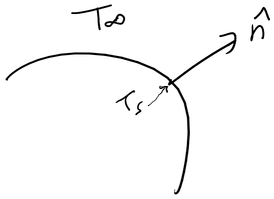

$$q''_n = -k\left(\vec{\nabla} T\right)\cdot \hat{n} = h\left(T_s - T_\infty\right)$$

The following is roughly copied from

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

The Robin condition

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

$$\hat{n}\cdot\left(\quad \vec{\nabla} T\right) = h(T - T_\infty)/(-k)$$

$\phi = T$

$b = 1$

$g = \dfrac{h(T-T_\infty)}{-k}$

can often be substituted for the flux in an equation

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

```
>>> 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‘:

in other words the diffusion term in the PDE of interest is not b.

$$\nabla\cdot(\Gamma\nabla\phi) \approx \sum_f \Gamma_f\,(\hat{n}\cdot\nabla\phi)_f\,A_f$$

Units don't work here

$\frac{1}{m}[\Gamma]\frac{1}{m}[\phi] \neq [\Gamma]\frac{[\phi]}{m}m^2$

$$= \sum_{f\neq f_0} \Gamma_f\,(\hat{n}\cdot\nabla\phi)_f\,A_f + \Gamma_{f_0}\,(\hat{n}\cdot\nabla\phi)_{f_0}\,A_{f_0}$$

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

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

$$\vec{d}_{fP} \approx \hat{n}\left(\vec{d}_{fP}\cdot\hat{n}\right)_{f_0}$$

into the Robin condition, where

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

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

$$\hat{n} \cdot \left(\vec{a}\phi + b\nabla\phi\right)_{f_0} = g$$

$$\hat{n} \cdot \left(\overbrace{\vec{a}\phi_P - \vec{a}\underbrace{\left(\hat{n}\cdot\nabla\phi\right)_{f_0}}\left(\vec{d}_{fP}\cdot\hat{n}\right)_{f_0}}^{\vec{a}\,\phi_{f_0}} + b\underbrace{\nabla\phi} \right)_{f_0} \approx g \qquad \vec{d}_{fP} \approx \hat{n}\left(\vec{d}_{fP}\cdot\hat{n}\right)_{f_0}$$

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

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

CellDistanceVectors

```
>>> 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
```

dot(a)))

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

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