

Nonnegative Methods for Bilinear Discontinuous Differencing of the S_N Equations on Quadrilaterals

Peter G. Maginot,^{a*} Jean C. Ragusa,^b and Jim E. Morel^b

^a*Lawrence Livermore National Laboratory, 7000 East Avenue, Livermore, California 94551*

^b*Texas A&M University, Department of Nuclear Engineering, 3133 TAMU, College Station, Texas 77843*

Received February 1, 2016

Accepted for Publication August 12, 2016

<http://dx.doi.org/10.13182/NSE16-38>

Abstract — Historically, matrix lumping and ad hoc flux fixups have been the only methods used to eliminate or suppress negative angular flux solutions associated with the unlumped bilinear discontinuous (UBLD) finite element spatial discretization of the two-dimensional S_N equations. Though matrix lumping inhibits negative angular flux solutions of the S_N equations, it does not guarantee strictly positive solutions. In this paper, we develop and define a strictly nonnegative, nonlinear, Petrov-Galerkin finite element method that fully preserves the bilinear discontinuous spatial moments of the transport equation. Additionally, we define two ad hoc fixups that maintain particle balance and explicitly set negative nodes of the UBLD finite element solution to zero but use different auxiliary equations to fully define their respective solutions.

We assess the ability to inhibit negative angular flux solutions and the accuracy of every spatial discretization that we consider using a glancing void test problem with a discontinuous solution known to stress numerical methods. Though significantly more computationally intense, the nonlinear Petrov-Galerkin scheme results in a strictly nonnegative solution and is a more accurate solution than all the other methods considered. One fixup, based on shape preserving, results in a strictly nonnegative final solution but has increased numerical diffusion relative to the Petrov-Galerkin scheme and is less accurate than the UBLD solution. The second fixup, which preserves as many spatial moments as possible while setting negative values of the unlumped solution to zero, is less accurate than the Petrov-Galerkin scheme but is more accurate than the other fixup. However, it fails to guarantee a strictly nonnegative final solution. The fully lumped bilinear discontinuous finite element solution is the least accurate method, with significantly more numerical diffusion than the Petrov-Galerkin scheme and both fixups.

Keywords — Radiation transport, discontinuous finite element method, nonnegative.

Note — Some figures may be in color only in the electronic version.

I. INTRODUCTION

Discontinuous finite element method (DFEM) spatial discretizations of the S_N transport equations can result in negative angular flux solutions. These nonphysical, negative, angular flux solutions are not limited to DFEM; negative solutions are common to most second-order or higher spatial discretizations, e.g., diamond difference.¹ A notable exception is the step characteristic scheme, which

is second order for slab geometry calculations and is strictly nonnegative.² However, in multiple spatial dimensions, the step characteristic scheme is only first order accurate.¹ Research to address negative angular flux solutions associated with various spatial discretizations has primarily focused on three different approaches: matrix lumping,³ flux fixups,⁴ and strictly nonnegative solution representations.⁵ Most of the work related to negative angular flux solutions associated with DFEM has been limited to the linear discontinuous (LD) finite element scheme.^{4–6} However, Adams showed that the LD discretization does not

*E-mail: maginot1@llnl.gov

maintain the thick diffusion limit on quadrilaterals.³ To be relevant for thermal radiative transfer calculations, spatial discretizations of the S_N neutron transport equations must maintain the thick diffusion limit. While Adams showed that LD on quadrilateral meshes fails in the thick diffusion limit, Adams also demonstrated that bilinear discontinuous (BLD) finite elements can maintain the thick diffusion limit on quadrilaterals.³ Ultimately, we are interested in accurate thermal radiative transfer calculations on meshes of quadrilaterals. This implies that we are interested in accurate, nonnegative spatial discretizations compatible with BLD differencing of the neutron transport equation on quadrilateral meshes.

It is well known that the unlumped bilinear discontinuous (UBLD) spatial discretization yields negative angular fluxes for quadrilateral cells with grazing radiation incidence and/or large optical thickness.³ Though matrix lumping, flux fixups, and strictly nonnegative solution representations are not limited to LD, to our knowledge, only matrix lumping has been considered in an attempt to inhibit the negative angular flux solutions associated with UBLD. While more common forms of matrix lumping, such as mass matrix or combination mass and surface matrix lumping, inhibit negative angular flux solutions, they do not guarantee a strictly nonnegative BLD angular flux solution; see, for instance, Ref. 3. Wareing et al. derived a fully lumped bilinear discontinuous (FLBLD) scheme⁷ through additional lumping of the UBLD gradient operator to yield a scheme more robust than mass and/or surface lumped schemes. In Ref. 8, Adams introduced a simple corner balance (SCB) spatial discretization that is algebraically equivalent to FLBLD and yields the same zone average angular flux solution as FLBLD. Unfortunately, while SCB was more accurate and robust than UBLD on optically thick meshes, SCB was significantly less accurate than UBLD on meshes consisting of optically thin or intermediate cells.⁸ Thus, the radiation transport community is still in need of a DFEM scheme that is at least as accurate as UBLD in the fine mesh limit and robust when UBLD yields negative angular flux solutions.

In Ref. 5, we developed a nonnegative, nonlinear Petrov-Galerkin DFEM spatial differencing scheme that maintains the LD spatial moments of the S_N neutron transport equations in slab and rectangular Cartesian geometry. In this paper, we extend the main idea of Ref. 5 to create a nonnegative, nonlinear Petrov-Galerkin DFEM scheme that will maintain the BLD spatial moments of the neutron transport equation on quadrilaterals. Petrov-Galerkin finite element methods are finite element methods whose solution space and test function space are different.⁹ Other examples of Petrov-Galerkin spatial discretizations in the

radiation transport literature include the appropriately named Petrov-Galerkin spatial differencing scheme¹⁰ and the exponential DFEM spatial discretization.¹¹ Additionally, we consider two flux fixups similar to the flux fixups considered by Fichtl et al.⁴ and Hamilton et al.⁶

The remainder of this paper is divided as follows. A derivation common to all bilinear DFEM is given in Sec. II, and the description and derivation of our new, bilinear consistent set-to-zero (BCSZ), strictly nonnegative, nonlinear Petrov-Galerkin DFEM scheme is given in Sec. III. We define two ad hoc flux fixups compatible with UBLD in Sec. IV, present computational results in Sec. V, and give some concluding remarks in Sec. VI. A list of acronyms used in this work is provided in the Appendix.

II. MOMENT EQUATIONS

We begin by considering the monoenergetic discrete ordinates transport equation for two-dimensional Cartesian geometry:

$$\vec{\Omega}_d \cdot \vec{\nabla} \psi(x, y, \vec{\Omega}_d) + \sigma_t(x, y) \psi(x, y, \vec{\Omega}_d) = S(x, y, \vec{\Omega}_d) , \quad (1)$$

where

$$\begin{aligned} \vec{\Omega}_d &= \text{neutron direction} \\ \psi(x, y, \vec{\Omega}_d) &= \text{angular flux } [\text{n}/(\text{cm}^2 \cdot \text{s} \cdot \text{sr})] \text{ in direction } \vec{\Omega}_d \\ \sigma_t(x, y) &= \text{total interaction cross section } (\text{cm}^{-1}) \\ S(x, y, \vec{\Omega}_d) &= \text{total source (scattering + fixed sources)} \\ &\quad \text{in direction } \vec{\Omega}_d. \end{aligned}$$

The discrete ordinates method approximates the scalar flux $\phi(x, y)$:

$$\phi(x, y) = \int_{4\pi} \psi(x, y, \vec{\Omega}) d\vec{\Omega} , \quad (2)$$

with units of $[\text{n}/(\text{cm}^2 \cdot \text{s})]$ as

$$\phi(x, y) \approx \sum_{d=1}^{N_{dir}} \omega_d \psi(x, y, \vec{\Omega}_d) , \quad (3)$$

where

$$\omega_d = \text{quadrature weight associated with direction } d$$

$$N_{dir} = \text{total number of discrete directions}$$

$$\sum_{d=1}^{N_{dir}} = 4\pi. \quad (4)$$

From this point forward, we drop the d subscript, implying $\vec{\Omega} = \vec{\Omega}_d$ and that $\psi(x, y)$ represents $\psi(x, y, \vec{\Omega}_d)$. Following the standard Galerkin procedure, we obtain the spatial moment of Eq. (1) with respect to basis function $B_i(x, y)$ by multiplying Eq. (1) by basis function $B_i(x, y)$ and integrating the result over spatial cell K . $B_i(x, y)$ is any function that can be defined in the physical coordinate system. Assuming cellwise constant σ_t , the i 'th spatial moment is

$$\begin{aligned} & \int_K B_i(x, y) [\vec{\Omega} \cdot \vec{\nabla} \psi(x, y) + \sigma_t \psi(x, y)] dx dy \\ &= \int_K B_i(x, y) S(x, y) dx dy. \end{aligned} \quad (5)$$

Using integration by parts, the left side of Eq. (5) becomes

$$\begin{aligned} (\vec{\Omega} \cdot \oint_K \vec{n} B_i(x, y) \psi(x, y) d\ell - \int_K \psi(x, y) [\vec{\Omega} \cdot (\vec{\nabla} B_i(x, y))] dx dy \\ + \sigma_t \int_K B_i(x, y) \psi(x, y) dx dy). \end{aligned} \quad (6)$$

Defining a reference element mapping as in Fig. 1, a reference point (s, t) , $s \in [-1, 1]$ and $t \in [-1, 1]$, is transformed into a physical point (x, y) such that

$$x = x_1 B_1(s, t) + x_2 B_2(s, t) + x_3 B_3(s, t) + x_4 B_4(s, t) \quad (7a)$$

and

$$y = y_1 B_1(s, t) + y_2 B_2(s, t) + y_3 B_3(s, t) + y_4 B_4(s, t). \quad (7b)$$

In Eq. (7), $B_1(s, t)$, $B_2(s, t)$, $B_3(s, t)$, and $B_4(s, t)$ are each a unique member of the set of four, bilinear Lagrange interpolatory polynomials that exactly interpolates at one vertex

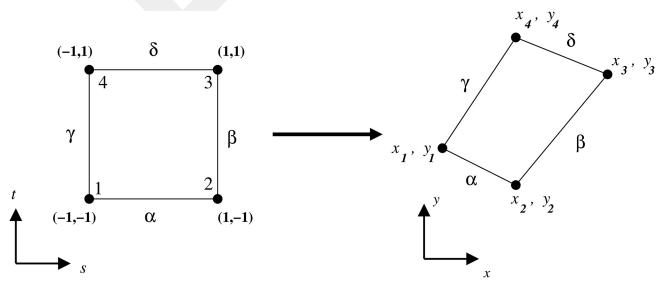


Fig. 1. Reference element mapping.

of a quadrilateral. The bilinear Lagrange functions are most easily defined on a reference element. However, the physical $B_i(x, y)$ can be visualized by plotting the isoparametric curve defined by $x(s, t)$, $y(s, t)$, $B_i(s, t)$. On the reference element, the bilinear Lagrange functions are defined as

$$B_1(s, t) = \frac{1-s}{2} \frac{1-t}{2}, \quad (8a)$$

$$B_2(s, t) = \frac{s+1}{2} \frac{1-t}{2}, \quad (8b)$$

$$B_3(s, t) = \frac{s+1}{2} \frac{t+1}{2}, \quad (8c)$$

and

$$B_4(s, t) = \frac{1-s}{2} \frac{t+1}{2}. \quad (8d)$$

With this choice of basis functions, the Galerkin procedure creates four equations with more unknowns than equations. To close the resultant system of equations, one must assume a solution representation $\tilde{\psi}(s, t)$ to approximate the true angular flux $\psi(s, t)$.

The UBLD scheme assumes a solution trial space equal to the basis space:

$$\tilde{\psi}_{UBLD}(s, t) = \sum_{j=1}^4 \vec{\psi}_{j,UBLD} B_j(s, t). \quad (9)$$

Under this assumption, the spatial moments of the transport equation become a 4×4 linear system of equations, with unknowns $\vec{\psi}_{UBLD}$ that describe bilinear function $\tilde{\psi}_{UBLD}(s, t)$. Our work here assumes that the reader is already familiar with the UBLD finite element spatial discretization. Interested readers are directed to any one of the already published, complete derivations of UBLD on rectangular or quadrilateral mesh cells. References 3, 7, and 12 are only a few examples.

The linear system of FLBLD equations is derived from the UBLD moment equations. To obtain the FLBLD equations, one lumps the mass, surface, and gradient matrices of the UBLD scheme, as detailed in Refs. 3 and 7.

The BCSZ scheme solution representation $\hat{\psi}_{BCSZ}(s, t)$ is defined as being a bilinear function $\hat{\psi}_{BCSZ}(s, t)$:

$$\hat{\psi}_{BCSZ}(s, t) = \sum_{j=1}^4 \vec{\psi}_{j,BCSZ} B_j(s, t) \quad (10)$$

everywhere $\hat{\psi}_{BCSZ}(s, t)$ is positive and zero otherwise:

$$\tilde{\psi}_{BCSZ}(s, t) = \begin{cases} \hat{\psi}_{BCSZ}(s, t) & \hat{\psi}_{BCSZ}(s, t) > 0 \\ 0 & \text{otherwise} \end{cases}. \quad (11)$$

The initial iterate of $\hat{\psi}_{BCSZ}$ is $\tilde{\psi}_{UBLD}$. If $\tilde{\psi}_{UBLD} \geq 0$ everywhere within a cell, then $\tilde{\psi}_{BCSZ} = \tilde{\psi}_{UBLD}$. Using the

definition of $\tilde{\Psi}_{\text{BCSZ}}$ given in Eq. (11) turns the four spatial moments of the transport equation into four nonlinear equations with four fundamental unknowns $\vec{\Psi}_{j,\text{BCSZ}}$.

In addition to the UBLD and FLBLD schemes, we also compare BCSZ to two flux fixup schemes. Though the flux fixup solution representations are bilinear functions, in general, the flux fixup solution representations will not satisfy the four UBLD moment equations. The exception to this rule is when the UBLD solution representation is everywhere positive within a mesh cell. The flux fixup schemes are detailed further in Sec. IV.

III. BCSZ SCHEME

The UBLD discretization within a mesh cell satisfies a linear system of equations that can be written as (see Refs. 3, 7, and 12 for more complete derivations)

$$\mathbf{A}_{\text{UBLD}} \vec{\Psi}_{\text{UBLD}} = \vec{R}_{\text{BLD}} , \quad (12)$$

where

$\mathbf{A}_{\text{UBLD}} = 4 \times 4$, nonsymmetric, dense matrix

$\vec{\Psi}_{\text{UBLD}} = 4 \times 1$ column vector of the unknown coefficients that describe $\tilde{\Psi}_{\text{UBLD}}(s, t)$

$\vec{R}_{\text{BLD}} = 4 \times 1$ column vector, with elements $\vec{R}_{\text{BLD},i}$ that represent the i 'th interpolatory spatial moments of the volumetric source within a spatial cell and upwinded angular flux contributions from the edge of the cell.

The nonlinear BCSZ scheme satisfies the following, related, nonlinear system of equations:

$$\vec{F}(\tilde{\Psi}_{\text{BCSZ}}) - \vec{R}_{\text{BLD}} = \vec{0} , \quad (13)$$

where $\vec{F}(\tilde{\Psi}_{\text{BCSZ}})$ is a vector function. Each component of $\vec{F}(\tilde{\Psi}_{\text{BCSZ}})$, $\vec{F}_i(\tilde{\Psi}_{\text{BCSZ}})$, corresponds to a unique bilinear moment equation, Eq. (6), evaluated using the nonlinear BCSZ solution representation. Every $\vec{F}_i(\tilde{\Psi}_{\text{BCSZ}})$ consists entirely of the following:

1. cell edge leakage terms, hereinafter denoted $\psi_{i,e}$
2. cell volume gradients terms, denoted $\psi_{i,\mu}$ and $\psi_{i,\eta}$
3. cell volume moments terms, denoted $\psi_{i,M}$.

The $\psi_{i,e}$ terms come from the integration by parts of the moment equations, yielding the following surface terms:

$$\begin{aligned} (\vec{\Omega} \cdot \int_K \vec{n} B_i(x, y) \psi(x, y) d\ell = & \\ (\vec{\Omega} \cdot \vec{n}_\alpha) \frac{|J_\alpha|}{2} \int_{-1}^1 B_i(s, -1) \psi(s, -1) ds & \\ + (\vec{\Omega} \cdot \vec{n}_\beta) \frac{|J_\beta|}{2} \int_{-1}^1 B_i(1, t) \psi(1, t) dt & \\ + (\vec{\Omega} \cdot \vec{n}_\delta) \frac{|J_\delta|}{2} \int_{-1}^1 B_i(s, 1) \psi(s, 1) ds & \\ + (\vec{\Omega} \cdot \vec{n}_\gamma) \frac{|J_\gamma|}{2} \int_{-1}^1 B_i(-1, t) \psi(-1, t) dt . & \end{aligned} \quad (14)$$

In Eq. (14), we have assumed that all zones K are convex with linear (not-curvilinear) edges. Thus, the outward directed edge normal \vec{n}_e along each edge e is a constant. Define an edge moment of the angular flux $\psi_{i,e}$. Using edge α as an example,

$$\psi_{i,\alpha} = \int_{-1}^1 B_i(s, -1) \psi(s, -1) ds , \quad (15)$$

Eq. (14) becomes Eq. (16):

$$\begin{aligned} (\vec{\Omega} \cdot \int_K \vec{n} B_i(x, y) \psi(x, y) d\ell = & (\vec{\Omega} \cdot \vec{n}_\alpha) \frac{|J_\alpha|}{2} \psi_{i,\alpha} \\ + (\vec{\Omega} \cdot \vec{n}_\beta) \frac{|J_\beta|}{2} \psi_{i,\beta} + (\vec{\Omega} \cdot \vec{n}_\delta) \frac{|J_\delta|}{2} \psi_{i,\delta} + (\vec{\Omega} \cdot \vec{n}_\gamma) \psi_{i,\gamma} , & \end{aligned} \quad (16)$$

where $|J_e|$ is the length of edge e and we have used edge shorthands ($\alpha, \beta, \delta, \gamma$) as in Fig. 1. The cell volume gradients, $\psi_{i,\mu}$ and $\psi_{i,\eta}$, stem from

$$\begin{aligned} \int_K \psi(x, y) [\vec{\Omega} \cdot (\vec{\nabla} B_i(x, y))] dx dy = & \\ \mu \int_{-1}^1 \int_{-1}^1 \psi(s, t) \left(\frac{\partial y}{\partial t} \frac{\partial B_i}{\partial s} - \frac{\partial y}{\partial s} \frac{\partial B_i}{\partial t} \right) ds dt + & \\ \eta \int_{-1}^1 \int_{-1}^1 \psi(s, t) \left(\frac{\partial x}{\partial s} \frac{\partial B_i}{\partial t} - \frac{\partial x}{\partial t} \frac{\partial B_i}{\partial s} \right) ds dt . & \end{aligned} \quad (17a)$$

Equation (17a) can be represented as

$$\int_K \psi(x, y) [\vec{\Omega} \cdot (\vec{\nabla} B_i(x, y))] dx dy = \mu \psi_{i,\mu} + \eta \psi_{i,\eta} , \quad (17b)$$

where $\vec{\Omega} = [\mu, \eta]^T$. In Eq. (17a), the only unknown term is $\psi(s, t)$; $\frac{\partial B_i}{\partial s}$ and $\frac{\partial B_i}{\partial t}$ can all be evaluated given the definitions of $B_i(s, t)$ in Eq. (8). Likewise, the geometric transformation terms, $\frac{\partial x}{\partial s}$, $\frac{\partial x}{\partial t}$, $\frac{\partial y}{\partial s}$, and $\frac{\partial y}{\partial t}$, come from taking the derivatives of Eq. (7) with respect to s and t , and $\vec{\Omega}$ is from a chosen

angular quadrature set. The $\psi_{i,M}$ term comes from the reaction term:

$$\begin{aligned}\sigma_t \int_K B_i(x,y) \psi(x,y) dx dy &= \sigma_t \int_{-1}^1 \int_{-1}^1 B_i(s,t) \psi |J| ds dt \\ &= \sigma_t \psi_{i,M}.\end{aligned}\quad (18)$$

In Eq. (18) $|J|$ is the determinant of the Jacobian of the coordinate transformation given in Eq. (7):

$$|J| = \begin{vmatrix} \frac{\partial x}{\partial s} & \frac{\partial y}{\partial s} \\ \frac{\partial x}{\partial t} & \frac{\partial y}{\partial t} \end{vmatrix}. \quad (19)$$

III.A. BCSZ Edge Integration

Evaluating $\psi_{i,e}$ terms requires integration of $\hat{\psi}_{BCSZ}$ along edges. Along any edge, $\hat{\psi}_{BCSZ}$ falls into one of four possible cases. Since $\hat{\psi}_{BCSZ} = 0$ or $\hat{\psi}_{BCSZ} = \hat{\psi}_{BCSZ}$, the four possible cases of $\hat{\psi}_{BCSZ}$ along an edge can be classified based on the value of $\hat{\psi}_{BCSZ}$ along each edge. Considering edge α , but with straight forward extension to any edge,

1. $\hat{\psi}_{BCSZ} > 0$ along the whole edge
2. $\hat{\psi}_{BCSZ} < 0$ along the whole edge
3. $\psi_{1,BCSZ} < 0, \psi_{2,BCSZ} > 0$
4. $\psi_{1,BCSZ} > 0, \psi_{2,BCSZ} < 0$.

Integration of case one is straightforward: The integration defining $\psi_{i,e}$ is identical to that used by the UBLD scheme. In case two, $\hat{\psi}_{BCSZ} < 0$ over the full length of the interval of integration; therefore, $\psi_{i,e}$ is zero. Cases three and four are handled by first determining s_z . s_z is the point on edge α such that $\hat{\psi}_{BCSZ}(s_z, -1) = 0$:

$$s_z = \frac{\psi_{1,BCSZ} + \psi_{2,BCSZ}}{\psi_{1,BCSZ} - \psi_{2,BCSZ}}. \quad (20)$$

Then, a two-point Gauss quadrature is mapped to the interval where $\hat{\psi}_{BCSZ} \geq 0$, and the integral of $\psi_{i,\alpha}$ is computed (exactly) using function evaluations at the mapped quadrature points.

III.B. BCSZ Cell Integration

The integral contributions to $\psi_{i,\mu}$, $\psi_{i,\eta}$, and $\psi_{i,M}$ are nontrivial only over the portion of the cell where $\hat{\psi}_{BCSZ}(s, t) \geq 0$. There are seven possible geometric integration cases to be considered for all values of $\hat{\psi}_{BCSZ}$. Six are shown in Fig. 2. The other case, when $\hat{\psi}_{BCSZ} > 0$ everywhere in the cell, devolves to the UBLD scheme. Though other combinations of nodes than those shown in Fig. 2 can be negative, all possible combinations of

negative/positive nodes and relative magnitudes can be transformed into one of the cases in Fig. 2 via permutation.

The values $\psi_{i,\mu}$, $\psi_{i,\eta}$, and $\psi_{i,M}$ are evaluated by summing the integral contribution in each region of a particular case given in Fig. 2. To evaluate $\psi_{i,\mu}$, $\psi_{i,\eta}$, and $\psi_{i,M}$ contributions in a region R bounded by a curved boundary, we make use of variable limits of integration. We are required to integrate over a region with a curved boundary because our region of nontrivial integration is everywhere $\hat{\psi}_{BCSZ}(s, t) \geq 0$. Since $\hat{\psi}_{BCSZ}$ is a bilinear function, the line where $\hat{\psi}_{BCSZ} = 0$ is a bilinear function; thus, R having a curved boundary is possible. But, because $\hat{\psi}_{BCSZ}$ is a bilinear function, we are able to define t as a function of s (or s as a function of t) everywhere along the curve where $\hat{\psi}_{BCSZ} = 0$, allowing us to use variable limits of integration. To find a variable limit of integration with respect to t as a function of s , we first transform $\hat{\psi}_{BCSZ}(s, t)$ from an interpolatory polynomial to a moment-based polynomial $f(s, t)$:

$$f(s, t) = f_c + sf_s + tf_t + stf_{st}, \quad (21)$$

where $f(s, t) = \hat{\psi}_{BCSZ}(s, t)$. Evaluating Eq. (21) at the four nodes of the reference element, and knowing that each basis function interpolates exactly at one node of the reference element, the coefficients of f are found by solving

$$\begin{bmatrix} 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} f_c \\ f_s \\ f_t \\ f_{st} \end{bmatrix} = \vec{\psi}_{BCSZ}, \quad (22)$$

where $\vec{\psi}_{BCSZ}$ is the 4×1 vector of coefficients that describe interpolatory bilinear function $\hat{\psi}_{BCSZ}(s, t)$. Equation (21) is then manipulated to find a variable limit of integration with respect to t , i.e., $\hat{l}_t(s)$:

$$\hat{l}_t(s) = -\frac{f_c + f_s s}{f_t + f_{st} s}. \quad (23)$$

Rather than integrating over the cell a total of 12 times to evaluate each $\psi_{i,\mu}$, $\psi_{i,\eta}$, and $\psi_{i,M}$ integral for every B_i , we evaluate a single, generic, bivariate polynomial integrand that is only a function of s and t . This generic bivariate polynomial is found by expanding the bivariate polynomials of the $\psi_{i,\mu}$, $\psi_{i,\eta}$, and $\psi_{i,M}$ integrands. Using $\psi_{i,M}$ as an example, we express each component of the integrand,

$$B_i(s, t) \hat{\psi}_{BCSZ}(s, t) |J|, \quad (24)$$

as a moment based, bivariate polynomial,

$$\begin{aligned}&= [b_{i,c} + sb_{i,s} + tb_{i,t} + stb_{i,st}] [f_c + sf_s + tf_t + stf_{st}] \\ &\quad \times [g_c + sg_s + tg_t],\end{aligned}\quad (25)$$

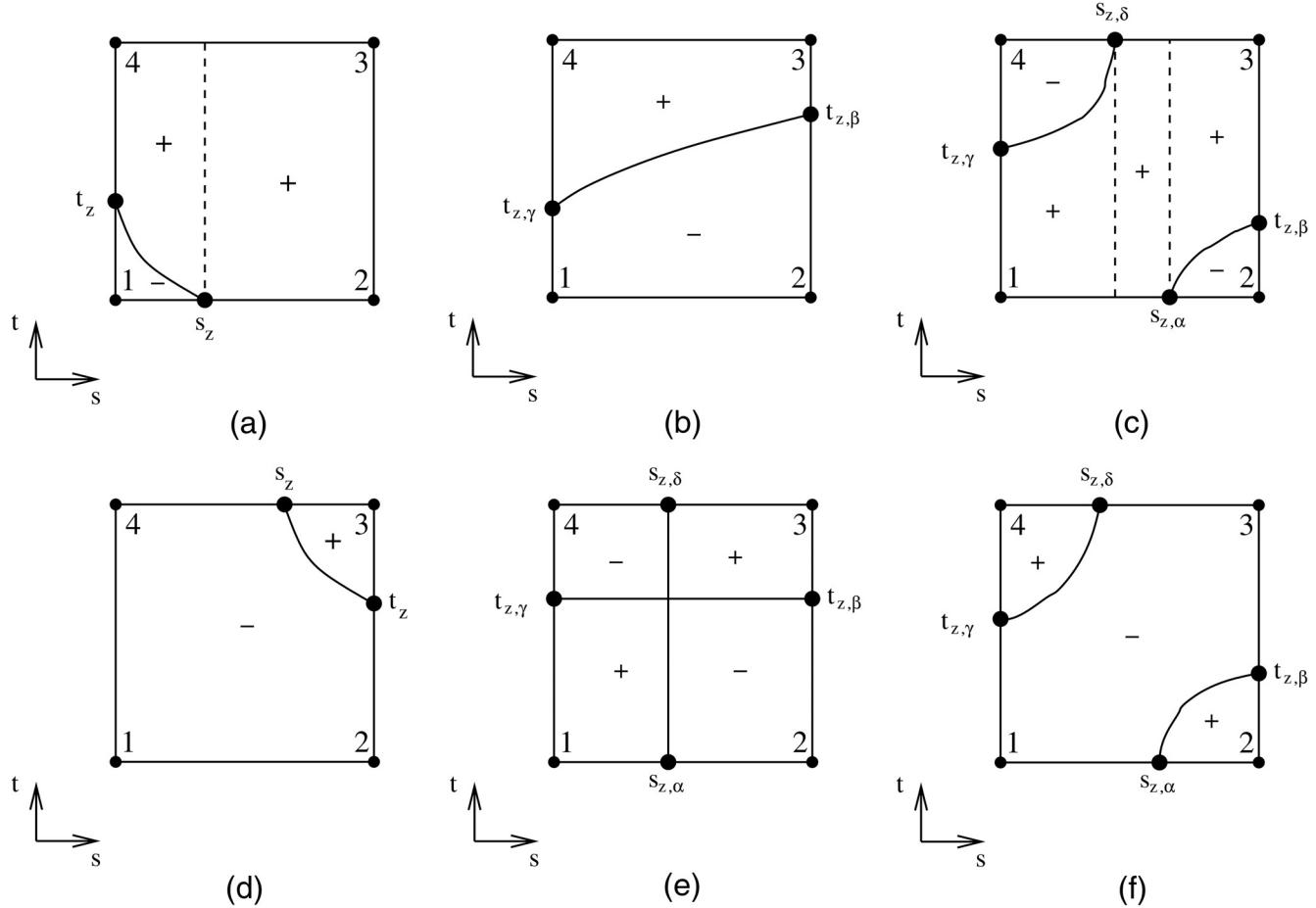


Fig. 2. BCSZ cell integration cases requiring special treatment: (a) one negative, (b) two negative same side, (c) two negative opposite sides $\psi_1\psi_3 > \psi_2\psi_4$, (d) three negative, (e) two negative block formation $\psi_1\psi_3 = \psi_2\psi_4$, and (f) two negative opposite sides $\psi_1\psi_3 > \psi_2\psi_4$.

with

$$B_i(s, t) = b_{i,c} + sb_{i,s} + tb_{i,t} + stb_{i,st} \quad (26a)$$

and

$$|J| = g_c + sg_s + tg_t. \quad (26b)$$

Then, using MATLAB (Ref. 13), Eq. (25) is further expanded by carrying out the multiplication, and terms of equal degree bivariate polynomials, $s^m t^n$, with $0 \leq m \leq 3$, $0 \leq n \leq 3$, are grouped. This allows us to calculate the 12 separate cell integrations of $\psi_{i,\mu}$, $\psi_{i,\eta}$, and $\psi_{i,M}$ required to evaluate $\vec{F}(\tilde{\psi}_{BCSZ})$ with integration of only a single generic bivariate polynomial over R . Each unique $\psi_{i,\mu}$, $\psi_{i,\eta}$, and $\psi_{i,M}$ is evaluated by replacing the constants of the generic bivariate polynomial integration with the unique set of constants associated with a given $\psi_{i,\mu}$, $\psi_{i,\eta}$, and $\psi_{i,M}$.

III.B.1. Symbolic Integration Versus Numerical Integration

Initially, symbolic algebra expressions for the integration of the generic bivariate polynomial over R were used to evaluate $\psi_{i,\mu}$, $\psi_{i,\eta}$, and $\psi_{i,M}$. With this integration scheme, the $\tilde{\psi}_{BCSZ}$ nonlinear iteration worked well for a glancing void test problem discretized with a few (≤ 25) cells but often failed at higher cell counts. At higher cell counts, the nonlinear iteration failed in cells with BCSZ unknowns $\psi_{BCSZ,i}$ on the order of slightly more than our nonlinear iteration tolerance. Given that the symbolic expressions for integrating the generic bivariate polynomial over R required several thousand floating point operations per integration, often taking the difference of high-order polynomials, e.g., seventh degree, we sought a method to verify the correctness of the symbolic expressions.

To verify the symbolic integration, we compared the exact symbolic algebra-generated expressions to the results of numerically estimating $\psi_{i,M}$ using Gauss quadrature for a particular value of $\tilde{\psi}_{BCSZ}$. In our test,

$\vec{\Psi}_{BCSZ} = [-2, 0.1, 200, 10]^T$, and we integrated over a generic quadrilateral the (x, y) coordinates, which are given in [Table I](#). We numerically evaluated the integrals using a N_s point Gauss quadrature mapped to the interval $s \in [-1, s_z]$ with a two-point Gauss quadrature in t for a given line integral at $s = s_q$. A four point, 2×2 Gauss quadrature tensor product was used to evaluate the contribution for $s \in [s_z, 1]$, $t \in [-1, 1]$. An example of the quadrature layout for $N_s = 4$ is given in [Fig. 3](#).

In [Fig. 4](#), we plot the normalized difference E_i of the symbolic evaluation $\psi_{i,M}^{sym}$ and the numerical estimation $\psi_{i,M}^{num}$ of the cell volume moments $\psi_{i,M}$, where

TABLE I

Physical Points of Quadrilateral for Verification Test

Reference Node	x	y
1	-0.9	-2
2	1.1	-1
3	0.5	1.2
4	-0.6	2

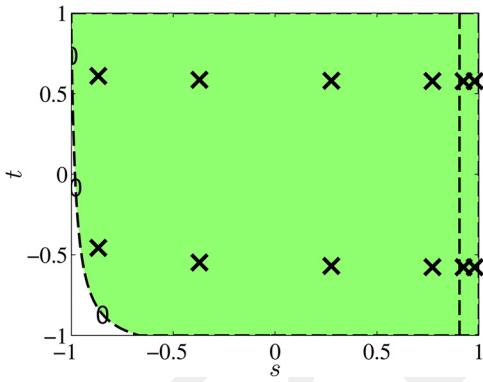
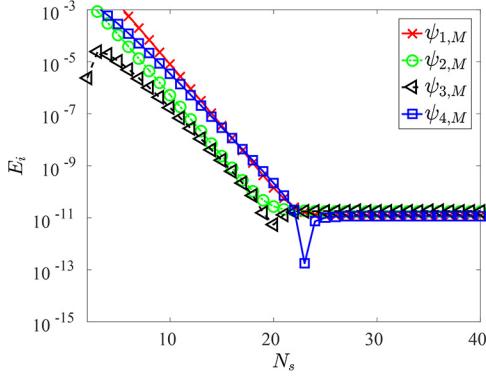


Fig. 3. Example quadrature point locations for quadrature integration test.

Fig. 4. E_i for quadrature test.

$$E_i = \frac{|\psi_{i,M}^{sym} - \psi_{i,M}^{num}|}{|\psi_{i,M}^{sym}|}. \quad (27)$$

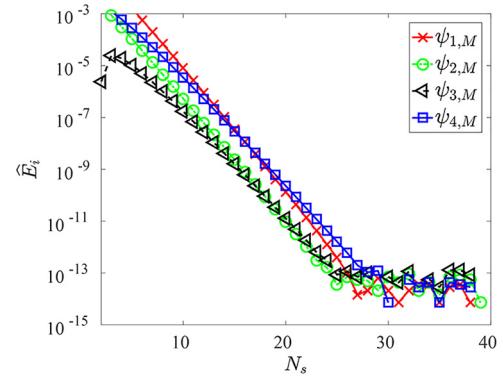
In [Eq. \(27\)](#), the quadrature evaluation of $\psi_{i,M}$ uses a total of $2N_s + 4$ quadrature points. We expect to see a plateauing of error as N_s increases, similar to the numerical plateau observed in [Fig. 4](#), indicating that the numerical estimate is converging to the symbolic expression. However, we would expect to see the plateau much nearer the machine precision of $\approx 10^{-16}$, not $\approx 10^{-11}$.

Compare the result of [Fig. 4](#) with [Fig. 5](#), which plots \hat{E}_i . As defined in [Eq. \(28\)](#), \hat{E}_i is the difference between a quadrature estimate of $\psi_{i,M}$, using a small number of quadrature points $\psi_{i,M}^{num}$, and a quadrature estimate of $\psi_{i,M}$, which uses a very fine quadrature set $\psi_{i,M}^{MAX}$

$$\hat{E}_i = \frac{|\psi_{i,M}^{MAX} - \psi_{i,M}^{num}|}{|\psi_{i,M}^{MAX}|}. \quad (28)$$

In [Eq. \(28\)](#), $\psi_{i,M}^{MAX}$ is the quadrature approximation of $\psi_{i,M}$ with $N_s = 40$. The error plateau of [Fig. 5](#) is noisier than that of [Fig. 4](#). However, the curve in [Fig. 5](#) plateaus near 10^{-14} , three orders of magnitude closer to machine precision than the curve in [Fig. 4](#). Assuming that increasing N_s increases the accuracy of a given quadrature approximation $\psi_{i,M}^{num}$, comparing [Figs. 4](#) and [5](#) demonstrates that the exact symbolically generated $\psi_{i,M}^{sym}$ suffers from numerical round-off error, presumably caused by repeatedly propagating small differences of large numbers. As such, we prefer the higher precision of quadrature integration to evaluate cell integral quantities for the BCSZ scheme.

For a fixed N_s , we cannot a priori estimate the error in our quadrature approximation of the bivariate polynomial integration over the region with a curved boundary. Because of particle balance concerns, we wish to tightly converge the nonlinear system of equations, requiring an ability to control the numerical integration error. Gauss-Kronrod quadrature¹⁴ allows for an estimate of the

Fig. 5. \hat{E}_i for quadrature test.

quadrature approximation error of an integration. In practice, we use the 7-point Gauss/15-point Kronrod quadrature set in s with a 2-point Gauss quadrature in t to integrate all BCSZ quantities. Using subinterval refinement, this quadrature integration strategy, and the decompositions shown in Fig. 2, allows for the computation of all BCSZ cell integral quantities to a tolerance less than our nonlinear iteration tolerance. Use of quadrature to evaluate BCSZ $\psi_{i,\mu}$, $\psi_{i,\eta}$, and $\psi_{i,M}$ is required to apply the BCSZ discretization to more refined spatial meshes.

III.C. Nonlinear Iteration

Ultimately, we seek to solve Eq. (13) by finding the strictly nonnegative, nonlinear BCSZ solution representation $\tilde{\Psi}_{BCSZ}(s, t)$. The fundamental quantity describing $\tilde{\Psi}_{BCSZ}$ is the vector of unknowns $\vec{\Psi}_{BCSZ}$ that describes the bilinear function $\hat{\psi}_{BCSZ}(s, t)$. Thus, solving for $\vec{\Psi}_{BCSZ}$ is equivalent to finding the $\tilde{\Psi}_{BCSZ}(s, t)$ that satisfies Eq. (13).

In practice, however, we do not directly solve Eq. (13) for $\vec{\Psi}_{BCSZ}$; we solve a related, scaled, nonlinear system of equations,

$$\vec{G}(\vec{\Psi}_{BCSZ}) - \vec{R}_{BLD} = \vec{0}, \quad (29)$$

where $\vec{\Psi}_{BCSZ}$ is the vector of unknown we are solving for. In Eq. (29), $\vec{\Psi}_{BCSZ}$ is a linearly scaled variant of $\tilde{\Psi}_{BCSZ}$,

$$\vec{\Psi}_{BCSZ} = \Lambda \vec{\Psi}_{BCSZ}, \quad (30)$$

where Λ is a diagonal matrix with constant coefficients.

As noted in Ref. 15, angular flux solutions can easily vary by several orders of magnitude in the same problem. This large variation in solution scale causes two problems for the BCSZ scheme. First, a universal prescription for how to choose the perturbation factor to evaluate a Jacobian of Eq. (13) via finite difference is difficult or impossible to find.¹⁵ Second, in using finite difference methods to approximate the Jacobian of a nonlinear system of equations, poorly scaled unknowns can introduce significant numerical round-off errors. To see how solving for poorly scaled unknowns can affect iteration performance, consider the first-order, finite difference approximation of the derivative of smooth scalar function $h(u)$:

$$\frac{dh}{du} \Big|_{u=u_0} \approx \frac{h(u_0 + \epsilon) - h(u_0)}{\epsilon}, \quad (31)$$

where ϵ is a small number relative to u_0 . In the limit as $\epsilon \rightarrow 0$ Eq. (31) is the definition of the derivative of h with

respect to u . However, consider Eq. (31) with numerical noise:

$$\frac{dh}{du} \Big|_{u=u_0} \approx \frac{h(u_0 + \epsilon) - h(u_0) + \xi}{\epsilon}, \quad (32)$$

where ξ is a source of numerical noise that is not a linear function of ϵ . ξ could be the result of finite machine precision, or if $h(u)$ is the result of inexact numerical integration—like the quadrature integration inherent to the BCSZ scheme— ξ could be the truncation error independent of ϵ . Then,

$$\lim_{\epsilon \rightarrow 0} \left[\frac{h(u_0 + \epsilon) - h(u_0) + \xi}{\epsilon} \right] = \frac{\xi}{0} = \infty. \quad (33)$$

Assuming ϵ is some small fraction of u_0 , $\epsilon \rightarrow 0$ if and only if $u_0 \approx 0$. However, for systems of equations that vary over several orders of magnitude, like the unscaled BCSZ equations, $\epsilon \approx 0$ can be common. Thus, we do not solve Eq. (13) for its fundamentals unknown $\vec{\Psi}_{BCSZ}$ directly. Rather, we solve its well-scaled variation, Eq. (29), for $\vec{\Psi}_{BCSZ}$.

We define $\vec{G}(\vec{\Psi}_{BCSZ})$ to be

$$\vec{G}(\vec{\Psi}_{BCSZ}) = \vec{F}(\tilde{\Psi}_{BCSZ}[\Lambda \vec{\Psi}_{BCSZ}]), \quad (34)$$

where $\tilde{\Psi}_{BCSZ}[\Lambda \vec{\Psi}_{BCSZ}]$ is the BCSZ solution representation $\tilde{\Psi}_{BCSZ}$ associated with the bilinear function $\hat{\psi}_{BCSZ}$, described by coefficients $\vec{\Psi}_{BCSZ} = \Lambda \vec{\Psi}_{BCSZ}$. In our work, we use the UBLD solution as the diagonal elements of Λ :

$$\Lambda_{ij} = \begin{cases} 0 & i \neq j \\ \vec{\Psi}_{UBLD,i} & i = j \end{cases}. \quad (35)$$

Our initial iterate $\vec{\Psi}^{(0)}$ is unity:

$$\vec{\Psi}_{BCSZ}^{(0)} = [1 \ 1 \ 1]^T. \quad (36)$$

To solve Eq. (29), we use the damped Newton iteration,¹⁶ evaluating the Jacobian matrix using finite difference methods. For a given Newton iteration ℓ ,

$$\vec{\Psi}_{BCSZ}^{(\ell)} = \Lambda \vec{\Psi}_{BCSZ}^{(\ell)}. \quad (37)$$

Our damped Newton iteration progresses as

$$\vec{\Delta}^{(\ell)} = -\mathbf{J}^{-1} \vec{r}^{(\ell)} \quad (38a)$$

and

$$\vec{\Psi}_{\text{BCSZ}}^{(\ell+1)} = \vec{\Psi}_{\text{BCSZ}}^{(\ell)} + \omega \vec{\Delta}^{(\ell)}, \quad (38\text{b})$$

where

$$\vec{r}^{(\ell)} = \vec{G}(\vec{\Psi}^{(\ell)}) - \vec{R}_{\text{BLD}}, \quad (39)$$

which is the residual associated with well-scaled iterate $\vec{\Psi}_{\text{BCSZ}}^{(\ell)}$:

\mathbf{J} = 4 × 4 Jacobian matrix of the nonlinear system evaluated at $\vec{\Psi}_{\text{BCSZ}}^{(\ell)}$

$\vec{\Delta}^{(\ell)}$ = undamped Newton update

ω = scalar damping parameter, $\omega \in (0, 1]$.

We evaluate \mathbf{J} using a first-order finite difference scheme:

$$\mathbf{J} = \left[\frac{1}{\epsilon_1} (\vec{r}^{(\ell),1} - \vec{r}^{(\ell)}) \left| \frac{1}{\epsilon_2} (\vec{r}^{(\ell),2} - \vec{r}^{(\ell)}) \right| \frac{1}{\epsilon_3} (\vec{r}^{(\ell),3} - \vec{r}^{(\ell)}) \right| \frac{1}{\epsilon_4} (\vec{r}^{(\ell),4} - \vec{r}^{(\ell)}) \right], \quad (40)$$

where $\vec{r}^{(\ell),k}$ is the residual associated with perturbed iterate $\vec{\Psi}_{\text{BCSZ}}^{(\ell),k}$. The elements of the perturbed iterate $\vec{\Psi}_{\text{BCSZ},j}^{(\ell),k}$ are defined as

$$\vec{\Psi}_{\text{BCSZ},j}^{(\ell),k} = \begin{cases} \vec{\Psi}_{\text{BCSZ},j}^{(\ell)} + \epsilon_k & j = k \\ \vec{\Psi}_{\text{BCSZ},j}^{(\ell)} & \text{otherwise} \end{cases}. \quad (41)$$

In Eqs. (40) and (41), ϵ_k is a small perturbation of $\vec{\Psi}_{\text{BCSZ}}^{(\ell),k}$. To ensure that elements of a perturbed iterate do not change sign relative to the current iterate of $\vec{\Psi}_{\text{BCSZ}}^{(\ell)}$, we define

$$\epsilon_k = 10^{-3} \left[\max \left(1, \left| \vec{\Psi}_{\text{BCSZ},k}^{(\ell)} \right| \right) \right] \text{sg}(\vec{\Psi}_{\text{BCSZ},k}^{(\ell)}), \quad (42)$$

with

$$\text{sg}(x) = \begin{cases} 1 & x \geq 0 \\ -1 & x < 0 \end{cases}. \quad (43)$$

Damping parameter ω is initialized to 1 at the start of the BCSZ solve and generally does not need to be decreased. However, if an iteration is slow to converge or is divergent, as indicated by requiring a large number of iterations, $\ell > \ell_{\text{cut}}$, the following may be stated:

1. The BCSZ nonlinear iteration is paused.
2. The ℓ is reset to zero.

3. The well-scaled iterate $\vec{\Psi}_{\text{BCSZ}}$ is reset to unity.
4. ω is decreased (typically by a factor of 0.5 to 0.75).
5. ℓ_{cut} is increased (typically by a factor of 2).
6. The BCSZ nonlinear iteration resumes.

As noted in Refs. 5 and 15, solving the small, local, nonlinear problem for each cell/group/direction solve during a sweep precludes the use of Krylov methods in the scattering source iteration. However, the purpose of this work is to demonstrate the viability, accuracy, and robustness of a strictly nonnegative, nonlinear Petrov-Galerkin DFEM. For a typical single-cell, single-energy-group, single-direction solve, BCSZ requires on the order of eight to ten Newton iterations to reduce the nonlinear residual ten orders of magnitude. A single Newton iteration requires five evaluations of $\vec{G}(\vec{\Psi}_{\text{BCSZ}}^{(\ell)})$. Evaluating the residual is by far the most time-consuming operation in this problem. Although calculating the residual requires a significant amount of computational work, all residuals can be computed independently. This allows for the possibility of parallel computation, which would greatly reduce the calculation time.

IV. LOCALLY CONSERVATIVE FLUX FIXUPS

We now define two fixup schemes compatible with BLD differencing of the transport equation on quadrilaterals. By “compatible,” we mean that the fixups yield bilinear functions. Both methods are defined to be locally conservative and explicitly set any negative nodes of the UBLD solution to zero. The first method follows in the line of fixups used in Refs. 4 and 6 but has been generalized by Smedley-Stevenson¹⁷ to be embarrassingly parallel within the innermost solve of a transport sweep. Further, this fixup can be readily extended to guarantee strictly nonnegative solutions at the interpolation points of arbitrary order DFEM schemes defined using interpolatory polynomial basis functions. We refer to this flux fixup option as the generalized fixup (GFU). The second fixup seeks to explicitly preserve as many finite element spatial moments as possible while explicitly setting any negative UBLD nodal values to zero. Since we are attempting to preserve as many finite element moments as possible with the second fixup, we refer to this flux fixup as the maximum moment preserving fixup (MMP).

Each fixup generates a bilinear solution representation. Using the GFU scheme as an example,

$$\tilde{\psi}_{\text{GFU}}(s, t) = \sum_{i=1}^4 \psi_{\text{GFU},i} B_i(s, t) \quad (44\text{a})$$

and

$$\vec{\psi}_{\text{GFU}} = [\psi_{\text{GFU},1} \ \psi_{\text{GFU},2} \ \psi_{\text{GFU},3} \ \psi_{\text{GFU},4}]^T , \quad (44b)$$

where replacing GFU with MMP in Eq. (44) defines the MMP spatial discretization. The GFU and MMP schemes differ on how they define their respective vectors of unknown coefficients, $\vec{\psi}_{\text{GFU}}$ and $\vec{\psi}_{\text{MMP}}$.

IV.A. Generalized Fixup

The GFU scheme defines

$$\vec{\psi}_{\text{GFU}} = c_{\text{scale}} \vec{\psi}_{\text{shape}} , \quad (45)$$

where c_{scale} is a scalar scaling coefficient required to preserve particle balance and $\vec{\psi}_{\text{shape}}$ is the shape of the UBLD solution with all negativities set to zero. Specifically,

$$\vec{\psi}_{\text{shape},j} = \begin{cases} \vec{\psi}_{\text{UBLD},j} & \vec{\psi}_{\text{UBLD},j} > 0 \\ 0 & \text{otherwise} \end{cases} . \quad (46)$$

The scaling coefficient c_{scale} is defined as

$$c_{\text{scale}} = \frac{R_{\text{collapsed}}}{\vec{C} \cdot \vec{\psi}_{\text{shape}}} , \quad (47)$$

where

$$R_{\text{collapsed}} = \sum_{i=1}^4 R_{\text{BLD},i} , \quad (48)$$

which is the summation of the bilinear, interpolatory spatial moments of all volumetric sources and upwinded cell inflows \vec{R}_{BLD} ; $\vec{C} = 4 \times 1$ vector, where each entry \vec{C}_j is the summation of column j of \mathbf{A}_{UBLD} :

$$\vec{C}_j = \sum_{i=1}^4 \mathbf{A}_{\text{UBLD},ij} . \quad (49)$$

Since the interpolatory basis functions sum to unity, collapsing the columns of \mathbf{A}_{UBLD} into \vec{C} is equivalent to multiplying the transport equation by the constant basis function, $B_c(s, t) = 1$, and integrating, generating a statement of particle balance. Thus, Eq. (47) can be viewed as the scaling of any bilinear function required to maintain particle balance. As such, if $\vec{\psi}_{\text{UBLD}}$ is strictly nonnegative, $c_{\text{scale}} = 1$, $\vec{\psi}_{\text{shape}} = \vec{\psi}_{\text{UBLD}}$, and $\vec{\psi}_{\text{GFU}} = \vec{\psi}_{\text{UBLD}}$.

IV.B. Maximum Moment Preserving Fixup

The MMP scheme is defined to preserve particle balance, explicitly set negative UBLD nodes to zero and

preserve the maximum number of Legendre spatial moments while yielding a determinate system of equations. Like GFU, if the UBLD solution is everywhere positive, $\vec{\psi}_{\text{MMP}} = \vec{\psi}_{\text{UBLD}}$. Assuming that at least one UBLD node is negative, $\vec{\psi}_{\text{MMP}}$, which describes bilinear function $\tilde{\Psi}_{\text{MMP}}$, is found by solving

$$\mathbf{A}_{\text{MMP}} \vec{\psi}_{\text{MMP}} = \vec{R}_{\text{MMP}} . \quad (50)$$

The elements of \mathbf{A}_{MMP} and \vec{R}_{MMP} depend on the number and location of the negativities of the UBLD solution and are such that $\mathbf{A}_{\text{MMP}} \neq \mathbf{A}_{\text{UBLD}}$ and $\vec{R}_{\text{MMP}} \neq \vec{R}_{\text{BLD}}$. The MMP scheme is designed to preserve the maximum number of Legendre spatial moments, B_c , B_s , B_t , and B_{st} , where

$$B_c = 1 , \quad (51a)$$

$$B_s = s , \quad (51b)$$

$$B_t = t , \quad (51c)$$

and

$$B_{st} = st . \quad (51d)$$

As mentioned in the derivation of the GFU scheme, the physically meaningful B_c spatial moment, particle balance, is only preserved by the interpolatory basis spatial moments, B_1 , B_2 , B_3 , and B_4 , when all interpolatory moments are preserved. Likewise, the B_s , B_t , and B_{st} moments are only preserved by the interpolatory basis function moments when all interpolatory moments are preserved. Assuming that the most accuracy is obtained by preserving as many whole moments as possible, we choose to preserve as many Legendre spatial moments as possible with the MMP scheme rather than preserving interpolatory spatial moments.

The first step in applying the MMP fixup is to convert the interpolatory spatial moments stored in \mathbf{A}_{UBLD} and \vec{R}_{BLD} to the Legendre spatial moments, to be stored in $\mathbf{A}_{\text{UBLD}}^{\text{Leg}}$ and $\vec{R}_{\text{BLD}}^{\text{Leg}}$. It must be noted that we are finding the Legendre spatial moments in terms of interpolatory unknowns, such that

$$\mathbf{A}_{\text{UBLD}}^{\text{Leg}} \vec{\psi}_{\text{UBLD}} = \vec{R}_{\text{BLD}}^{\text{Leg}} , \quad (52)$$

where $\vec{\psi}_{\text{UBLD}}$ is the same as in Eq. (12). Equation (52) implies that there exists a matrix \mathbf{T} such that

$$\mathbf{T} \mathbf{A}_{\text{UBLD}} = \mathbf{A}_{\text{UBLD}}^{\text{Leg}} \quad (53a)$$

and

$$\mathbf{T} \vec{R}_{\text{BLD}} = \vec{R}_{\text{BLD}}^{\text{Leg}} . \quad (53b)$$

The interpolatory moments can be expressed as

$$\begin{aligned} \frac{1}{4} \begin{bmatrix} 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ s \\ t \\ st \end{bmatrix} &= \frac{1}{4} \begin{bmatrix} 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \\ &\times \begin{bmatrix} B_c(s, t) \\ B_s(s, t) \\ B_t(s, t) \\ B_{st}(s, t) \end{bmatrix} = \begin{bmatrix} B_1(s, t) \\ B_2(s, t) \\ B_3(s, t) \\ B_4(s, t) \end{bmatrix}. \end{aligned} \quad (54)$$

Therefore,

$$\mathbf{T} = 4 \begin{bmatrix} 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ -1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}. \quad (55)$$

Having found $\mathbf{A}_{\text{UBLD}}^{\text{Leg}}$ and $\vec{R}_{\text{UBLD}}^{\text{Leg}}$, we now define \mathbf{A}_{MMP} and \mathbf{R}_{MMP} . Noting that it is impossible to yield an everywhere negative UBLD solution (four negative nodes) with physical (non-negative) volumetric sources and inflow angular fluxes, we must define \mathbf{A}_{MMP} and \vec{R}_{MMP} for a total of five possible cases:

1. UBLD solution has one negative node.
2. UBLD solution has two negative nodes adjacent to one another on an edge parallel to the s -axis.
3. UBLD solution has two negative nodes adjacent to one another on an edge parallel to the t -axis.
4. UBLD solution has two negative nodes on opposite corners.
5. UBLD solution has three negative nodes.

For $\hat{\psi}_{\text{MMP}}$ to be unique, we must choose a total of four independent linear equations that $\hat{\psi}_{\text{MMP}}$ will satisfy. One equation will always be the particle balance equation, or B_c moment equation. Each negativity in the UBLD solution determines one equation of the MMP system of equations, explicitly stating that the negative

UBLD node that was negative will be zero in the MMP solution representation. The remaining $3 - N_{\text{UBLD},\text{neg}}$ equations that close Eq. (50) and define $\hat{\psi}_{\text{MMP}}(s, t)$ are chosen from B_s , B_t , and/or B_{st} Legendre spatial moments, where $N_{\text{UBLD},\text{neg}}$ is the number of negative nodes in the UBLD solution. If the UBLD solution has three negative nodes, the MMP system of equations is fully determined by preserving particle balance and explicitly setting all nodes that were negative in the UBLD solution representation to zero. The higher-order Legendre spatial moments preserved by the MMP scheme for a given UBLD solution case are given in Table II.

To demonstrate how we form \mathbf{A}_{MMP} and \vec{R}_{MMP} , we build \mathbf{A}_{MMP} and \vec{R}_{MMP} for the case of the UBLD solution representation having three negative nodes in Sec. IV.B.1 and for the case of the UBLD solution having one negative node in Sec. IV.B.2. Both examples will assume $\mathbf{A}_{\text{UBLD}}^{\text{Leg}}$ and $\vec{R}_{\text{UBLD}}^{\text{Leg}}$ of the forms

$$\mathbf{A}_{\text{UBLD}}^{\text{Leg}} = \begin{bmatrix} a_{c,1} & a_{c,2} & a_{c,3} & a_{c,4} \\ a_{s,1} & a_{s,2} & a_{s,3} & a_{s,4} \\ a_{t,1} & a_{t,2} & a_{t,3} & a_{t,4} \\ a_{st,1} & a_{st,2} & a_{st,3} & a_{st,4} \end{bmatrix} \quad (56)$$

and

$$\vec{R}_{\text{UBLD}}^{\text{Leg}} = [r_c \ r_s \ r_t \ r_{st}]^T. \quad (57)$$

Each column of $\mathbf{A}_{\text{UBLD}}^{\text{Leg}}$ contains the contributions of the corresponding $\vec{\psi}_{\text{UBLD},j}^{\text{Leg}}$ to every Legendre moment. Each row of $\mathbf{A}_{\text{UBLD}}^{\text{Leg}}$ contains the noted Legendre moment. For example, the first row of $\mathbf{A}_{\text{UBLD}}^{\text{Leg}}$ contains the respective contributions to the B_c spatial moment, the second row contains contributions to the B_s moment, and so on. Likewise, each element of $\vec{R}_{\text{UBLD}}^{\text{Leg}}$ contains the analogous spatial moments of the volumetric sources and upwinded angular fluxes.

By design, the MMP scheme guarantees only that the nodes in the UBLD solution that were negative will be

TABLE II
Additional Spatial Moments Preserved by MMP Scheme

UBLD Negativities	Higher Legendre Spatial Moments Preserved
One negative node ^a Two negative nodes, same edge, edge parallel to s -axis Two negative nodes, same edge, edge parallel to t -axis Two negative nodes, opposite corners	B_s and B_t B_s B_t B_{st}

^aAn MMP scheme explicitly specified in Sec. IV.B.2.

zero in the MMP solution representation, particle balance will be preserved, and the MMP bilinear function will satisfy the chosen higher-order Legendre moments. Our computational results will show that while this usually leads to a strictly nonnegative solution, a strictly nonnegative solution is not guaranteed with the MMP scheme. MMP only guarantees that if a node was negative in the UBLD solution representation, it will be zero in the MMP solution representation.

IV.B.1. MMP System of Equations with Three Negative UBLD Nodes

We now build \mathbf{A}_{MMP} and $\vec{\mathbf{R}}_{\text{MMP}}$ equation by equation for the case when the UBLD solution has three negative nodes. As we select equations, unknown coefficients (represented by question marks) of \mathbf{A}_{MMP} and $\vec{\mathbf{R}}_{\text{MMP}}$ will be replaced by known coefficients.

In this example, we assume that $\psi_{\text{UBLD},1}$, $\psi_{\text{UBLD},3}$, and $\psi_{\text{UBLD},4}$ are negative. The MMP scheme will explicitly set the nodes of the UBLD solution that are negative explicitly to zero in the MMP solution:

$$\mathbf{A}_{\text{MMP}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ ? & ? & ? & ? \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (58)$$

and

$$\vec{\mathbf{R}}_{\text{MMP}} = [0 ? 0 0]^T. \quad (59)$$

The final equation chosen to close the system enforces particle balance:

$$\mathbf{A}_{\text{MMP}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ a_{c,1} & a_{c,2} & a_{c,3} & a_{c,4} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (60)$$

and

$$\vec{\mathbf{R}}_{\text{MMP}} = [0 r_c 0 0]^T. \quad (61)$$

IV.B.2. MMP System of Equations with One Negative UBLD Node

We now build \mathbf{A}_{MMP} and $\vec{\mathbf{R}}_{\text{MMP}}$ for the case of one negative node. Using the same convention as before, unknown values of \mathbf{A}_{MMP} and $\vec{\mathbf{R}}_{\text{MMP}}$ are represented by question marks until defined; we assume that $\psi_{\text{UBLD},4}$ is the

negative UBLD node. Explicitly setting node four to zero defines one linear equation:

$$\mathbf{A}_{\text{MMP}} = \begin{bmatrix} ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (62)$$

and

$$\vec{\mathbf{R}}_{\text{MMP}} = [? ? ? 0]^T. \quad (63)$$

Preserving particle balance provides one more equation:

$$\mathbf{A}_{\text{MMP}} = \begin{bmatrix} a_{c,1} & a_{c,2} & a_{c,3} & a_{c,4} \\ ? & ? & ? & ? \\ ? & ? & ? & ? \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (64)$$

and

$$\vec{\mathbf{R}}_{\text{MMP}} = [r_c ? ? 0]^T, \quad (65)$$

and preserving the B_s and B_t spatial moments fully closes the system,

$$\mathbf{A}_{\text{MMP}} = \begin{bmatrix} a_{c,1} & a_{c,2} & a_{c,3} & a_{c,4} \\ a_{s,1} & a_{s,2} & a_{s,3} & a_{s,4} \\ a_{t,1} & a_{t,2} & a_{t,3} & a_{t,4} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (66)$$

and

$$\vec{\mathbf{R}}_{\text{MMP}} = [r_c r_s r_t 0]^T. \quad (67)$$

V. NUMERICAL RESULTS

We now present computational results for a simple test problem, a $10[\text{cm}] \times 10[\text{cm}]$ void, with an incident angular flux of $\frac{\pi}{3} [\text{n}/(\text{cm}^2 \cdot \text{s} \cdot \text{sr})]$ only in the direction of $\mu = 0.868890300722$, $\eta = 0.350021174582$ along the bottom edge. The primary benefit of this test problem is that it challenges most numerical methods. However, near voids can and do arise in practical applications, e.g., gas-cooled nuclear reactors, and all but the lowest-resolution angular quadrature sets will have directions that are of glancing incidence to spatial zones, particularly when considering unstructured meshes.

In Fig. 6 we plot the UBLD scalar flux solution on a uniform, orthogonal mesh with 64×64 cells. The cell average scalar fluxes are plotted on a logarithmic scale. We plot the FLBLD, BCSZ, GFU, and MMP scalar flux

solutions for the same problem in Fig. 7. The logarithmic scale is used to accent any negativities, oscillations, or numerical diffusion possibly present in the different

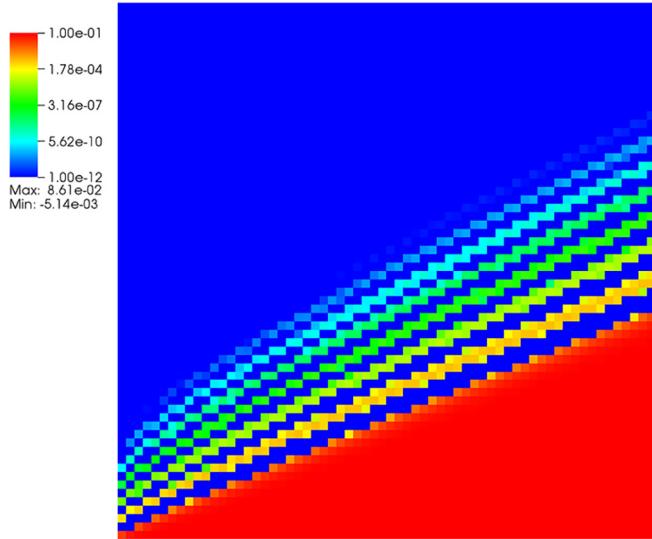


Fig. 6. UBLD cell average scalar flux solution on a 64×64 cell orthogonal mesh.

spatial discretizations we consider. In Fig. 6, the alternating dark blue stripes above the red region are locations where the UBLD zone average flux is negative. Though undesirable, the negativities and oscillations of the UBLD scheme dampen rapidly.

In Fig. 7, we observe that for this test problem, the FLBLD, BCSZ, and GFU solutions are all strictly non-negative. The inaccuracy of the FLBLD scheme is readily apparent in the large numerical diffusion shown in Fig. 7a. Both the BCSZ and GFU schemes exhibit very little numerical diffusion. However, very near the discontinuity, the GFU solution exhibits slightly larger magnitude cell average angular fluxes than the BCSZ scheme, causing it to be less accurate than BCSZ.

Interestingly, the MMP solution does not yield strictly nonnegative solutions, as indicated by the scattered dark blue squares within the green-colored region of Fig. 7d. Though it seems counterintuitive that a method designed to yield strictly nonnegative solutions yields negative solutions, one must remember that the MMP scheme is guaranteed only to set negative nodes of the UBLD solution to zero. The MMP solution sacrifices strictly positive

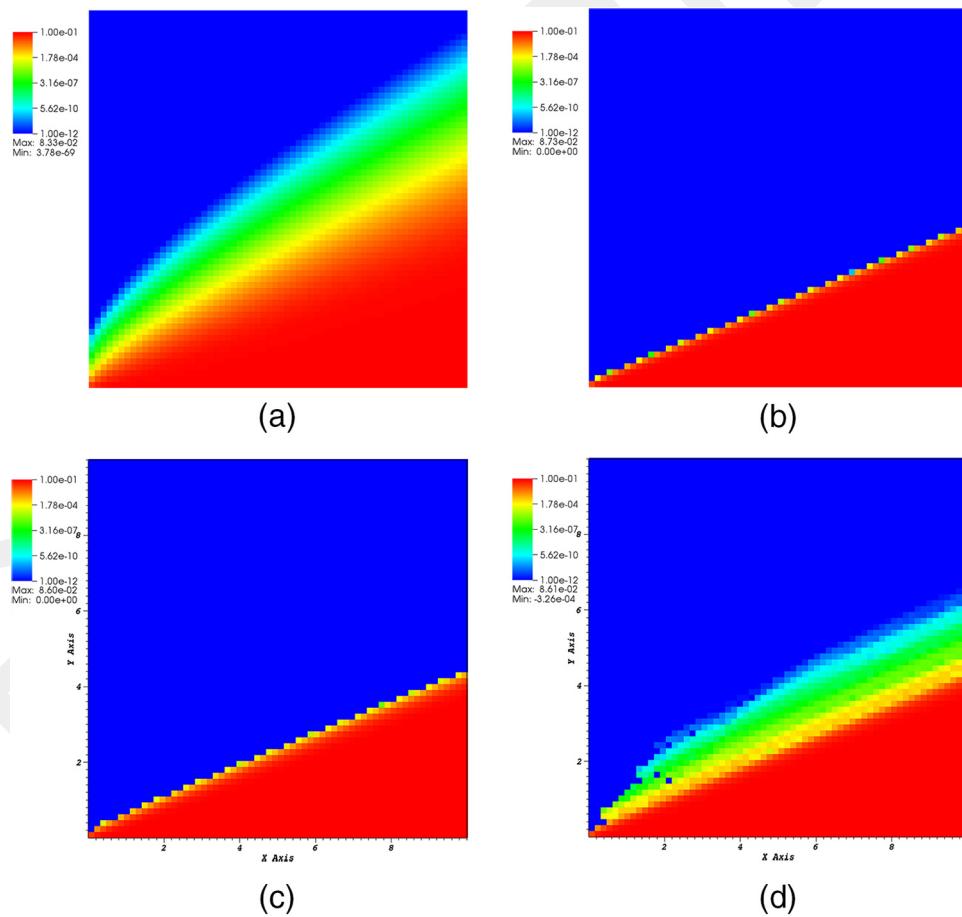


Fig. 7. Cell average scalar flux solutions on an orthogonal mesh of 64×64 cells for alternatives to the UBLD spatial discretization: (a) FLBLD solution, (b) BCSZ solution, (c) GFU solution, and (d) MMP solution.

solutions to preserve the s , t , and/or st Legendre spatial moments. We have manually verified that the negative nodal MMP solutions occur only at nodes that were positive in the UBLD solution. Since the MMP solution can yield negative outflows, for simplicity, we chose to set these cells to zero. In our testing, preserving moments other than those listed in Table II did not eliminate the possibility of the MMP scheme producing negative angular flux solutions. Additionally, the MMP scheme exhibits more numerical diffusion than the BCSZ scheme.

In Figs. 8 and 9, we plot the same solutions as shown in Figs. 6 and 7, respectively, but on a randomly perturbed grid where the structured grid interior vertices have been moved by a distance equal to $\leq 25\%$ of the initial, structured grid cell width. All discretizations use the same mesh.

The results on the distorted mesh largely replicate the trends of the more refined orthogonal mesh:

1. The UBLD solution oscillates and generates negative solutions.
2. The FLBLD solution is strictly nonnegative for this problem but is very numerically diffusive.
3. The MMP solution is not strictly nonnegative and is numerically diffusive.
4. The GFU and BCSZ solutions are both strictly nonnegative and exhibit minimal numerical diffusion.

Finally, we quantify the errors of each method for the glancing void problem on the orthogonal mesh. In Fig. 10, we plot E_{ϕ_A} ,

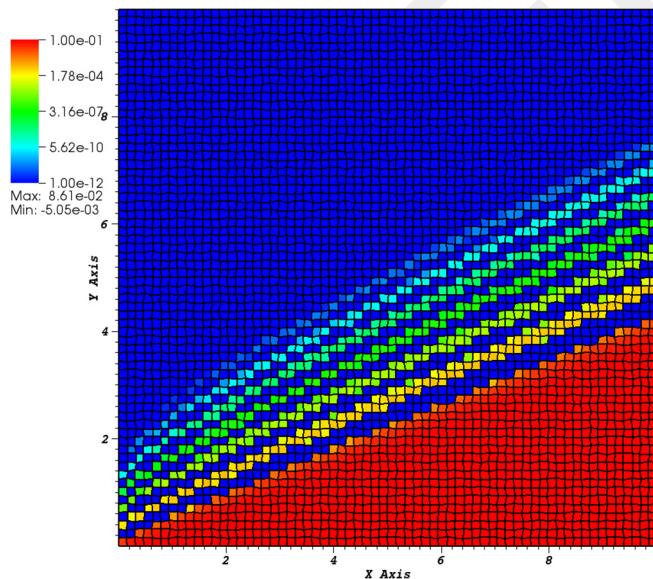


Fig. 8. UBLD cell average scalar flux solution on a distorted mesh of 64×64 cells. Interior vertices perturbed by 25% of the unperturbed mesh spacing.

$$E_{\phi_A} = \sqrt{\sum_{K=1}^{N_{cells}} \Delta x_K \Delta y_K (\tilde{\phi}_{A,K} - \phi_{A,K}^{\text{exact}})^2}, \quad (68)$$

an L_2 -like norm of the average scalar flux, as a function of average cell width Δx . In Eq. (68), $\tilde{\phi}_{A,K}$ is the numerical cell average scalar flux for a particular scheme in cell K , $\phi_{A,K}^{\text{exact}}$ is the analytic cell average scalar flux in cell K , N_{cells} is the total number of mesh cells, and

$$\Delta x_K = \Delta y_K = \frac{10[\text{cm}]}{\sqrt{N_{cells}}}. \quad (69)$$

From Fig. 10, we see that BCSZ is the most accurate method. Interestingly, the BCSZ and MMP both calculate the exact zone average scalar flux for one mesh cell. After only a small amount of refinement, the UBLD, GFU, and MMP solutions are essentially identical in overall accuracy. Since the BCSZ, GFU, and MMP methods all yield $\tilde{\phi}_{\text{UBLD}}$ when the UBLD solution is everywhere non-negative in a cell, we expect all methods to converge to the same solution. All methods are converging $E_{\phi_A} \propto \Delta x^{1/4}$. The low rate of convergence is due to the singularity of the analytic solution.¹⁸

VI. CONCLUSIONS

We have compared four alternative spatial discretizations to UBLD for a test problem where the UBLD solution contains nonphysical, negative angular flux solutions. The nonlinear, strictly nonnegative, Petrov-Galerkin DFEM scheme BCSZ yielded strictly nonnegative solutions. Because it maintains all of the bilinear moments, the BCSZ scheme is more accurate than all other discretizations considered. However, the increased accuracy of BCSZ requires a significant increase in computational work to solve the nonlinear BCSZ equations. The GFU fixup also yielded strictly nonnegative solutions but was less accurate than BCSZ, UBLD, and the MMP fixup. Though the MMP fixup calculated the exact cell average scalar flux on a one-zone mesh, the MMP scheme can yield negative angular flux solutions, is less accurate than the BCSZ scheme, and should no longer be considered. The FLBLD scheme yielded strictly nonnegative solutions for the glancing void test problem but was the least accurate due to significant numerical dispersion.

We have intentionally omitted a detailed performance analysis of BCSZ. When necessary for a given zone, group, and direction angular flux solve, BCSZ is easily an order of magnitude more computationally intense than UBLD. BCSZ requires several Newton iterations, each requiring the solution of a 4×4 linear system of equations. For comparison, UBLD (and DFEM in

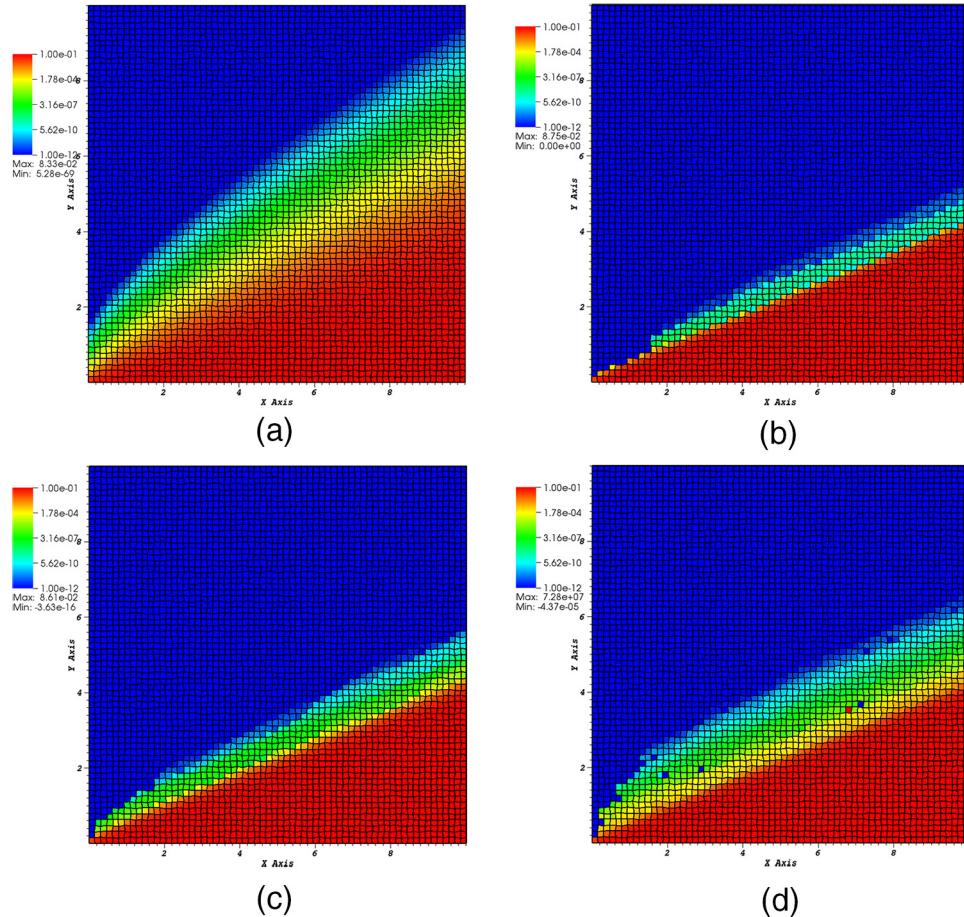


Fig. 9. Cell average scalar flux solutions on a distorted mesh of 64×64 cells for alternatives to the UBLD spatial discretization: (a) FLBLD solution, (b) BCSZ solution, (c) GFU solution, and (d) MMP solution. Interior vertices randomly perturbed 25% of unperturbed cell spacing.

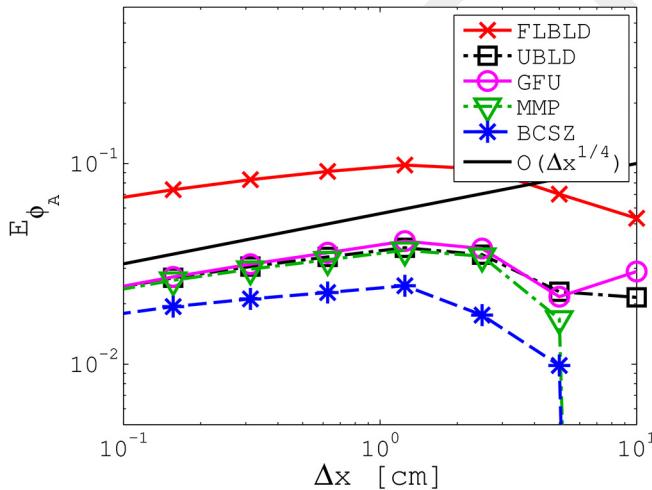


Fig. 10. Convergence of cell average scalar flux on orthogonal grids.

general) are considered computationally expensive relative to other spatial discretizations (diamond difference or upstream corner balance) by merely requiring the

solution of a single, dense linear system of equations.¹⁹ However, BCSZ does not need to be applied in every angular flux zone, direction, and group solve, and so its performance could be nearly equal to that of UBLD for certain problems. Further, a performance analysis alone cannot adequately account for BCSZ being able to run problems that fail with UBLD. For example, in nonlinear thermal radiative transfer problems, UBLD can generate sufficiently large-magnitude negative angular intensity solutions that the entire nonlinear solve fails, whereas BCSZ would yield strictly nonnegative solutions, and the simulation could continue. Prior to applying BCSZ to radiative transfer problems, iterative acceleration techniques compatible with BCSZ must be developed; just as the work of Bruss et al.¹⁵ accelerated the nonnegative modification to LD (Ref. 5), future work must include accelerating the BCSZ global solution process.

Two immediate opportunities present themselves for extending this work. The first is to investigate whether the adaptive quadrature integration developed for BCSZ can be extended to be adaptive in both s and t to permit higher-order

(biquadratic, bicubic, etc.) strictly nonnegative, nonlinear, Petrov-Galerkin DFEM without the need to determine the roots of an arbitrary degree bivariate polynomial. Finally, though our ultimate goal is to apply the BCSZ discretization to the thermal radiative transfer equations, the intermediate step of extending our localized BCSZ discretization to radiative diffusion applications is also of interest. Since the number of unknowns is much smaller for a radiative diffusion calculation as compared to a thermal radiative transfer calculation, we will be able to more tractably solve the global, nonlinear system of diffusion equations.

APPENDIX

ACRONYMS

BCSZ = bilinear consistent set-to-zero spatial discretization

BLD = bilinear (Q^1) discontinuous finite element (four unknowns per quadrilateral cell)

DFEM = discontinuous finite element method

FLBLD = fully lumped bilinear discontinuous finite element method

GFU = generalized fixup spatial discretization

LD = linear (P^1) discontinuous finite element method (three unknowns per quadrilateral cell)

MMP = maximum moment preserving spatial discretization

SCB = simple corner balance spatial discretization

S_N = discrete ordinates angular approximation (summation over N angles)

UBLD = unlumped bilinear discontinuous finite element method spatial discretization

Acknowledgments

The authors wish to thank W. D. Hawkins for his invaluable and timely assistance in implementing and testing these methods. At nonoverlapping times, the work of P. G. Maginot was funded by the U.S. Department of Energy (DOE) Computational Science Graduate Fellowship program, administered by the Krell Institute, under grant DE-FG02-97ER25308, or conducted under the auspices of DOE by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. The work of J. C. Ragusa and J. E. Morel was supported in part by the Center for Exascale Radiation Transport, under DOE, National Nuclear Security Administration, award number DE-NA0002376.



References

- K. D. LATHROP, "Spatial Differencing of the Transport Equation: Positivity vs. Accuracy," *J. Comput. Phys.*, **4**, 475 (1969); [http://dx.doi.org/10.1016/0021-9991\(69\)90015-1](http://dx.doi.org/10.1016/0021-9991(69)90015-1).
- E. W. LARSEN and W. F. MILLER, "Convergence Rates of Spatial Difference Equations for the Discrete-Ordinates Neutron Transport Equations in Slab Geometry," *Nucl. Sci. Eng.*, **73**, 76 (1980); <http://dx.doi.org/10.13182/NSE80-3>.
- M. L. ADAMS, "Discontinuous Finite Element Transport Solutions in Thick Diffusive Problems," *Nucl. Sci. Eng.*, **137**, 298 (2001); <http://dx.doi.org/10.13182/NSE00-41>.
- E. D. FICHTL, J. S. WARSA, and J. D. DENSMORE, "The Newton-Krylov Method Applied to Negative-Flux Fixup in S_N Transport Calculations," *Nucl. Sci. Eng.*, **165**, 331 (2010); <http://dx.doi.org/10.13182/NSE09-51>.
- P. G. MAGINOT, J. E. MOREL, and J. C. RAGUSA, "A Non-Negative Moment Preserving Spatial Discretization Scheme for the Linearized Boltzmann Transport Equation in 1-D and 2-D Cartesian Geometries," *J. Comput. Phys.*, **231**, 6801 (2012); <http://dx.doi.org/10.1016/j.jcp.2012.06.018>.
- S. HAMILTON, M. BENZI, and J. S. WARSA, "Negative Flux Fixups in Discontinuous Finite Elements S_N Transport," *Proc. Int. Conf. Mathematics, Computational Methods and Reactor Physics*, Saratoga Springs, New York, May 3–7, 2009, American Nuclear Society (2009).
- T. A. WAREING, E. W. LARSEN, and M. L. ADAMS, "Diffusion Accelerated Discontinuous Finite Element Schemes for the S_N Equations in Slab and X, Y Geometries," *Proc. Int. Topl. Mtg. Advances in Mathematics, Computations and Reactor Physics*, Pittsburgh, Pennsylvania, April 28–May 2, 1991, American Nuclear Society (1991).
- M. L. ADAMS, "Subcell Balance Methods for Radiative Transfer of Arbitrary Grids," *Transp. Theory Stat. Phys.*, **26**, 385 (1997); <http://dx.doi.org/10.1080/00411459708017924>.
- J. DONEÁ and A. HUERTA, *Finite Element Methods for Flow Problems*, Wiley (2003).
- A. GRENNBAUM and J. M. FERGUSON, "A Petrov-Galerkin Finite Element Method for Solving the Neutron Transport Equation," *J. Comput. Phys.*, **64**, 1, 97 (1986); [http://dx.doi.org/10.1016/0021-9991\(86\)90020-3](http://dx.doi.org/10.1016/0021-9991(86)90020-3).
- T. A. WAREING, "An Exponential Discontinuous Scheme for Discrete-Ordinates Calculations in Cartesian Geometries," *Proc. Joint Int. Conf. Mathematical Methods and Supercomputing for Nuclear Applications*, Saratoga Springs, New York, October 5–9, 1997, American Nuclear Society (1997).
- B. TURCKSIN and J. C. RAGUSA, "A Diffusion Synthetic Acceleration Scheme for Rectangular Geometries Based on Bilinear Discontinuous Finite Elements," *Proc. Int. Conf. Mathematics and Computational Methods Applied to*

- Nuclear Science and Engineering*, Sun Valley, Idaho, May 5–9, 2013, American Nuclear Society (2013).
- 13. MATLAB Release 2011a, The MathWorks Inc., Natick, Massachusetts.
 - 14. T. N. L. PATTERSON, “The Optimum Addition of Points to Quadrature Formulae,” *Math. Comput.*, **22**, 847 (1968); <http://dx.doi.org/10.1090/S0025-5718-68-99866-9>.
 - 15. D. E. BRUSS, J. E. MOREL, and J. C. RAGUSA, “S₂SA Preconditioning for the S_n Equations with Strictly Non-Negative Spatial Discretization,” *J. Comput. Phys.*, **273**, 706 (2014); <http://dx.doi.org/10.1016/j.jcp.2014.05.022>.
 - 16. J. NOCEDAL and S. J. WRIGHT, *Numerical Optimization*, 2nd ed., Springer (2000).
 - 17. R. SMEDLEY-STEVENSON, Personal Communication (2016).
 - 18. C. JOHNSON and J. PITKARANTA, “Convergence of a Fully Discrete Scheme for Two-Dimensional Neutron Transport,” *SIAM J. Numer. Anal.*, **20**, 5, 951 (1983); <http://dx.doi.org/10.1137/0720065>.
 - 19. P. G. MAGINOT et al., “The Improved Simple Corner Balance Method and Efforts to Enhance Its Computational Performance,” *Trans. Am. Nucl. Soc.*, **114**, 373 (2016).