Unsteady correction term (`ddtCorr`)

- The converged solution of the original Rhie–Chow algorithm for unsteady flows is time step dependent (Choi, Numerical Heat Transfer, Part A, 36, 545–550, 1999).
- OpenFOAM `ddtCorr` resembles the unsteady correction term developed by Choi, (1999).

```
surfaceScalarField phiHbyA
(
    "phiHbyA",
    fvc::flux(HbyA)
  + fvc::interpolate(rAU)*fvc::ddtCorr(U, phi)
);
```

- The correction term is relatively small in many practical calculations. However, obtaining a time step dependent solution is undesirable.
- `ddtCorr` is a general correction method that has different definitions in for stationary (`ddtCorr(U, phi)`) and dynamic meshes (`ddtCorr(U, Uf)`).
- Here, we focus on stationary mesh, for instance the `icoFoam` solver. In such cases, the function `fvcDdtPhiCorr(U, phi)` is called.
- The implementation of `fvcDdtPhiCorr(U, phi)` depends on the employed time-discretization scheme. In other words, based on the employed scheme, the correct `fvcDdtPhiCorr(U, phi)` is called (dynamic binding).

Unsteady correction term (`ddtCorr`)

- For simplicity, we investigate the this correction term for first order implicit time discretization (`Euler`). Therefore, we should look at the `fvcDdtPhiCorr(U, phi)` function in the `EulerDdtScheme` class.

- The definition reads:

$$\phi_{\text{ddtCorr}} = \frac{\gamma}{\Delta t\ (a_P^{\mathbf{u}})_f^n} \left(\phi_{\text{PISO}}^{n-1} - \mathbf{u}_f^{n-1} \cdot \mathbf{s}_f \right)$$

- ϕ_{PISO}^{n-1} and $\mathbf{u}^{n-1}$ are conservative face flux and velocity field at the previous time step. $a_P^{\mathbf{u}}$ is the diagonal coefficients of the discretized momentum equation. Operator $(\cdot)_f$ is interpolation on the cell face f.

- $\frac{1}{(a_P^{\mathbf{u}})_f^n}$ is implemented in the construction of `phiHbyA` as `fvc::interpolate(rAU)`.

- The rest of the equation is implemented in the `fvcDdtPhiCorr(U, phi)` function.

Unsteady correction term (`ddtCorr`)

$$\phi_{\text{ddtCorr}} = \frac{\gamma}{\Delta t\ (a_P^{\mathbf{u}})_f^n} \left(\phi_{\text{PISO}}^{n-1} - \mathbf{u}_f^{n-1} \cdot \mathbf{s}_f \right)$$

```
dimensionedScalar rDeltaT = 1.0/mesh().time().deltaT();
fluxFieldType phiCorr
(
    phi.oldTime() - fvc::dotInterpolate(mesh().Sf(), U.oldTime())
);
return tmp<fluxFieldType>
(
    new fluxFieldType
    (
        IOobject
        (
            "ddtCorr(" + U.name() + ',' + phi.name() + ')',
            mesh().time().timeName(),
            mesh()
        ),
        this->fvcDdtPhiCoeff(U.oldTime(), phi.oldTime(), phiCorr)
        *rDeltaT*phiCorr
    )
);
```

- γ is an undocumented additional empirical coefficient introduced in OpenFOAM and is not present in the original formulation of Choi, (1999).

Unsteady correction term (`ddtCorr`)

- γ is defined in `fvcDdtPhiCoeff` as:

$$\gamma = 1 - \min\left(\frac{|\phi_{\text{PISO}}^{n-1} - \mathbf{u}_f^{n-1} \cdot \mathbf{s}_f|}{|\phi_{\text{PISO}}^{n-1}| + \epsilon}, 1\right)$$

s

```
ddtCouplingCoeff -= min
(
    mag(phiCorr)
    /(mag(phi) + dimensionedScalar("small", phi.dimensions(), SMALL)),
    scalar(1)
);
```

- γ dampens the time correction term. Setting γ to 1 (full Choi correction) results in instabilities.
- Quoted from Henry Weller in one of the OpenFOAM bug reports: “Removing fvcDdtPhiCoeff is not an option as it would cause many cases to crash."
- Vuorinen et al. (2014) studied effect of this correction term argued that `ddtCorr` could add numerical diffusivity in some cases (Vuorinen et al., Computers & Fluids, 93, 153–163, 2014).

Effect of `ddtCorr` in `icoFoam` solver on vortex shedding behind a square