Solving the 3D steady Radiative Transfer Equation with FreeFem++

D. Le Hardy, Y. Favennec, B. Rousseau

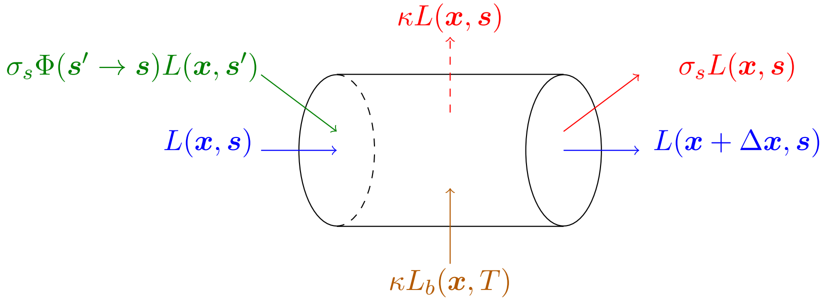
16th December 2015

Plan

1. Forward Model and Weak Formulation
2. Iterative Method and Parallelization Strategy
3. Validation
4. Conclusion

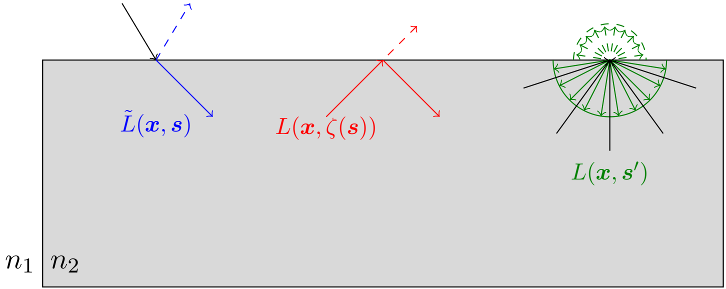
Radiative Transfer Equation

$$\underbrace{\boldsymbol{s}\cdot\nabla L(\boldsymbol{x},\boldsymbol{s})}_{\text{Transport}} + \underbrace{(\kappa+\sigma_s)L(\boldsymbol{x},\boldsymbol{s})}_{\text{Loss by absorbtion and scattering}} = \underbrace{\sigma_s\int_{4\pi}\Phi(\boldsymbol{s}'\to\boldsymbol{s})L(\boldsymbol{x},\boldsymbol{s}')\,\mathrm{d}\boldsymbol{s}'}_{\text{Gain by scattering}} + \underbrace{\kappa L_b(\boldsymbol{x},T)}_{\text{Emission}}$$



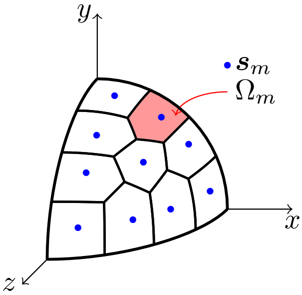
Boundary Conditions

$$L(\boldsymbol{x},\boldsymbol{s}) = \underbrace{\tilde{L}(\boldsymbol{x},\boldsymbol{s})}_{\text{Entering Radiance}} + (1-\alpha)\underbrace{\rho(\boldsymbol{s}\cdot\boldsymbol{n})L(\boldsymbol{x},\zeta(\boldsymbol{s}))}_{\text{Specular Reflection}} + \alpha \underbrace{\frac{1-\varepsilon}{\pi}\int_{\boldsymbol{s}'\cdot\boldsymbol{n}>0} L(\boldsymbol{x},\boldsymbol{s}')\boldsymbol{s}'\cdot\boldsymbol{n}\,\mathrm{d}\boldsymbol{s}'}_{\text{Diffuse Reflection}}$$



Discrete Ordinate Method

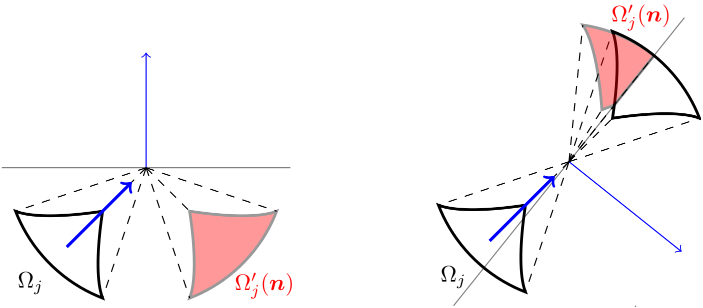
Angular Discretization



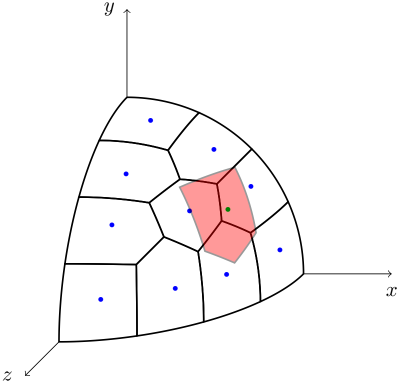
$$(\mathsf{SDS}_m) : \ (\boldsymbol{s}_m \cdot \nabla + \kappa + \sigma_s)\, L(\boldsymbol{x}, \boldsymbol{s}_m) - \sigma_s \sum_{j=1}^{\boldsymbol{N}_d} \omega_j L(\boldsymbol{x}, \boldsymbol{s}_j) \Phi_{m,j} = \kappa L_b(T)$$

with $\omega_m = \text{mes } \Omega_m$

Discrete Boundary Conditions



Discrete Boundary Conditions



$$L_m(\boldsymbol{x}) = \tilde{L}_m + \delta_{m,m}(\boldsymbol{n})L_m + \sum_{j\neq m} \delta_{m,j}(\boldsymbol{n})L_j$$

for $\boldsymbol{s}_m \cdot \boldsymbol{n} < 0$

Weak Formulation of (SDS$_m$) by the SUP-G finite element methods

$$
\begin{aligned}
&\int_{\mathcal{D}} (\boldsymbol{s}_m \cdot \nabla L_m)\,(\boldsymbol{s}_m \cdot \nabla v)\ \mathrm{d}\boldsymbol{x} - \int_{\mathcal{D}} \tilde{\beta}_m (\boldsymbol{s}_m \cdot \nabla L_m) v\ \mathrm{d}\boldsymbol{x} \\
&+ \int_{\partial\mathcal{D}^{m+}} \tilde{\beta}_m L_m v (\boldsymbol{s}_m \cdot \boldsymbol{n})\ \mathrm{d}\Gamma + \int_{\partial\mathcal{D}^{m-}} \tilde{\beta}_m \delta_{m,m}(\boldsymbol{n}) L_m v (\boldsymbol{s}_m \cdot \boldsymbol{n})\ \mathrm{d}\Gamma \\
&- \sum_{j \neq m} \left[\omega_j \Phi_{m,j} \int_{\mathcal{D}} \sigma_s L_j (\boldsymbol{s}_m \cdot \nabla v)\ \mathrm{d}\boldsymbol{x} + \int_{\partial\mathcal{D}^{m-}} \tilde{\beta}_m \delta_{m,j}(\boldsymbol{n}) L_j v (\boldsymbol{s}_m \cdot \boldsymbol{n})\ \mathrm{d}\Gamma \right] \\
&= - \int_{\partial\mathcal{D}^{m-}} \tilde{\beta}_m \tilde{L}_m v (\boldsymbol{s}_m \cdot \boldsymbol{n})\ \mathrm{d}\Gamma + \int_{\mathcal{D}} \kappa L_b (\boldsymbol{s}_m \cdot \nabla v)\ \mathrm{d}\boldsymbol{x}
\end{aligned}
$$

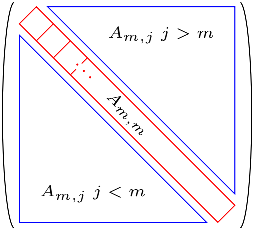
$$\forall m = 1, \cdots, N_d$$

Size Problem $(N_d \times N_{dof})^2$

$$\sum_{m=1}^{N_d}\left[a_{m,m}(L_m, v) + \sum_{j \neq m} a_{m,j}(L_j, v)\right] = \sum_{m=1}^{N_d} l_m(v)$$

$$\Downarrow \qquad\qquad \Downarrow \qquad\qquad \Downarrow$$

$$A_{m,m} \qquad\qquad A_{m,j} \qquad\qquad \boldsymbol{b}_m$$

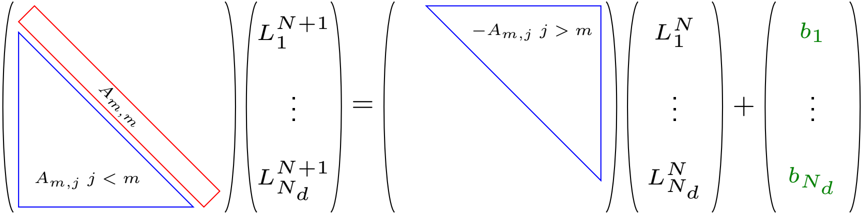
$A =$ 

Plan

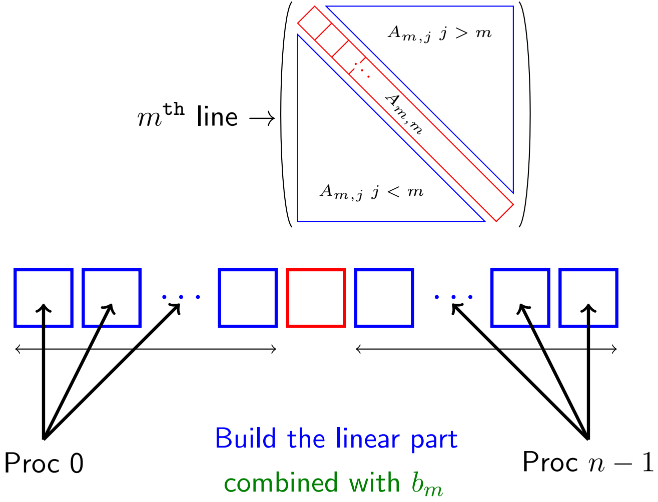
1. Forward Model and Weak Formulation
2. Iterative Method and Parallelization Strategy
3. Validation
4. Conclusion

Gauss-Siedel iterative method

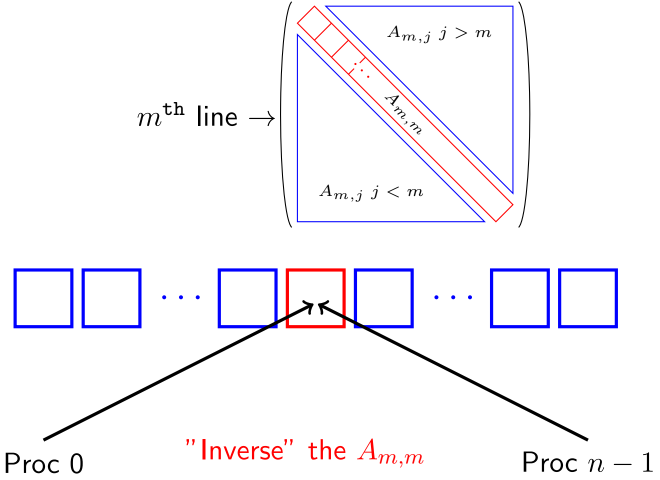
$$\sum_{m=1}^{N_d}\left[a_{m,m}(L_m^{N+1},v)+\sum_{j<m}a_{m,j}(L_j^{N+1},v)+\sum_{j>m}a_{m,j}(L_j^{N},v)\right]=\sum_{m=1}^{N_d}l_m(v)$$



Parallelization



Parallelization



"Inverse" the $A_{m,m}$ matrix with MUMPS

Building matrices $A_{m,m}$ in FreeFem++

```
varf bil1(L,v)=
int3d(Mh)((s1*dx(L)+s2*dy(L)+s3*dz(L))
*(s1*dx(v)+s2*dy(v)+s3*dz(v))
-(kap+sig*(1-Pp))*(s1*dx(L)+s2*dy(L)+s3*dz(L))*v
)
+int2d(Mh)(
(kap+sig*(1-Pp))*L*vh*(s1*N.x+s2*N.y+s3*N.z)
*(s1*N.x+s2*N.y+s3*N.z>=-10.^-10)
)
+int2d(Mh,g1)(
(kap+sig*(1-Pp))*F1(Np,N.x,N.y,N.z,Np)*L*v
)
+int2d(Mh,g2)(
(kap+sig*(1-Pp))*F2(Np,N.x,N.y,N.z,Np)*L*v
);
```

Building "matrices" $-A_{m,j}$ in FreeFem++

```
varf bil2(L,v)=
-int3d(Mh0)(
    -sig*Ps*(s1*dx(v)+s2*dy(v)+s3*dz(v))*Ln
)
-int2d(Mh0,g1)(
    +(kap+sig*(1-Pp))*F1(Np,N.x,N.y,N.z,Ns)*Ln*v
)
-int2d(Mh0,g2)(
  +(kap+sig*(1-Pp))*F2(Np,N.x,N.y,N.z,Ns)*Ln*v
)
;
```

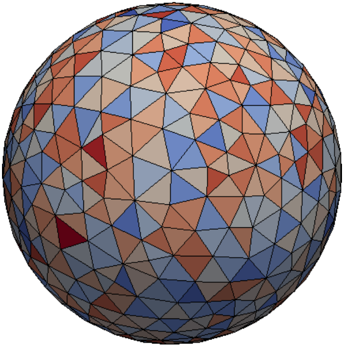
Building "matrices" $-A_{m,j}$ in FreeFem++

```
varf bil2(L,v)=
-int3d(Mh0)(
    -sig*Ps*(s1*dx(v)+s2*dy(v)+s3*dz(v))*Ln
)
-int2d(Mh0,g1)(
    +(kap+sig*(1-Pp))*F1(Np,N.x,N.y,N.z,Ns)*Ln*v
)
-int2d(Mh0,g2)(
  +(kap+sig*(1-Pp))*F2(Np,N.x,N.y,N.z,Ns)*Ln*v
)
;
```

F2 function

```
func real F2(int Np, real Nx, real Ny, real Nz, int Ns)
{
  int NumNorm;
  for(int iNorm=0;iNorm<NbNormal;iNorm++)  {
    if(abs(Nx-norm(iNorm,0))<10.^-5
   && abs(Ny-norm(iNorm,1))<10.^-5
   && abs(Nz-norm(iNorm,2))<10.^-5){
      NumNorm=iNorm;
      break;
    }
  }
  return Delta2(NumNorm*Nd+Ns,Np);
}
```

CPU time



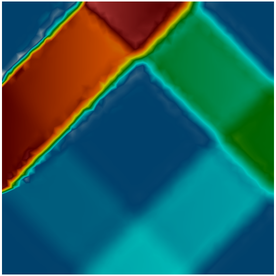
- 761 vertices
- 3088 tetrahedrons
- 980 triangles
- 4183 numElements
- $\mathbb{P}_1$ basis
- 48 directions
- 8 proc

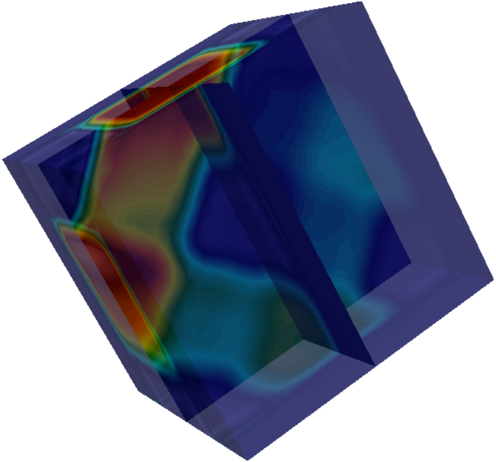
	With F1/F2	Without F1/F2
CPU/iteration	$\simeq 800s$	$\simeq 50s$

Plan

1. Forward Model and Weak Formulation
2. Iterative Method and Parallelization Strategy
3. Validation
4. Conclusion

2D and 3D code

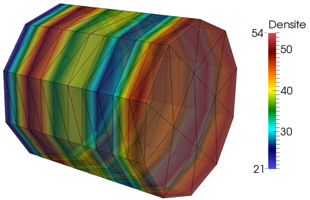


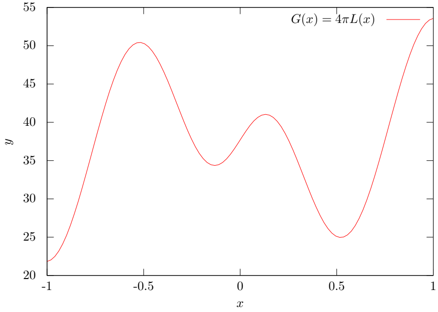


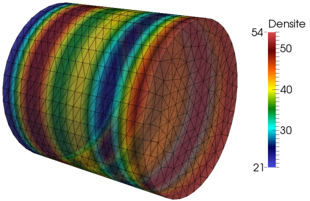
2D : Discontinuous Galerkin (DG) and SUP-G
3D : SUP-G

Analytical Validation

$$\begin{cases} L(\boldsymbol{x}) = \arctan(\pi x)\cos(2\pi x) + 3 \\ \kappa L_b = \boldsymbol{s}\cdot\nabla L(\boldsymbol{x}) + \kappa L(\boldsymbol{x}) \\ G(\boldsymbol{x}) = \int_{4\pi} L(\boldsymbol{x},\boldsymbol{s})\ \mathrm{d}\Omega(\boldsymbol{s}) = 4\pi L(\boldsymbol{x}) \\ \kappa = 0.5\mathrm{cm}^{-1}, \quad \sigma_s = 1\mathrm{cm}^{-1}, \quad g = 0.8 \end{cases}$$

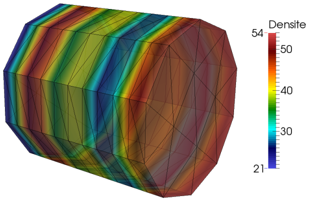


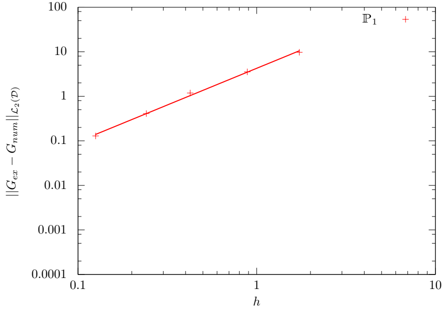


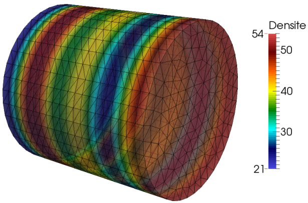


Analytical Validation

$$\begin{cases} L(\boldsymbol{x}) = \arctan(\pi x)\cos(2\pi x) + 3 \\ \kappa L_b = \boldsymbol{s}\cdot\nabla L(\boldsymbol{x}) + \kappa L(\boldsymbol{x}) \\ G(\boldsymbol{x}) = \displaystyle\int_{4\pi} L(\boldsymbol{x},\boldsymbol{s})\,\mathrm{d}\Omega(\boldsymbol{s}) = 4\pi L(\boldsymbol{x}) \\ \kappa = 0.5\mathrm{cm}^{-1}, \quad \sigma_s = 1\mathrm{cm}^{-1}, \quad g = 0.8 \end{cases}$$

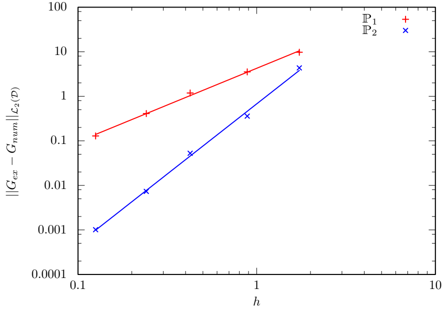


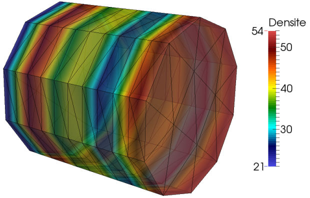


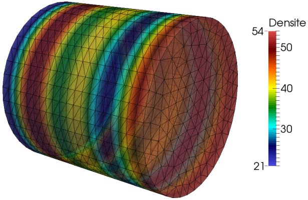


Analytical Validation

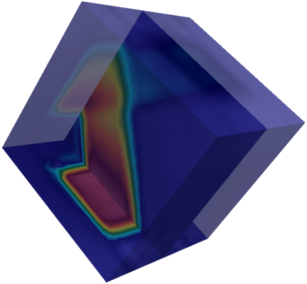
$$
\begin{cases}
L(\boldsymbol{x}) = \arctan(\pi x)\cos(2\pi x) + 3 \\
\kappa L_b = \boldsymbol{s}\cdot\nabla L(\boldsymbol{x}) + \kappa L(\boldsymbol{x}) \\
G(\boldsymbol{x}) = \displaystyle\int_{4\pi} L(\boldsymbol{x},\boldsymbol{s})\ \mathrm{d}\Omega(\boldsymbol{s}) = 4\pi L(\boldsymbol{x}) \\
\kappa = 0.5\text{cm}^{-1}, \quad \sigma_s = 1\text{cm}^{-1}, \quad g = 0.8
\end{cases}
$$







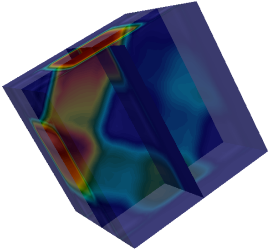
Refracted pulse function on the $\frac{\pi}{4}$ direction



- $\kappa = 0.1\text{cm}^{-1}$
- $\sigma_s = 0.5\text{cm}^{-1}$
- $g = 0$
- $n_2 = 1.4$
- 1 000 000 photons

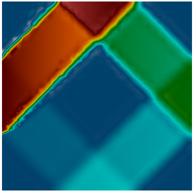
T_{nh}/R_{nh}	FEM	$E_{\text{MC}}(X)$	$\sigma_{\text{MC}}(X)$
Reflectance	1.34×10^{-1}	1.33×10^{-1}	6.66×10^{-4}
Transmittance	1.12×10^{-1}	1.08×10^{-1}	6.10×10^{-4}
Lateral transmittance 1	7.64×10^{-2}	7.55×10^{-2}	5.18×10^{-4}
Lateral transmittance 2	7.64×10^{-2}	7.50×10^{-2}	5.16×10^{-4}
Lateral transmittance 3	3.47×10^{-1}	3.26×10^{-1}	9.19×10^{-4}
Lateral transmittance 4	7.29×10^{-2}	6.97×10^{-2}	4.99×10^{-4}

Conclusion



- Solving the **RTE** with **specular reflection** on the boundary with **DG in 2D** and with **SUP-G in 2D and 3D**

- **Parallelization** stategy to use the code on a **laptop**



- **Forward** : Find a alternative to replace the F2 functions ($\mathbb{P}_0$ function, emptymesh, labels, $\cdots$)