We consider the task of finding a solution to the following Neumann problem:

$$\left.\begin{aligned} -\nabla^2\phi &= f && \in \Omega \\ -\frac{\partial \phi}{\partial n} &= g && \in \Gamma = \partial\Omega \\ \int_\Omega \phi &= V_0 && \end{aligned}\right\} \tag{1}$$

Problem (1) is well-posed despite having only flux boundary conditions, since the solution space is restricted to functions with a given average value V_0. The weak form is derived by considering the following constrained optimization problem:

$$\inf_{\phi\in H^1(\Omega)} \sup_{\lambda\in\mathbb{R}} \quad \int_\Omega \left(\frac{1}{2}|\nabla\phi|^2 - f\phi\right) \mathrm{d}\Omega + \int_\Gamma g\phi \;\mathrm{d}\Gamma + \lambda\left(\int_\Omega \phi \;\mathrm{d}\Omega - V_0\right) \tag{2}$$

Taking the first variation of the augmented Lagrangian in (2) and setting it equal to zero leads to the following first-order optimality system:

$$\int_\Omega \nabla\phi\cdot\nabla v \;\mathrm{d}\Omega + \lambda\int_\Omega v \;\mathrm{d}\Omega + \int_\Gamma gv \;\mathrm{d}\Gamma = \int_\Omega fv \;\mathrm{d}\Omega \qquad \forall v\in H^1(\Omega) \tag{3}$$

$$\mu\int_\Omega \phi \;\mathrm{d}\Omega = \mu V_0 \qquad \forall\mu\in\mathbb{R} \tag{4}$$

In (3) and (4), the "test" functions v and μ are associated with the first variations of the unknowns $\delta\phi$ and $\delta\lambda$, respectively. In particular, note that μ simply "drops" from (4) since it appears on both sides of the equation. In discrete residual form, equations (3) and (4) can be written as:

$$F_i^{(\phi)} \equiv \int_\Omega \nabla\phi^h\cdot\nabla\varphi_i \;\mathrm{d}\Omega + \lambda^h\int_\Omega \varphi_i \;\mathrm{d}\Omega + \int_\Gamma g\varphi_i \;\mathrm{d}\Gamma - \int_\Omega f\varphi_i \;\mathrm{d}\Omega = 0 \tag{5}$$

$$F^{(\lambda)} \equiv \int_\Omega \phi^h \;\mathrm{d}\Omega - V_0 = 0 \tag{6}$$

where the φ_i are the usual C^0 finite element shape functions, and (ϕ^h, λ^h) are the unknowns. Expanding ϕ^h as

$$\phi^h = \sum_j p_j\varphi_j \tag{7}$$

lets us immediately write down the Jacobian contributions for the "primal" equation (5) as:

$$\frac{\partial F_i^{(\phi)}}{\partial p_j} = \int_\Omega \nabla\varphi_j\cdot\nabla\varphi_i \;\mathrm{d}\Omega \tag{8}$$

$$\frac{\partial F_i^{(\phi)}}{\partial \lambda^h} = \int_\Omega \varphi_i \;\mathrm{d}\Omega \tag{9}$$

For the Lagrange multiplier equation (6), the Jacobian contributions are

$$\frac{\partial F^{(\lambda)}}{\partial p_j} = \int_\Omega \varphi_j \;\mathrm{d}\Omega \tag{10}$$

$$\frac{\partial F^{(\lambda)}}{\partial \lambda^h} = 0 \tag{11}$$

These contributions are computed by the following MOOSE classes/member functions:

(8) $\rightarrow$ `Diffusion::computeQpJacobian()`
(9) $\rightarrow$ `ScalarLagrangeMultiplier::computeOffDiagJacobianScalar()`
(10) $\rightarrow$ `ScalarLagrangeMultiplier::computeOffDiagJacobianScalar()`
(11) $\rightarrow$ `PostprocessorCED::computeQpJacobian()`

We note that it is convenient to compute (9) and (10) simultaneously in the same function, since one is the "transpose" of the other, which in the scalar Lagrange multiplier case means simply a row and column vector with the same entries. In MOOSE, an `ElementIntegralVariablePostprocessor` is used to accumulate the volume integral in (6), as no element loop is performed when computing the scalar residual.