The following is roughly copied from

https://github.com/usnistgov/fipy/blob/develop/documentation/USAGE.rst#applying-robin-b

The Robin condition

$$\hat{n} \cdot (\vec{a}\phi + b\nabla\phi) = g \qquad \text{on } f = f_0$$

can often be substituted for the flux in an equation

$$\begin{aligned} \frac{\partial \phi}{\partial t} &= \nabla \cdot (\vec{a}\phi) + \nabla \cdot (b\nabla\phi) \\ \int_V \frac{\partial \phi}{\partial t}\, dV &= \int_S \hat{n} \cdot (\vec{a}\phi + b\nabla\phi)\; dS \\ \int_V \frac{\partial \phi}{\partial t}\, dV &= \int_{S\neq f_0} \hat{n} \cdot (\vec{a}\phi + b\nabla\phi)\; dS + \int_{f_0} g\, dS \end{aligned}$$

```
>>> convectionCoeff = FaceVariable(mesh=mesh, value=[a])
>>> convectionCoeff.setValue(0., where=mask)
>>> diffusionCoeff = FaceVariable(mesh=mesh, value=b)
>>> diffusionCoeff.setValue(0., where=mask)
>>> eqn = (TransientTerm() == PowerLawConvectionTerm(coeff=convectionCoeff)
>>>        + DiffusionTerm(coeff=diffusionCoeff) + (g * mask).divergence)
```

When the Robin condition does not exactly map onto the boundary flux, we can attempt to apply it term by term by taking note of the discretization of the :class:‘ fipy.terms.diffusionTerm.DiffusionTerm‘:

$$\begin{aligned} \nabla \cdot (\Gamma\nabla\phi) &\approx \sum_f \Gamma_f \left(\hat{n} \cdot \nabla\phi\right)_f A_f \\ &= \sum_{f\neq f_0} \Gamma_f \left(\hat{n} \cdot \nabla\phi\right)_f A_f + \Gamma_{f_0} \left(\hat{n} \cdot \nabla\phi\right)_{f_0} A_{f_0} \end{aligned}$$

The Robin condition can be used to substitute for the expression

$$\hat{n} \cdot (\vec{a}\phi + b\nabla\phi) = g \qquad \text{on } f = f_0$$

but we note that :term:‘FiPy‘ calculates variable values at cell centers and gradients at intervening faces. We obtain a first-order approximation for

$$\hat{n} \cdot (\vec{a}\phi + b\nabla\phi) = g \qquad \text{on } f = f_0$$

in terms of neighboring cell values by substituting

$$\begin{aligned} \phi_{f_0} &\approx \phi_P - \left(\vec{d}_{fP} \cdot \nabla\phi\right)_{f_0} \\ &\approx \phi_P - \left(\hat{n} \cdot \nabla\phi\right)_{f_0} \left(\vec{d}_{fP} \cdot \hat{n}\right)_{f_0} \end{aligned}$$

into the Robin condition, where

$$\vec{d}_{fP}$$

is the distance vector from the face center to the adjoining cell center:

$$\begin{aligned}
\hat{n}\cdot\left(\vec{a}\phi + b\nabla\phi\right)_{f_0} &= g \\
\hat{n}\cdot\left(\vec{a}\phi_P - \vec{a}\left(\hat{n}\cdot\nabla\phi\right)_{f_0}\left(\vec{d}_{fP}\cdot\hat{n}\right)_{f_0} + b\nabla\phi\right)_{f_0} &\approx g \\
\left(\hat{n}\cdot\nabla\phi\right)_{f_0} &\approx \frac{g - \hat{n}\cdot\vec{a}\phi_P}{-\left(\vec{d}_{fP}\cdot\vec{a}\right)_{f_0} + b}
\end{aligned}$$

such that

$$\nabla\cdot\left(\Gamma\nabla\phi\right) \approx \sum_{f\neq f_0}\Gamma_f\left(\hat{n}\cdot\nabla\phi\right)_f A_f + \Gamma_{f_0}\frac{g - \hat{n}\cdot\vec{a}\phi_P}{-\left(\vec{d}_{fP}\cdot\vec{a}\right)_{f_0} + b}A_{f_0}$$

an equation of the form

```
>>> eqn = TransientTerm() == DiffusionTerm(coeff=Gamma0)
```

can be constrained to have a Robin condition at a face identifed by mask by making the following modifications

```
>>> Gamma = FaceVariable(mesh=mesh, value=Gamma0)
>>> Gamma.setValue(0., where=mask)
>>> dPf = FaceVariable(mesh=mesh, value=mesh._faceToCellDistanceRatio * mesh.cellDistanc
>>> Af = FaceVariable(mesh=mesh, value=mesh._faceAreas)
>>> RobinCoeff = (mask * Gamma0 * Af / (dPf.dot(a) + b)).divergence
>>> eqn = (TransientTerm() == DiffusionTerm(coeff=Gamma)
...        + RobinCoeff * g - ImplicitSourceTerm(coeff=RobinCoeff * mesh.faceNormals.dot
```

For a :class:‘ fipy.terms.convectionTerm.ConvectionTerm‘, we can use the Robin condition directly:

$$\begin{aligned}
\nabla\cdot\left(\vec{u}\phi\right) &\approx \sum_f\left(\hat{n}\cdot\vec{u}\right)_f\phi_f A_f \\
&= \sum_{f\neq f_0}\left(\hat{n}\cdot\vec{u}\right)_f\phi_f A_f + \left(\hat{n}\cdot\vec{u}\right)_{f_0}\frac{g - b\left(\hat{n}\cdot\nabla\phi\right)_{f_0}}{\hat{n}\cdot\vec{a}}A_{f_0} \\
&= \sum_{f\neq f_0}\left(\hat{n}\cdot\vec{u}\right)_f\phi_f A_f + \left(\hat{n}\cdot\vec{u}\right)_{f_0}\frac{-g\left(\hat{n}\cdot\vec{d}_{fP}\right)_{f_0} + b\phi_P}{-\left(\vec{d}_{fP}\cdot\vec{a}\right)_{f_0} + b}A_{f_0}
\end{aligned}$$