Multilevel S_N for applications in radiation therapy

The idea of this project is to implement an efficient, but readable S_N code framework. Such a code framework can be used for different applications and we wish to first use it for problems in radiation therapy. Furthermore, we can use the framework to test the idea of multilevel S_N, which splits the spatial domain into grids with varying resolution and uses a fine quadrature set on coarse grids as well as coarse quadratures on fine grids[1]. Two main working "products" could be papers on *S_N for radiation therapy*, *Multilevel S_N*, *Effects of quadrature sets and triangular discretizations* as well as a functioning and documented software. In the following, we briefly mention the fundamentals of the S_N method and state the idea of multilevel S_N:

1 The discrete ordinates (S_N) method

The radiative transfer (or linear Boltzmann) equation reads

$$\partial_t \psi(t,\boldsymbol{x},\boldsymbol{\Omega}) + \boldsymbol{\Omega} \cdot \nabla \psi(t,\boldsymbol{x},\boldsymbol{\Omega}) + \sigma_a(\boldsymbol{x})\psi(t,\boldsymbol{x},\boldsymbol{\Omega}) = \sigma_s(\boldsymbol{x}) \int_{\mathbb{S}^2} k(\boldsymbol{\Omega}\cdot\boldsymbol{\Omega}')\left(\psi(t,\boldsymbol{x},\boldsymbol{\Omega}') - \psi(t,\boldsymbol{x},\boldsymbol{\Omega})\right) d\boldsymbol{\Omega}'.$$

(1)

The equation describes the dynamics of particles traveling through a background material. Particles are subject to absorption (described by the absorption cross section σ_a) as well as scattering (described by the scattering cross section σ_s as well as the collision kernel k). Note that for now, we omit the effects of sources. Also, when following the work of Kerstin [?], one often solves the continuous slowing down (CSD) equation for treatment planning, which is closely related to the radiative transfer equation. Hence, a starting point for obtaining a working code base could be (??).

In the field of radiation therapy, the radiative transfer equation is often discretized by Monte-Carlo methods, which commonly yield an increased runtime and stochastic noise. Therefore, we choose a discrete ordinates (S_N) discretization (which has its own disadvantages such as ray-effects) instead. The first step to obtain a discretized system is to derive a finite dimensional, nodal representation of the angular variable $\boldsymbol{\Omega} \in \mathbb{S}^2$. Here, different strategies to obtain an accurate discretization exist and we are going to start with a straight forward product quadrature rule on the sphere. To obtain a finite set of directions in $\mathbb{S}^2$, we describe the angular variable

$$\boldsymbol{\Omega} = (\cos\phi\sin\theta, \sin\phi\sin\theta, \cos\theta)^T$$

at finite positions of θ and ϕ. In the following, we choose the nodal positions of $\mu = \cos(\theta) \in [-1,1]$ to be the quadrature points of a Gauss quadrature rule with N_q quadrature points, while choosing an equally spaced grid for ϕ, i.e.

$$\phi_i = i\Delta\phi \quad \text{for} \quad i = 1, \ldots, N_q \quad \text{and} \quad \Delta\phi = \frac{2\pi}{N_q} \,.$$

Now, we have a finite number of nodal directions (also called ordinates) on the sphere, which we denote by Ω_k for $k = 1, \cdots, Q$, where $Q := N_q^2$. The solution at these ordinates is defined as $\psi_p(t,\boldsymbol{x}) := \psi(t,\boldsymbol{x},\boldsymbol{\Omega}_p)$. Furthermore, one can construct a quadrature rule

$$\int_{\mathbb{S}^2} \psi(t,\boldsymbol{x},\boldsymbol{\Omega}) \, d\boldsymbol{\Omega} \approx \sum_{q=1}^{Q} w_q \psi_q(t,\boldsymbol{x}).$$

---

[1]One could also think about going one step further by extending the multilevel idea to uncertainty quantification for multilevel S_N.

Now, evaluating the radiative transfer equation (**??**) at the chosen set of ordinates while making use of the above quadrature rule yields

$$\partial_t \psi_p(t, \boldsymbol{x}) + \boldsymbol{\Omega}_p \cdot \nabla \psi_p(t, \boldsymbol{x}) + \sigma_a(\boldsymbol{x})\psi_p(t, \boldsymbol{x}) = \sigma_s(\boldsymbol{x}) \sum_{q=1}^{Q} w_q k(\boldsymbol{\Omega}_p \cdot \boldsymbol{\Omega}_q) \left(\psi_q(t, \boldsymbol{x}) - \psi_p(t, \boldsymbol{x})\right).$$

This is now a system of Q advection equations with a coupling term on the right hand side. Such an equation can be solved in a straight forward manner with a finite volume discretization (we could also think about using DG). For this, we split the (for now two-dimensional) spatial domain into cells $V_{ij} = [x_{i-1/2}, x_{i+1/2}] \times [y_{j-1/2}, y_{j+1/2}]$ and describe the time evolution of the solution on these cells. Let us denote the solution in cell V_{ij} at time t_n as

$$\psi_{pij}^n := \frac{1}{|V_{ij}|} \int_{V_{ij}} \psi_p(t_n, \boldsymbol{x}) \, d\boldsymbol{x}.$$

At time $t_0 = 0$, the initial particle density is known. To obtain a scheme which time updates the solution, we (when leaving out certain details) replace the time derivative by a finite difference approximation

$$\partial_t \psi_p(t, \boldsymbol{x}_{ij})\Big|_{t=t_n} \approx \frac{\psi_{pij}^n - \psi_{pij}^n}{\Delta t} \,,$$

$$(\boldsymbol{\Omega}_p)_x \partial_x \psi_p(t^n, \boldsymbol{x})\Big|_{\boldsymbol{x}=\boldsymbol{x}_{ij}} \approx \frac{1}{|V_{ij}|} \left(g_{p,i+1/2,j}^n - g_{p,i-1/2,j}^n\right) \,,$$

$$(\boldsymbol{\Omega}_p)_y \partial_y \psi_p(t^n, \boldsymbol{x})\Big|_{\boldsymbol{x}=\boldsymbol{x}_{ij}} \approx \frac{1}{|V_{ij}|} \left(g_{p,i,j+1/2}^n - g_{p,i,j-1/2}^n\right) \,.$$

The function $g_{p,i+1/2,j}$ is called the numerical flux, which approximates the solution at the cell interface at $x_{i+1/2}$. One straight forward choice is the upwind flux

$$g_{p,i+1/2,j} := \begin{cases} (\boldsymbol{\Omega}_p)_x \psi_{pij}^n & \text{if } (\boldsymbol{\Omega}_p)_x > 0 \\ (\boldsymbol{\Omega}_p)_x \psi_{p,i+1,j}^n & \text{else} \end{cases} \,.$$

When replacing differential operators by these approximations, we obtain the first-order scheme

$$\psi_{pij}^{n+1} = \psi_{pij}^n - \frac{\Delta t}{|V_{ij}|} \left(g_{p,i+1/2,j} - g_{p,i-1/2,j} + g_{p,i,j+1/2} - g_{p,i,j-1/2}\right)$$

$$+ \Delta t \sigma_{s,ij} \sum_{q=1}^{N_q} w_q k(\boldsymbol{\Omega}_p \cdot \boldsymbol{\Omega}_q) \left(\psi_{q,ij}^n - \psi_{p,ij}^n\right) - \Delta t \sigma_{a,ij} \psi_{pij}^n \,. \tag{2}$$

This is the equation which needs to be implemented. It can be extended to higher order, but we should start with order one.

2 Physical model

3 Multilevel $\mathbf{S}_N$

In the following, we wish to apply ideas from multilevel Monte-Carlo as well as multilevel stochastic-Collocation to the S_N context. Usually, one is not interested in the angular flux ψ, but one rather wishes to determine some quantity of interest, which could for example be the radiation dose received by the patient or the angular flux in a certain region. Let us call this scalar quantity d, where the bracket denotes a spherical integration and $d : \mathbb{S}^2 \to \mathbb{R}$. We assume a quantity constant in time and space for sake of simplicity. Assume that we can approximate the

quantity d with our code using a certain spatial resolution $h := \Delta x = \Delta y$ as well as Q quadrature points. Then, when the output for resolution h and Q is denoted as $\tilde{d}_k^{h,Q}$ for $k = 1, \cdots, Q$. This output can be interpreted as a continuous grid function[2]

$$d_{h,Q}(\Omega) := \tilde{d}_k^{h,Q} \quad \text{with } k \text{ s.t. } \Omega_k \text{ is the closest ordinate to } \Omega.$$

Now the L^2 distance to the exact quantity of interest d can be divided into an quadrature as well as a spatial error. Adding and subtracting the term d_h inside the brackets, where $d_h : \mathbb{S}^2 \to \mathbb{R}$ uses an exact quadrature with a fixed spatial resolution h (i.e. $Q \to \infty$) yields

$$\|d_{h,Q} - d\|_{L^2(\mathbb{S}^2)} \leq \|d_{h,Q} - d_h\|_{L^2(\mathbb{S}^2)} + \|d_h - d\|_{L^2(\mathbb{S}^2)}.$$

Then, the first term is the quadrature error and the second term is the spatial discretization error. Assume that the following error estimates hold:

$$\|d_h - d\|_{L^2(\mathbb{S}^2)} \lesssim h^\alpha \simeq N_x^{-\alpha}$$
$$\|d_{h,Q} - d_h\|_{L^2(\mathbb{S}^2)} \lesssim Q^{-\beta}.$$

A certain accuracy ε can be obtained by bounding both error terms by $\varepsilon/2$, i.e. the number of spatial cells must be

$$N_x^{-\alpha} \simeq \varepsilon \quad \Rightarrow \quad N_x \simeq \varepsilon^{-1/\alpha}$$

and the number of quadrature points has to be

$$Q^{-\beta} \simeq \varepsilon \quad \Rightarrow \quad Q \simeq \varepsilon^{-1/\beta}$$

Assume the cost to run a simulation are $C = N_x^3 \cdot Q$. Then we get

$$C \simeq \varepsilon^{-\frac{3}{\alpha} - \frac{1}{\beta}}. \tag{3}$$

Now let us make a multilevel ansatz. For this we define L levels for the spatial grid resolution h_ℓ with $\ell = 1, \cdots, L$. Then, we can write $d_{h_L,Q}(\Omega)$, i.e. the quantity of interest for the finest resolution h_L with a certain number of quadrature points as

$$d_{h_L,Q}(\Omega) = d_{h_1,Q}(\Omega) + \sum_{\ell=2}^{L} \left(d_{h_\ell,Q}(\Omega) - d_{h_{\ell-1},Q}(\Omega)\right) = \sum_{\ell=1}^{L} Y_{\ell,Q}(\Omega) \tag{4}$$

where we use $Y_{1,Q} = d_{h_1,Q}$ and for $\ell \neq 1$ we use $Y_{\ell,Q} := d_{h_\ell,Q} - d_{h_{\ell-1},Q}$. Now we must pick an adequate quadrature for the difference terms $Y_{\ell,Q}$. Note that since (**??**) should converge for $L \to \infty$, we expect the difference terms on higher levels to fall to zero. This indicates requiring a less accurate quadrature rule on higher levels. Hence, let us define the number of quadrature points on L levels to be $Q_1 \geq \cdots \geq Q_L$, which gives the multilevel S_N (MLS$_N$) estimator

$$d_{(h)_\ell,(Q)_\ell}(\Omega) = d_{h_1,Q_1}(\Omega) + \sum_{\ell=2}^{L} \left(d_{h_\ell,Q_\ell}(\Omega) - d_{h_{\ell-1},Q_\ell}(\Omega)\right) = \sum_{\ell=1}^{L} Y_{\ell,Q_\ell}(\Omega). \tag{5}$$

From this starting point, we can again derive the required costs as done in (**??**). An optimal number of levels L as well as the resolution of quadrature and space discretization are then obtained by minimizing these costs for a fixed desired accuracy ε.

---

[2]A more elegant interpretation is possible when using a spherical harmonics reconstruction or a grid-based quadrature like the octahedron or icosahedron quadrature.

settings

IO

Quadrature

Physics
σ_s, σ_a

Quadrature
Grid

Solver

Meshical
interpolate

Mesh

MLSN

SN

?

sofort Kundenbasiert?