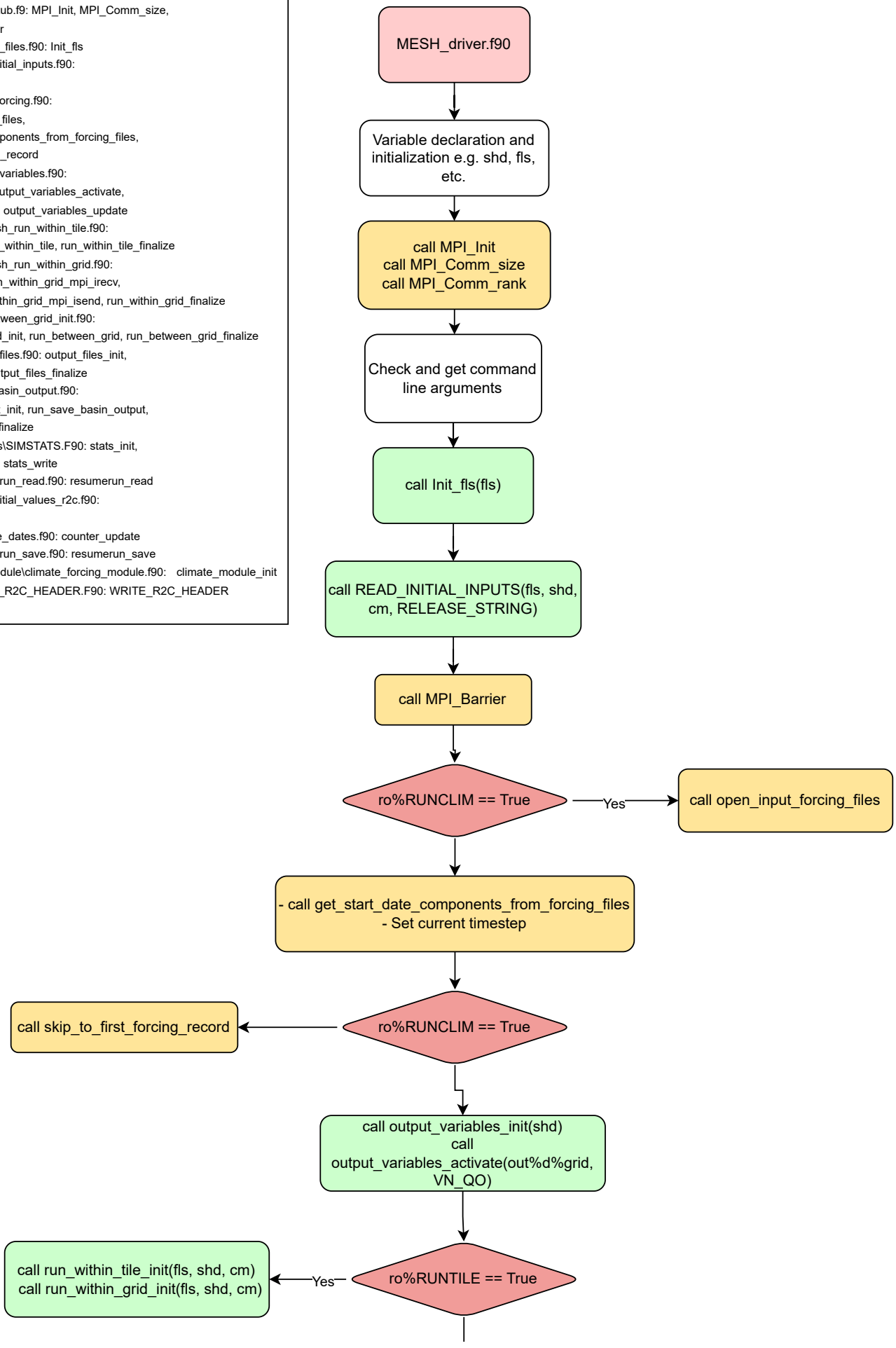
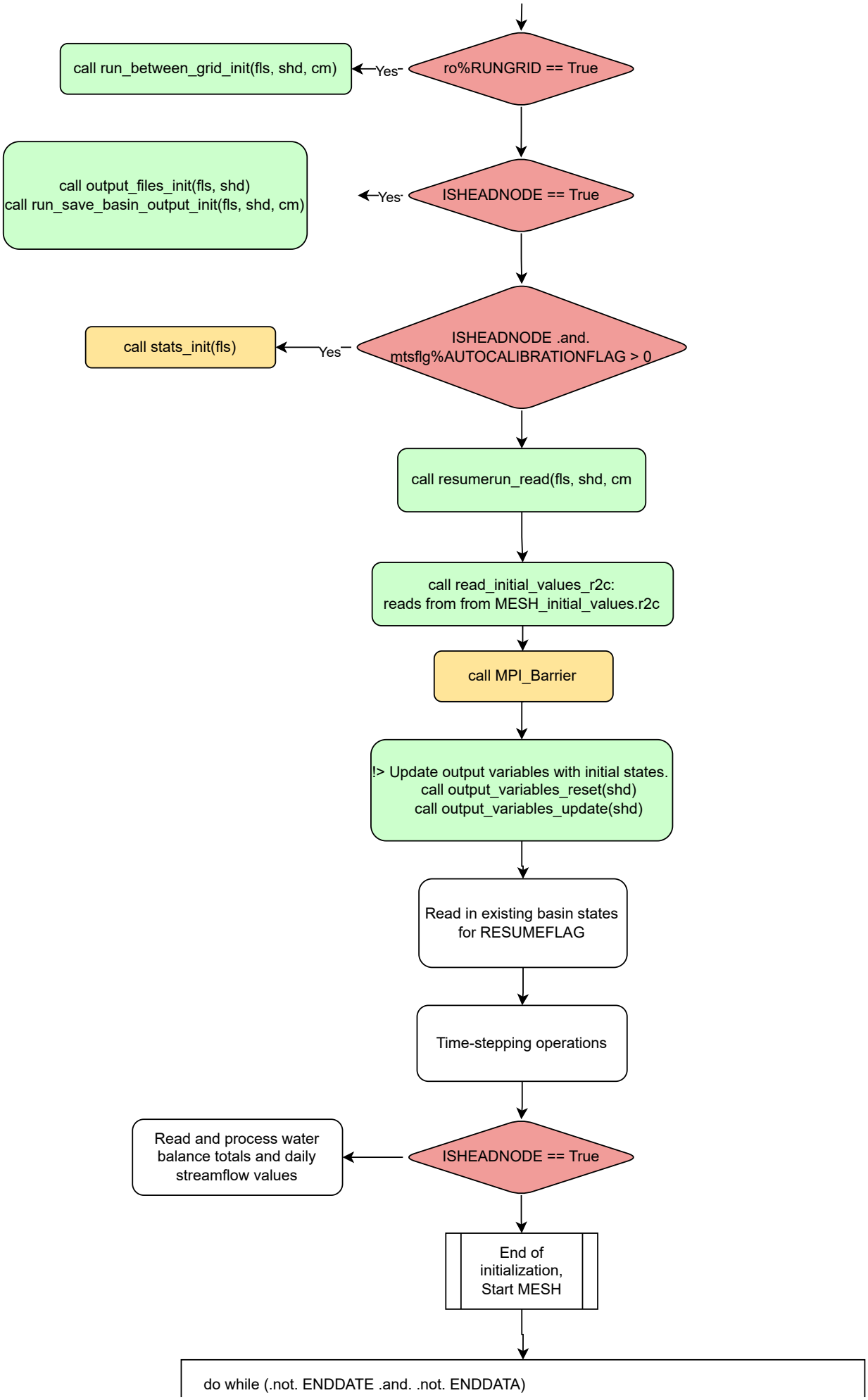
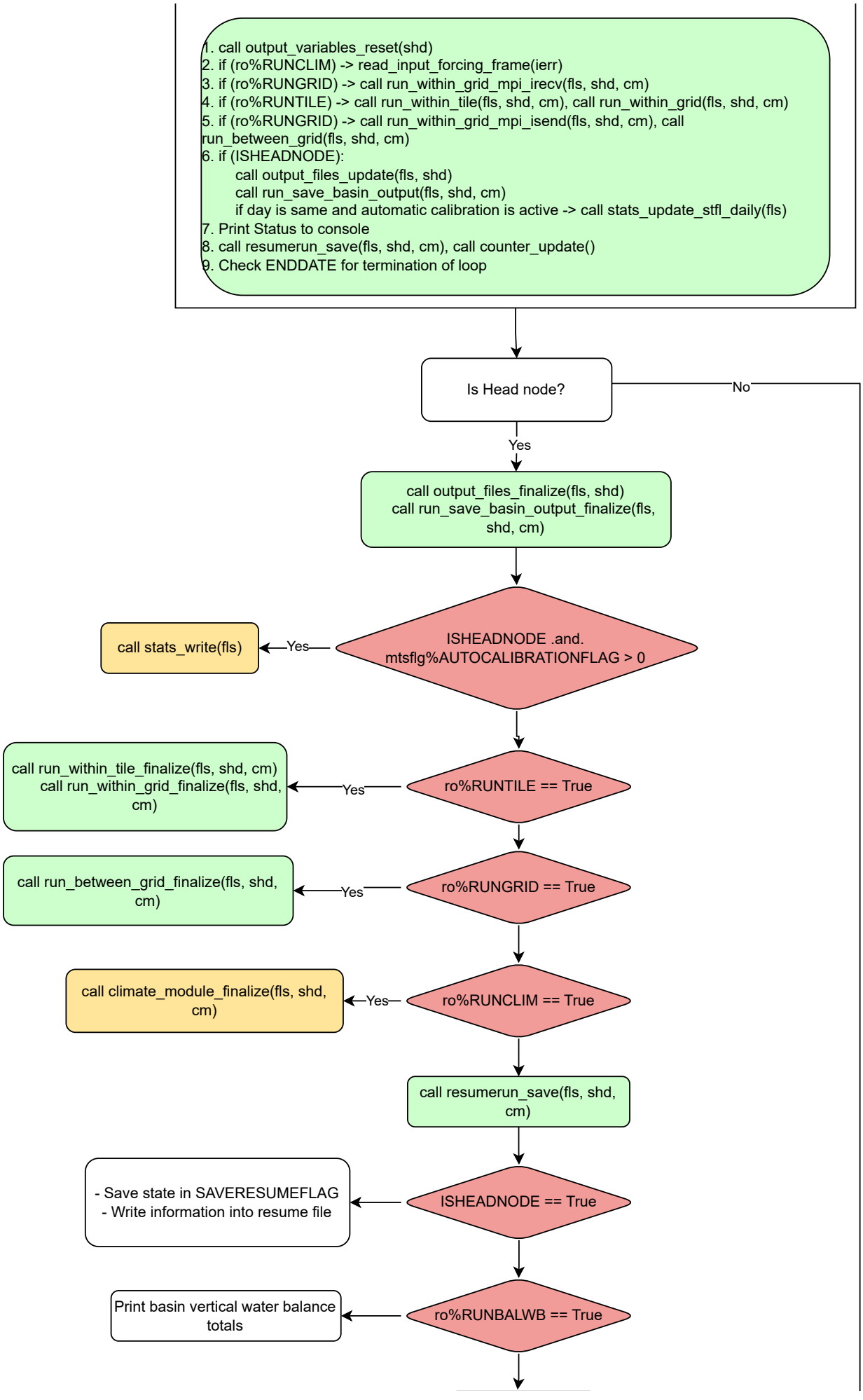
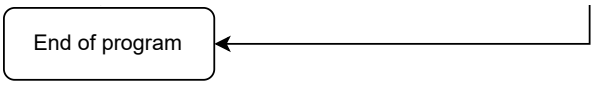
Dependencies:
- Modules\mpi_module\mpi_stub.f9: MPI_Init, MPI_Comm_size, MPI_Comm_rank, MPI_Barrier
- Driver\MESH_Driver\module_files.f90: Init_fls
- Driver\MESH_Driver\read_initial_inputs.f90: READ_INITIAL_INPUTS
- Modules\io_modules\input_forcing.f90: open_input_forcing_files, get_start_date_components_from_forcing_files, skip_to_first_forcing_record
- Driver\MESH_Driver\output_variables.f90: output_variables_init, output_variables_activate, output_variables_reset, output_variables_update
- Driver\MESH_Driver\sa_mesh_run_within_tile.f90: run_within_tile_init, run_within_tile, run_within_tile_finalize
- Driver\MESH_Driver\sa_mesh_run_within_grid.f90: run_within_grid_init, run_within_grid_mpi_irecv, run_within_grid, run_within_grid_mpi_isend, run_within_grid_finalize
- Driver\MESH_Driver\run_between_grid_init.f90: run_between_grid_init, run_between_grid, run_between_grid_finalize
- Driver\MESH_Driver\output_files.f90: output_files_init, output_files_update, output_files_finalize
- Driver\MESH_Driver\save_basin_output.f90: run_save_basin_output_init, run_save_basin_output, run_save_basin_output_finalize
- Modules\simulation_statistics\SIMSTATS.F90: stats_init, stats_update_stfl_daily, stats_write
- Driver\MESH_Driver\resumerun_read.f90: resumerun_read
- Driver\MESH_Driver\read_initial_values_r2c.f90: read_initial_values_r2c
- Driver\MESH_Driver\module_dates.f90: counter_update
- Driver\MESH_Driver\resumerun_save.f90: resumerun_save
- Modules\climate_forcing_module\climate_forcing_module.f90: climate_module_init
- Driver\MESH_Driver\WRITE_R2C_HEADER.F90: WRITE_R2C_HEADER




ro%RUNGRID == True
Yes
call run_between_grid_init(fls, shd, cm)
ISHEADNODE == True
Yes
call output_files_init(fls, shd)
call run_save_basin_output_init(fls, shd, cm)
ISHEADNODE .and.
mtsflg%AUTOCALIBRATIONFLAG > 0
Yes
call stats_init(fls)
call resumerun_read(fls, shd, cm
call read_initial_values_r2c:
reads from from MESH_initial_values.r2c
call MPI_Barrier
!> Update output variables with initial states.
call output_variables_reset(shd)
call output_variables_update(shd)
Read in existing basin states
for RESUMEFLAG
Time-stepping operations
ISHEADNODE == True
Read and process water
balance totals and daily
streamflow values
End of
initialization,
Start MESH
do while (.not. ENDDATE .and. .not. ENDDATA)


1. call output_variables_reset(shd)
2. if (ro%RUNCLIM) -> read_input_forcing_frame(ierr)
3. if (ro%RUNGRID) -> call run_within_grid_mpi_irecv(fls, shd, cm)
4. if (ro%RUNTILE) -> call run_within_tile(fls, shd, cm), call run_within_grid(fls, shd, cm)
5. if (ro%RUNGRID) -> call run_within_grid_mpi_isend(fls, shd, cm), call run_between_grid(fls, shd, cm)
6. if (ISHEADNODE):
call output_files_update(fls, shd)
call run_save_basin_output(fls, shd, cm)
if day is same and automatic calibration is active -> call stats_update_stfl_daily(fls)
7. Print Status to console
8. call resumerun_save(fls, shd, cm), call counter_update()
9. Check ENDDATE for termination of loop
Is Head node?
No
Yes
call output_files_finalize(fls, shd)
call run_save_basin_output_finalize(fls, shd, cm)
ISHEADNODE .and. mtsflg%AUTOCALIBRATIONFLAG > 0
Yes
call stats_write(fls)
ro%RUNTILE == True
Yes
call run_within_tile_finalize(fls, shd, cm)
call run_within_grid_finalize(fls, shd, cm)
ro%RUNGRID == True
Yes
call run_between_grid_finalize(fls, shd, cm)
ro%RUNCLIM == True
Yes
call climate_module_finalize(fls, shd, cm)
call resumerun_save(fls, shd, cm)
ISHEADNODE == True
- Save state in SAVERESUMEFLAG
- Write information into resume file
ro%RUNBALWB == True
Print basin vertical water balance totals


End of program