**Overall research thrust**

Adjoint formulations are often preferred to determine Quantities of Interests (QoI) and their sensitivities in response to perturbed inputs when the number of uncertain parameter is large. The method requires two unperturbed transport solves, one for the forward flux, one for the adjoint flux; the remainder of the calculations involve inner products employing these fluxes to quickly find the response and its sensitivity to a large number of perturbation scenarios. However, for time-dependent systems, using an adjoint-based method requires storing the full unperturbed forward angular flux at each time step. Only storing the converged scattering source, as is often done in steady state to save up on memory requirements, is not an applicable approach for transient problems. A potential workaround the high-dimensionality of the transport solution is to use the Variable Eddington Tensor formulation (VET) in which the angular dependence has been of the solution has been ``factored’’ out, reducing the dimensionality of the problem by 2 dimensions. This formulation relies on an Eddington factor, which relates the 2nd angular moment of the flux to the scalar flux, which would greatly reduce the memory burden by requiring only the scalar flux and Eddington factor be stored at each time step. The general idea is that a transient forward Sn problem would still be required (to generate and store the Eddington tensor at various moments in time: a memory cost of 6 real values per mesh vertex, as opposed to the memory cost of the transport solution of the number of directions per vertex). One would then perform adjoint-based sensitivity studies using the adjoint VET formulation instead of the adjoint Sn formulation. The resulting sensitivity coefficients will be missing the first-order perturbation effect on the Eddington tensor (which is only computed using the nominal parameter values).

The key to using the VET equation for finding the response to perturbations is that the Eddington factor remains relatively unperturbed. A discontinuous FEM solver was written to construct the forward and adjoint Sn-transport solutions in 1D, which can also be used to find an Eddington factor for use in the VET equation. Similarly, another discontinuous FEM solver was constructed to find the forward and adjoint VET equations. The necessary inner product functions to determine sensitivity of the system under changes to material cross sections, source strengths, and incident flux using the adjoint solution were derived for both Sn and VET formulations, and implemented alongside the solvers. We have verified that the solvers and inner products return the expected sensitivity values in theoretically simple 1D systems.

**Updates for the past quarter**

* Model systems were examined using a discontinuous FEM solver for both the Sn-transport and VET formulations for both forward and adjoint. The sensitivity results were compared between the two formulations both forward and adjoint to discern typical cases where VEF adjoint appears to differ drastically from the forward and Sn-transport adjoint methods, indicating where the unperturbed Eddington approximation begins to fail.
* In response, the VET adjoint system was reformulated without the unperturbed Eddington approximations. The additional Eddington perturbation terms are being evaluated for the model systems for error analysis to examine error resulting from the unperturbed Eddington approximation, as compared to more typical sources of error.

**Goals for the next quarter**

* Fully implement the additional Eddington perturbation terms and verify that the implementation is correct.
* For specific systems, these additional terms should be able to reconcile the difference between the VET forward derived sensitivity and the VET adjoint derived sensitivity as an additional verification for the VET adjoint implementation.
* Examine additional test cases, specifically with streaming systems where the VET adjoint derived sensitivity appears to differ significantly from the other sensitivity methods.