```
!-------------------------------------------------------------------------
subroutine vor_div_from_uv_grid_3d(u_grid, v_grid, vor_spec, div_spec, triang)
!-------------------------------------------------------------------------

real,    intent(in),  dimension(:,:,:) :: u_grid, v_grid
complex, intent(out), dimension(:,:,:) :: vor_spec, div_spec
logical, intent(in), optional :: triang

complex, dimension(size(vor_spec,1), size(vor_spec,2), size(vor_spec,3)) :: dx_spec,
dy_spec
real   , dimension(size(u_grid  ,1), size(u_grid  ,2), size(u_grid  ,3)) :: grid_tmp

logical :: do_triang

if(.not.module_is_initialized) then
  call error_mesg('vor_div_from_uv_grid','transforms module is not initialized', FATAL)
end if

if(present(triang)) then
  do_triang = triang
else
  do_triang = .true.
endif

grid_tmp = u_grid
call divide_by_cos(grid_tmp)
call trans_grid_to_spherical(grid_tmp, dx_spec, do_truncation=.false.)

grid_tmp = v_grid
call divide_by_cos(grid_tmp)
call trans_grid_to_spherical(grid_tmp, dy_spec, do_truncation=.false.)

call compute_vor_div(dx_spec, dy_spec, vor_spec, div_spec)

if(do_triang) then
  call triangular_truncation(vor_spec)
  call triangular_truncation(div_spec)
else
  call rhomboidal_truncation(vor_spec)
  call rhomboidal_truncation(div_spec)
endif

return
end subroutine vor_div_from_uv_grid_3d

!-------------------------------------------------------------------------
subroutine compute_vor_div_3d(u_cos, v_cos, vorticity, divergence)
!-------------------------------------------------------------------------

complex, intent(in), dimension (:,:,:) :: u_cos
complex, intent(in), dimension (:,:,:) :: v_cos
complex, intent(out), dimension (:,:,:) :: vorticity
complex, intent(out), dimension (:,:,:) :: divergence

vorticity  = compute_alpha_operator(v_cos, u_cos, -1)
divergence = compute_alpha_operator(u_cos, v_cos, +1)

return
end subroutine compute_vor_div_3d

!-------------------------------------------------------------------------
```

```
subroutine uv_grid_from_vor_div_3d(vor_spec, div_spec, u_grid, v_grid)
!-----------------------------------------------------------------------

complex, intent(in) , dimension(:,:,:) :: vor_spec, div_spec
real,    intent(out), dimension(:,:,:) :: u_grid, v_grid

complex , dimension(size(vor_spec,1), size(vor_spec,2), size(vor_spec,3)) :: dx_spec,
dy_spec

if(.not.module_is_initialized) then
  call error_mesg('uv_grid_from_vor_div','transforms module is not initialized', FATAL)
end if

call compute_ucos_vcos          (vor_spec, div_spec, dx_spec, dy_spec)
call trans_spherical_to_grid    (dx_spec , u_grid)
call trans_spherical_to_grid    (dy_spec , v_grid)
call divide_by_cos              (u_grid)
call divide_by_cos              (v_grid)

return
end subroutine uv_grid_from_vor_div_3d
```