section 3

(Compute bernouilli term = Kinetic Energy + geopotential)

(boussinesq)

(first use hydrostatic balance with theta*rhodz to find pk (Lagrange multiplier=pressure))

(uppermost layer)

ij=ij_begin_ext,ij_end_ext

pk(ij,llm) = ptop + (.5*g)*theta(ij,llm)*rhodz(ij,llm)

(other layers)

l = llm-1, 1, -1

ij=ij_begin_ext,ij_end_ext

pk(ij,l) = pk(ij,l+1) + (.5*g)*(theta(ij,l)*rhodz(ij,l)+theta(ij,l+1)*rhodz(ij,l+1))

(now pk contains the Lagrange multiplier (pressure))

l=ll_begin,ll_end

ij=ij_begin_ext,ij_end_ext

berni(ij,l) = pk(ij,l)

ij=ij_begin,ij_end

(Ai(ij) > 1.e-30)

```
berni(ij,l) = berni(ij,l) + &
1/(4*Ai(ij))*(le(ij+u_right)*de(ij+u_right)*u(ij+u_right,l)**2 +    &
le(ij+u_rup)*de(ij+u_rup)*u(ij+u_rup,l)**2 +         &
le(ij+u_lup)*de(ij+u_lup)*u(ij+u_lup,l)**2 +         &
le(ij+u_left)*de(ij+u_left)*u(ij+u_left,l)**2 +      &
le(ij+u_ldown)*de(ij+u_ldown)*u(ij+u_ldown,l)**2 +   &
le(ij+u_rdown)*de(ij+u_rdown)*u(ij+u_rdown,l)**2 )
```

(from now on pk contains the vertically-averaged geopotential)

ij=ij_begin_ext,ij_end_ext

pk(ij,l) = .5*(geopot(ij,l)+geopot(ij,l+1))

(compressible)

berni(:,:) = 0.0

l=ll_begin,ll_end

ij=ij_begin_ext,ij_end_ext

berni(ij,l) = .5*(geopot(ij,l)+geopot(ij,l+1))

ij=ij_begin,ij_end

(Ai(ij) > 1.e-30)

```
berni(ij,l) = berni(ij,l)
+ 1/(4*Ai(ij))*(le(ij+u_right)*de(ij+u_right)*u(ij+u_right,l)**2 +    &
le(ij+u_rup)*de(ij+u_rup)*u(ij+u_rup,l)**2 +         &
le(ij+u_lup)*de(ij+u_lup)*u(ij+u_lup,l)**2 +         &
le(ij+u_left)*de(ij+u_left)*u(ij+u_left,l)**2 +      &
le(ij+u_ldown)*de(ij+u_ldown)*u(ij+u_ldown,l)**2 +   &
le(ij+u_rdown)*de(ij+u_rdown)*u(ij+u_rdown,l)**2 )
```

(Add gradients of Bernoulli and Exner functions to du)

l=ll_begin,ll_end

ij=ij_begin,ij_end

(de(ij+u_right) > 1.0)

```
du(ij+u_right,l) = du(ij+u_right,l) + 1/de(ij+u_right) * ( &
0.5*(theta(ij,l)+theta(ij+t_right,l))              &
*( ne_right*pk(ij,l)+ne_left*pk(ij+t_right,l))     &
+ ne_right*berni(ij,l)+ne_left*berni(ij+t_right,l) )
```

(de(ij+u_lup) > 1.0)

```
du(ij+u_lup,l) = du(ij+u_lup,l) +  1/de(ij+u_lup) * (              &
0.5*(theta(ij,l)+theta(ij+t_lup,l))                &
*( ne_lup*pk(ij,l)+ne_rdown*pk(ij+t_lup,l))        &
+ ne_lup*berni(ij,l)+ne_rdown*berni(ij+t_lup,l) )
```

(de(ij+u_ldown) > 1.0)

```
du(ij+u_ldown,l) = du(ij+u_ldown,l) + 1/de(ij+u_ldown) * (          &
0.5*(theta(ij,l)+theta(ij+t_ldown,l))              &
*( ne_ldown*pk(ij,l)+ne_rup*pk(ij+t_ldown,l))      &
+ ne_ldown*berni(ij,l)+ne_rup*berni(ij+t_ldown,l) )
```

end subroutine compute_caldyn_hoziz