/rho(i,1,j)
        rt_tendf_pg(i,kts,j)=rt_tendf_pg(i,kts,j)+mu(i,j)*heat_flux*rdzw(i,kts,j)
ENDDO
ENDDO
```

9. module_big_step_utilities_em.F → module_big_step_utilities_em.F.PG

Line 17: USE module_state_description, only: p_qg, p_qs, p_qi, gdscheme, tiedtkescheme, kfetascheme, g3scheme, p_qv, param_first_scalar, p_qr, p_qc, **p_qv_pg**

Line 4590: SUBROUTINE phy_prep (config_flags, & ! input
mu, muu, muv, u, v, p, pb, alt, ph, & ! input
phb, t, **t_pg**, tsk, moist, n_moist, & ! input
rho, th_phy,**th_phy_pg**, p_phy , pi_phy , & ! output
u_phy, v_phy, p8w, t_phy, **t_phy_pg**,t8w, & ! output
z, z_at_w, dz8w, & ! output
p_hyd, p_hyd_w, dnw, & ! output
fzm, fzp, znw, p_top, & ! params
RTHRATEN, &
RTHBLTEN, RUBLTEN, RVBLTEN, &
RQVBLTEN, RQCBLTEN, RQIBLTEN, &
RUCUTEN, RVCUTEN, RTHCUTEN, &
RQVCUTEN, RQCCUTEN, RQRCUTEN, &
RQICUTEN, RQSCUTEN, &
RUSHTEN, RVSHTEN, RTHSHTEN, &
RQVSHTEN, RQCSHTEN, RQRSHTEN, &
RQISHTEN, RQSSHTEN, RQGSHTEN, &
RTHFTEN, **RTHFTENPG**, RQVFTEN, **RQVFTENPG**, &
RUNDGDTEN, RVNDGDTEN, RTHNDGDTEN, &
RPHNDGDTEN,RQVNDGDTEN, RMUNDGDTEN, &
ids, ide, jds, jde, kds, kde, &
ims, ime, jms, jme, kms, kme, &
its, ite, jts, jte, kts, kte)

Line 4754: th_phy_pg(i,k,j) = t_pg(i,k,j) + t0
t_phy_pg(i,k,j) = th_phy_pg(i,k,j)*pi_phy(i,k,j)

Line 5077: RTHFTENPG(I,K,J)=RTHFTENPG(I,K,J)/mu(I,J)

Line 5087: RQVFTENPG(I,K,J)=RQVFTENPG(I,K,J)/mu(I,J)

Line 5154: SUBROUTINE moist_physics_prep_em(t_new, **t_pg_new**, t_old, **t_pg_old**,t0, rho, al, alb, &

p, p8w, p0, pb, ph, phb, &
th_phy, **th_phy_pg**, pii, pf, &
z, z_at_w, dz8w, &
dt,h_diabatic, **h_diabatic_pg**, &
config_flags,fzm, fzp, &
ids,ide, jds,jde, kds,kde, &
ims,ime, jms,jme, kms,kme, &
its,ite, jts,jte, kts,kte)

Line 5269: t_pg_new(i,k,j) = t_pg_new(i,k,j)-h_diabatic_pg(i,k,j)*dt

Line 5273: th_phy_pg(i,k,j) = t_pg_new(i,k,j) + t0
h_diabatic_pg(i,k,j) = th_phy_pg(i,k,j)

Line 5326: SUBROUTINE moist_physics_finish_em(t_new,**t_pg_new**,t_old,**t_pg_old**,t0,mut, &
th_phy, **th_phy_pg**,h_diabatic,**h_diabatic_pg**, dt, &
config_flags, &
#if (WRF_DFI_RADAR == 1)
dfi_tten_rad,**dfi_tten_rad_pg**,dfi_stage, &
#endif
ids,ide, jds,jde, kds,kde, &
ims,ime, jms,jme, kms,kme, &
its,ite, jts,jte, kts,kte)

Line 5410:

```
#if ( WRF_DFI_RADAR == 1 )
      IF ( PRESENT(dfi_stage) .and.  PRESENT(dfi_tten_rad_pg) ) THEN
        IF ( dfi_stage ==DFI_FWD ) THEN
          WRITE(wrf_err_message,*)'Add radar tendency: i_start,j_start: ', i_start, j_start
          CALL wrf_debug ( 100 , TRIM(wrf_err_message) )
        ENDIF
      ENDIF
    dfi_tten_max_pg=-999
    old_max_pg=-999
#endif
```

Line 5480-5536: add microphysics theta diff to perturbation theta, set h_diabatic piggyback

Line 5591: SUBROUTINE rk_rayleigh_damp(ru_tendf, rv_tendf, &
rw_tendf, t_tendf, **t_tendf_pg**, &
u, v, w, t, **t_pg**, t_init, &
mut, muu, muv, ph, phb, &
u_base, v_base, t_base, **t_base_pg**, z_base, &
dampcoef, zdamp, &
ids, ide, jds, jde, kds, kde, &
ims, ime, jms, jme, kms, kme, &
its, ite, jts, jte, kts, kte)

Line 5855-5903: Adjust potential temperature to base state piggyback.

Line 5912: SUBROUTINE theta_relaxation(t_tendf, **t_tendf_pg**, t, **t_pg**, t_init, &
mut, ph, phb, &
t_base, **t_base_pg**, z_base, &
ids, ide, jds, jde, kds, kde, &
ims, ime, jms, jme, kms, kme, &

its, ite, jts, jte, kts, kte)

Line 6030-6051: Adjust potential temperature to base state piggyback.

10. module_bc_em.F → module_bc_em.F.PG

Line 156: SUBROUTINE relax_bdy_dry (config_flags, &

ru_tendf, rv_tendf, ph_tendf, t_tendf**, t_tendf_pg**,&

rw_tendf, mu_tend, &

ru, rv, ph, t, **t_pg**, &

w, mu, mut, &

u_bxs,u_bxe,u_bys,u_bye, &

v_bxs,v_bxe,v_bys,v_bye, &

ph_bxs,ph_bxe,ph_bys,ph_bye, &

t_bxs,t_bxe,t_bys,t_bye, &

t_pg_bxs,t_pg_bxe,t_pg_bys,t_pg_bye, &

w_bxs,w_bxe,w_bys,w_bye, &

mu_bxs,mu_bxe,mu_bys,mu_bye, &

u_btxs,u_btxe,u_btys,u_btye, &

v_btxs,v_btxe,v_btys,v_btye, &

ph_btxs,ph_btxe,ph_btys,ph_btye, &

t_btxs,t_btxe,t_btys,t_btye, &

t_pg_btxs,t_pg_btxe,t_pg_btys,t_pg_btye, &

w_btxs,w_btxe,w_btys,w_btye, &

mu_btxs,mu_btxe,mu_btys,mu_btye, &

spec_bdy_width, spec_zone, relax_zone, &

dtbc, fcx, gcx, &

ids,ide, jds,jde, kds,kde, & ! domain dims

ims,ime, jms,jme, kms,kme, & ! memory dims

ips,ipe, jps,jpe, kps,kpe, & ! patch dims

its, ite, jts, jte, kts, kte)

Line 307-324

Line 432: SUBROUTINE spec_bdy_dry (config_flags, &

ru_tend, rv_tend, ph_tend, t_tend, **t_tend_pg**, &
rw_tend, mu_tend, &
u_bxs,u_bxe,u_bys,u_bye, &
v_bxs,v_bxe,v_bys,v_bye, &
ph_bxs,ph_bxe,ph_bys,ph_bye, &
t_bxs,t_bxe,t_bys,t_bye, &
t_pg_bxs,t_pg_bxe,t_pg_bys,t_pg_bye, &
w_bxs,w_bxe,w_bys,w_bye, &
mu_bxs,mu_bxe,mu_bys,mu_bye, &
u_btxs,u_btxe,u_btys,u_btye, &
v_btxs,v_btxe,v_btys,v_btye, &
ph_btxs,ph_btxe,ph_btys,ph_btye, &
t_btxs,t_btxe,t_btys,t_btye, &
t_pg_btxs,t_pg_btxe,t_pg_btys,t_pg_btye, &
w_btxs,w_btxe,w_btys,w_btye, &
mu_btxs,mu_btxe,mu_btys,mu_btye, &
spec_bdy_width, spec_zone, &
ids,ide, jds,jde, kds,kde, & ! domain dims
ims,ime, jms,jme, kms,kme, & ! memory dims
ips,ipe, jps,jpe, kps,kpe, & ! patch dims
its, ite, jts, jte, kts, kte)

Line 539: CALL spec_bdytend **(t_tend_pg, &**
t_pg_bxs,t_pg_bxe,t_pg_bys,t_pg_bye,
t_pg_btxs,t_pg_btxe,t_pg_btys,t_pg_btye, &
't' , config_flags, &
spec_bdy_width, spec_zone, &
ids,ide, jds,jde, kds,kde, & ! domain dims
ims,ime, jms,jme, kms,kme, & ! memory dims
ips,ipe, jps,jpe, kps,kpe, & ! patch dims
its,ite, jts,jte, kts,kte)

Line 731: SUBROUTINE set_phys_bc_dry_2(config_flags, &
u_1, u_2, v_1, v_2, w_1, w_2, &
t_1, **t_pg_1**, t_2,**t_pg_2**, ph_1, ph_2, mu_1, mu_2, &
ids,ide, jds,jde, kds,kde, &
ims,ime, jms,jme, kms,kme, &
ips,ipe, jps,jpe, kps,kpe, &
its,ite, jts,jte, kts,kte)

Line 789: CALL set_physical_bc3d**(t_pg_1**, 'p', config_flags, &
ids, ide, jds, jde, kds, kde, &
ims, ime, jms, jme, kms, kme, &
ips, ipe, jps, jpe, kps, kpe, &
its, ite, jts, jte, kts, kte)

Line 809: CALL set_physical_bc3d(**t_pg_2**, 'p', config_flags, &
ids, ide, jds, jde, kds, kde, &
ims, ime, jms, jme, kms, kme, &
ips, ipe, jps, jpe, kps, kpe, &
its, ite, jts, jte, kts, kte)

Line 977: SUBROUTINE rk_phys_bc_dry_2(config_flags, u, v, w, &
t, **t_pg**, ph, mu, &
ids,ide, jds,jde, kds,kde, &
ims,ime, jms,jme, kms,kme, &
ips,ipe, jps,jpe, kps,kpe, &
its,ite, jts,jte, kts,kte)

Line 1023: CALL set_physical_bc3d(**t_pg**, 'p', config_flags, &
ids, ide, jds, jde, kds, kde, &
ims, ime, jms, jme, kms, kme, &

ips, ipe, jps, jpe, kps, kpe, &
its, ite, jts, jte, kts, kte)

Line 1050: SUBROUTINE zero_bdytend (&
u_btxs,u_btxe,u_btys,u_btye, &
v_btxs,v_btxe,v_btys,v_btye, &
ph_btxs,ph_btxe,ph_btys,ph_btye, &
t_btxs,t_btxe,t_btys,t_btye, &
t_pg_btxs,t_pg_btxe,t_pg_btys,t_pg_btye, &
w_btxs,w_btxe,w_btys,w_btye, &
mu_btxs,mu_btxe,mu_btys,mu_btye, &
moist_btxs,moist_btxe, &
moist_btys,moist_btye, &
spec_bdy_width,n_moist, &
ids,ide, jds,jde, kds,kde, & ! domain dims
ims,ime, jms,jme, kms,kme, & ! memory dims
ips,ipe, jps,jpe, kps,kpe, & ! patch dims
its,ite, jts,jte, kts,kte)

Line 1106:

t_pg_btxs = 0.
t_pg_btxe = 0.
t_pg_btys = 0.
t_pg_btye = 0.

11. couple_or_uncouple_em.F → couple_or_uncouple_em.F.PG

Line 243:

```
grid%t_pg_2(i,k,j)  =  grid%t_pg_2(i,k,j)*mut_2(i,j)
```

Line 340:

```
CALL set_physical_bc3d( grid%t_pg_1, 't',       &
                        config_flags,           &
                        ids,ide, jds,jde, kds,kde,  & ! domain dims
                        ims,ime, jms,jme, kms,kme,  & ! memory dims
                        ips,ipe, jps,jpe, kps,kpe,  & ! patch  dims
                        ips,ipe, jps,jpe, kps,kpe )
```

Line 353:

```
CALL set_physical_bc3d( grid%t_pg_2, 't',       &
                        config_flags,           &
                        ids,ide, jds,jde, kds,kde,  & ! domain dims
                        ims,ime, jms,jme, kms,kme,  & ! memory dims
                        ips,ipe, jps,jpe, kps,kpe,  & ! patch  dims
                        ips,ipe, jps,jpe, kps,kpe )
```