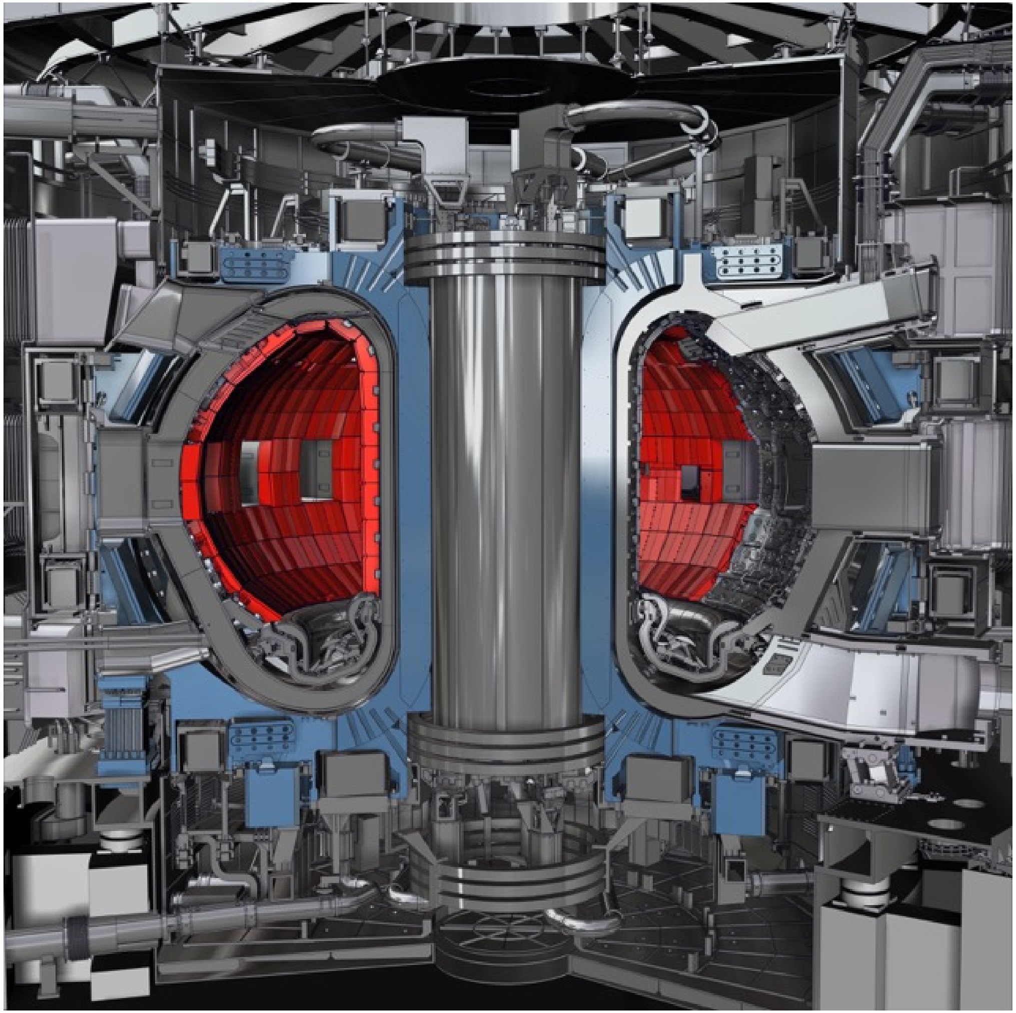
# Diffusion of deuterium through plasma facing components: explicit and implicit models

**Alex Hagen**

*Purdue University*

May 3, 2016 at POTR 376
West Lafayette, IN

# Motivation:



# Assumptions and Simplification:

## Assumptions

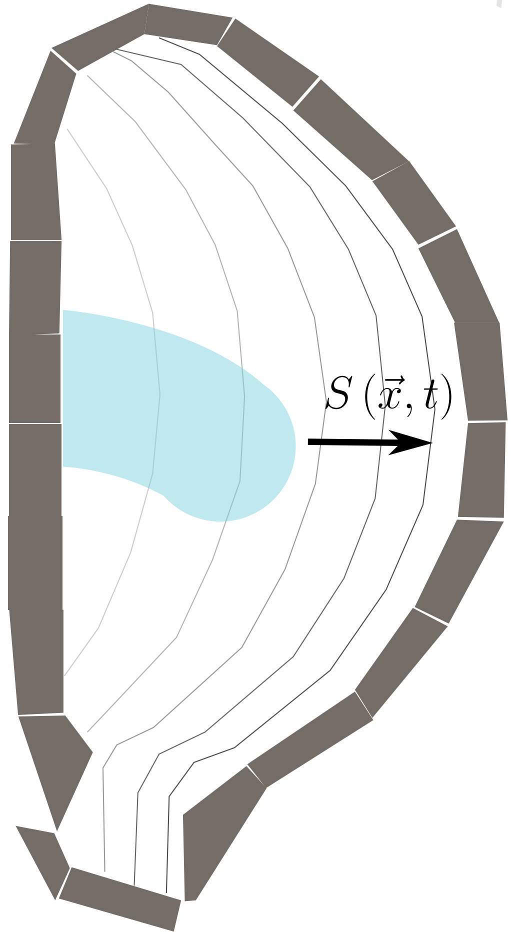



Figure 1: Diagram of current simulation setup

# Assumptions and Simplification:

## Assumptions

$\Rightarrow$ Curvature is larger than scale of diffusion/implantation (semi-infinite slab)

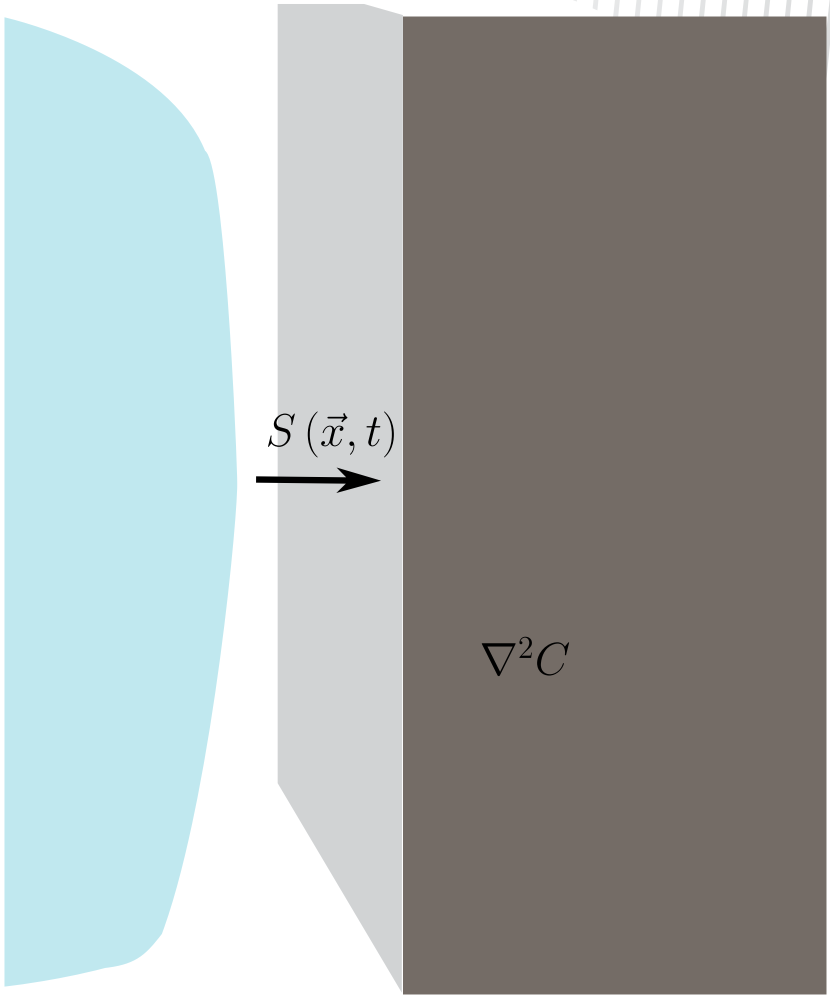



Figure 2: Diagram of current simulation setup

# Assumptions and Simplification:

## Assumptions

⇒ Curvature is larger than scale of diffusion/implantation (semi-infinite slab)

⇒ Incident flux is constant in space and time (one dimensional, constant source)

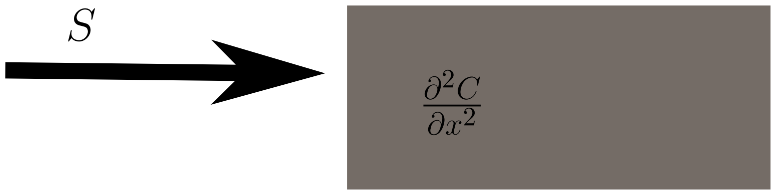



Figure 3: Diagram of current simulation setup

# Assumptions and Simplification:

## Assumptions

- $\Rightarrow$ Curvature is larger than scale of diffusion/implantation (semi-infinite slab)
- $\Rightarrow$ Incident flux is constant in space and time (one dimensional, constant source)
- $\Rightarrow$ Recombintation occurs on left boundary ($\frac{\partial C}{\partial x} = -kC^2$)

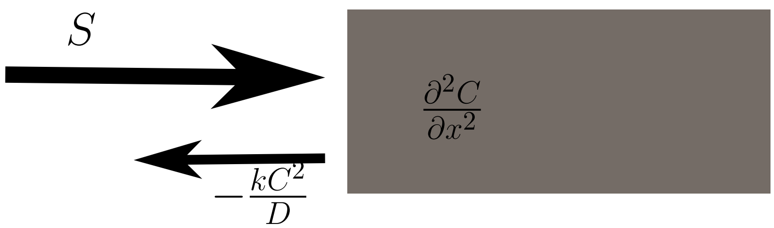



Figure 4: Diagram of current simulation setup

# Assumptions and Simplification:

## Assumptions

- $\Rightarrow$ Curvature is larger than scale of diffusion/implantation (semi-infinite slab)
- $\Rightarrow$ Incident flux is constant in space and time (one dimensional, constant source)
- $\Rightarrow$ Recombintation occurs on left boundary ($\frac{\partial C_{-1}}{\partial x} = -kC_0^2$)
- $\Rightarrow$ The left boundary is a sink ($C_{n+1} = 0$)

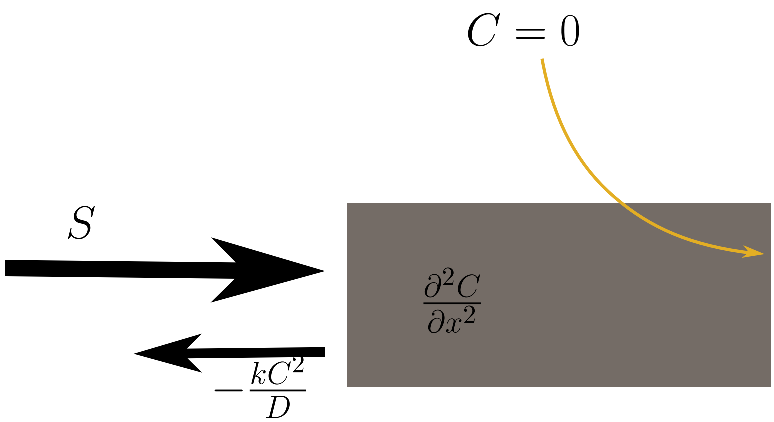
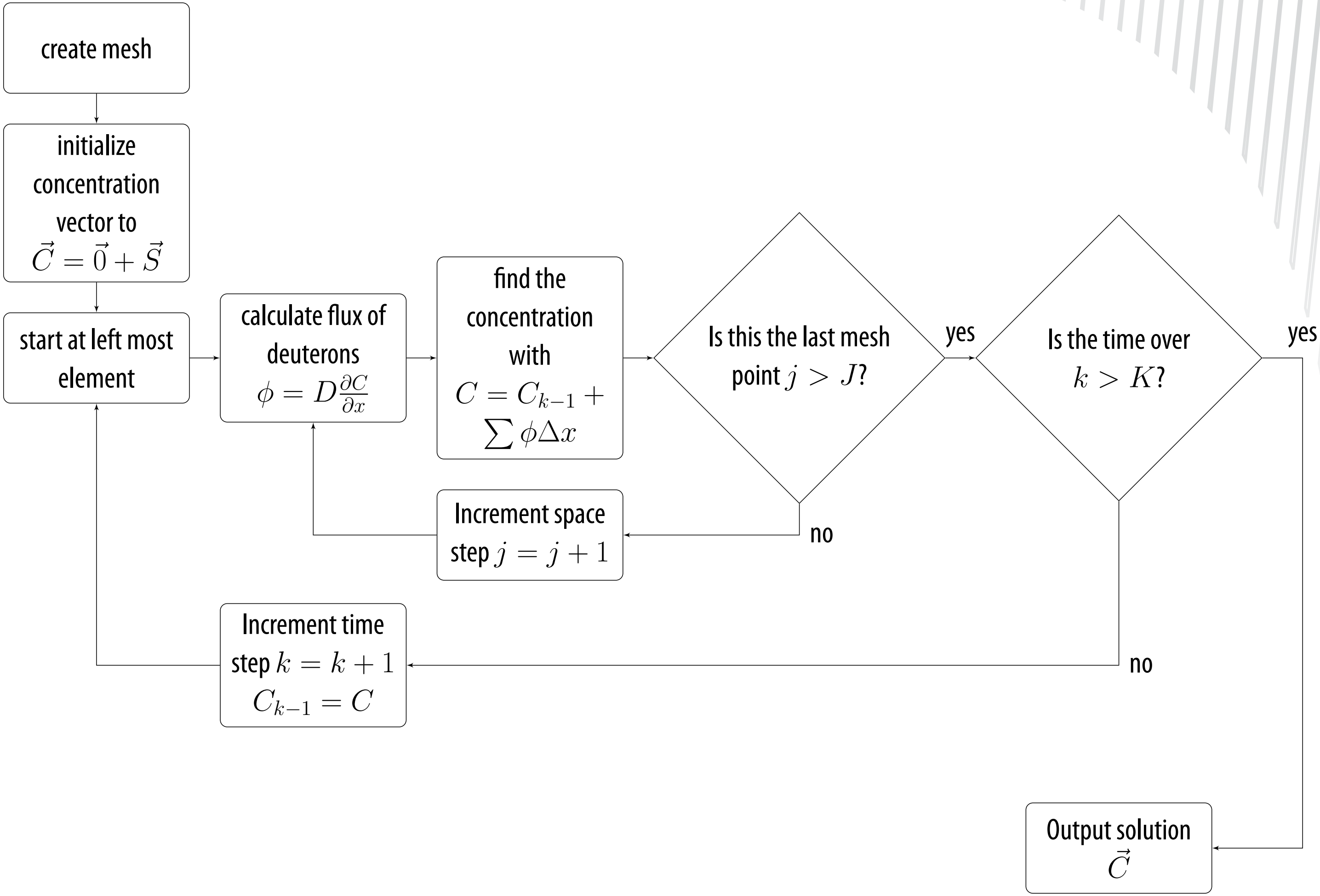



Figure 5: Diagram of current simulation setup

# Solution of Simplified Model: Finite Difference Method



# Solution of Simplified Model: Finite Element Method

We strive to solve the equation

$$
\begin{aligned}
\frac{\partial C(x,t)}{\partial t} &= D\frac{\partial^2 C(x,t)}{\partial x^2} && \text{in domain} \\
D\frac{\partial C}{\partial x} &= -kC^2 && \text{on left boundary} \\
C &= 0 && \text{on right boundary} \\
C(t+\Delta t) &= C(t) + \Delta t\, S && \text{in domain}
\end{aligned}
$$

We can generate a weighted residual for this

$$
R\left(\tilde{C}\right) = -\underbrace{\int_0^{x_r} \frac{\partial \tilde{C}}{\partial t} dx}_{\substack{\text{history terms} \\ \text{"dissipation matrix"}}} + \underbrace{\left[ wD\frac{\partial \tilde{C}}{\partial x} \right]_0^{x_r}}_{\substack{\text{boundary terms} \\ \text{"forcing vector"}}} - \underbrace{\int_0^{x_r} \frac{\partial w}{\partial x} D \frac{\partial \tilde{C}}{\partial x} dx}_{\substack{\text{domain terms} \\ \text{"stiffness matrix"}}}
$$

And we try to minimize this

# Solution of Simplified Model: Finite Element Method - Continued

$$
R\left(\vec{C}\right) = \underbrace{\begin{bmatrix} \frac{4\Delta x}{6D\Delta t} & \frac{\Delta x}{6D\Delta t} & & & \\ \frac{\Delta x}{6D\Delta t} & \frac{4\Delta x}{6D\Delta t} & \frac{\Delta x}{6D\Delta t} & & \\ & \frac{\Delta x}{6D\Delta t} & \frac{4\Delta x}{6D\Delta t} & \frac{\Delta x}{6D\Delta t} & \\ & & \frac{\Delta x}{6D\Delta t} & \frac{4\Delta x}{6D\Delta t} & \frac{\Delta x}{6D\Delta t} \\ & & & \frac{\Delta x}{6D\Delta t} & \frac{4\Delta x}{6D\Delta t} \end{bmatrix}}_{\mathbb{M}} \begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \end{bmatrix} - \underbrace{\begin{bmatrix} \frac{\Delta x}{D\Delta t} & & & & \\ & \frac{\Delta x}{D\Delta t} & & & \\ & & \frac{\Delta x}{D\Delta t} & & \\ & & & \frac{\Delta x}{D\Delta t} & \\ & & & & \frac{\Delta x}{D\Delta t} \end{bmatrix}}_{\mathbb{M}_{k-1}} \begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \end{bmatrix}_{k-1} + \underbrace{\begin{bmatrix} -\frac{kC_1^2}{D} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{\vec{l}} +
$$

$$
D \underbrace{\begin{bmatrix} \left(\frac{1}{\Delta x_0^2} + \frac{1}{\Delta x_1^2}\right) & -\frac{1}{\Delta x_1^2} & & & \\ -\frac{1}{\Delta x_1^2} & \left(\frac{1}{\Delta x_1^2} + \frac{1}{\Delta x_2^2}\right) & -\frac{1}{\Delta x_2^2} & & \\ & -\frac{1}{\Delta x_2^2} & \left(\frac{1}{\Delta x_2^2} + \frac{1}{\Delta x_3^2}\right) & -\frac{1}{\Delta x_3^2} & \\ & & -\frac{1}{\Delta x_3^2} & \left(\frac{1}{\Delta x_3^2} + \frac{1}{\Delta x_4^2}\right) & -\frac{1}{\Delta x_4^2} \\ & & & -\frac{1}{\Delta x_4^2} & \left(\frac{1}{\Delta x_4^2} + \frac{1}{\Delta x_5^2}\right) \end{bmatrix}}_{\mathbb{K}} \begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \end{bmatrix}
$$

So we have the matrix equation

$$
\begin{aligned}
0 &= -\mathbb{M}\vec{C} + \mathbb{M}_{k-1}\vec{C}_{k-1} + \vec{l} - \mathbb{K}\vec{C} \\
&= -\underbrace{\left(\mathbb{M} + \mathbb{K}\right)}_{\mathbb{A}} \underbrace{\vec{C}}_{\vec{x}} + \underbrace{\left(\mathbb{M}_{k-1}\vec{C}_{k-1} + \vec{l}\right)}_{\vec{b}}
\end{aligned}
$$

so we can simplify this to

$$\mathbb{A}\vec{x} = \vec{b}$$

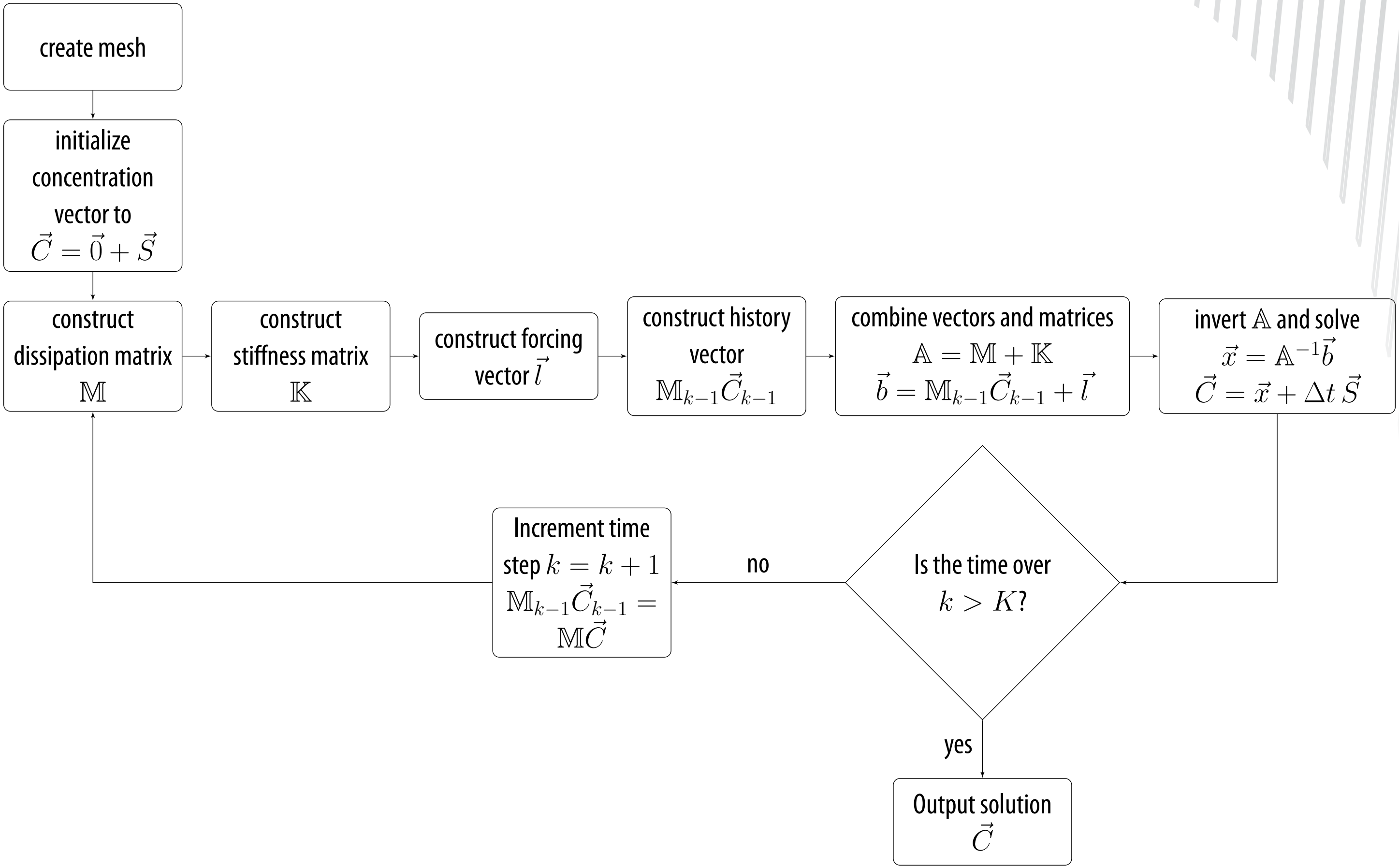
which can be solved with

$$\vec{x} = \mathbb{A}^{-1}\vec{b}$$

at each time step. Then, our approximated concentration is

$$\vec{C} = \vec{x} + \Delta t\, \vec{S}$$

# Solution of Simplified Model: Procedure Flow



# Solution Parameters:

- $\Rightarrow D = 5 \times 10^{-8}\ \frac{\text{cm}^2}{\text{s}}$
- $\Rightarrow k_r = 7 \times 10^{-22}\ \frac{\text{cm}^4}{\text{s}}$
- $\Rightarrow \Phi = 1 \times 10^{17}\ \frac{^2\text{H}}{\text{s}}$
- $\Rightarrow$ Initial source from SRIM simulation of deuterium on tungsten

$t_{final} = 1$ ms
$\Delta t = \frac{\Delta x^2}{2D}$

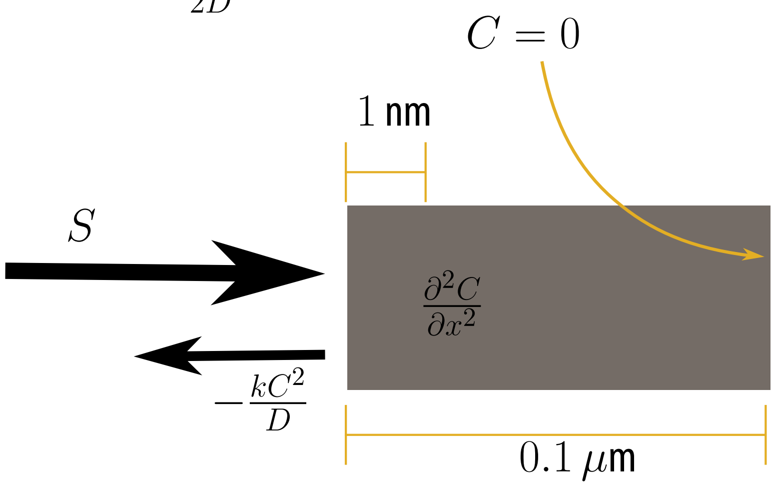



Figure 6: Diagram of current simulation setup with mesh parameters

# Results and Method Analysis: Comparison to Finite Difference Method

⇒ Solution behaves as expected

⇒ Small difference between FDM and FEM still under analysis

⇒ No current way to determine number of desorbed deuterons

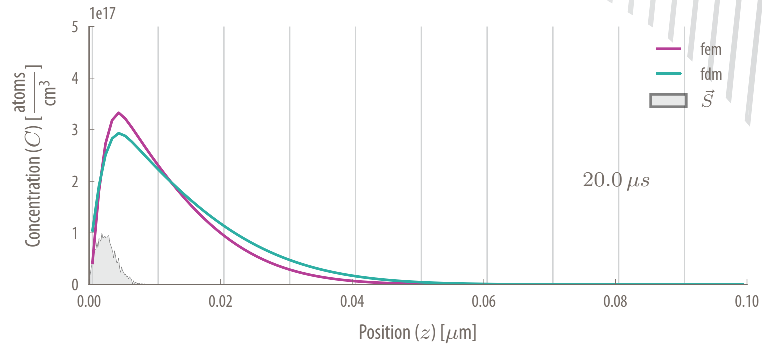



Figure 7: Solution of diffusion in Tungsten using finite element and finite difference methods, over 1000 $\mu s$

⇒ Longer final time analysis should still be performed

⇒ Grid is slightly coarser than desired

# Results and Method Analysis: Time Stability Analysis

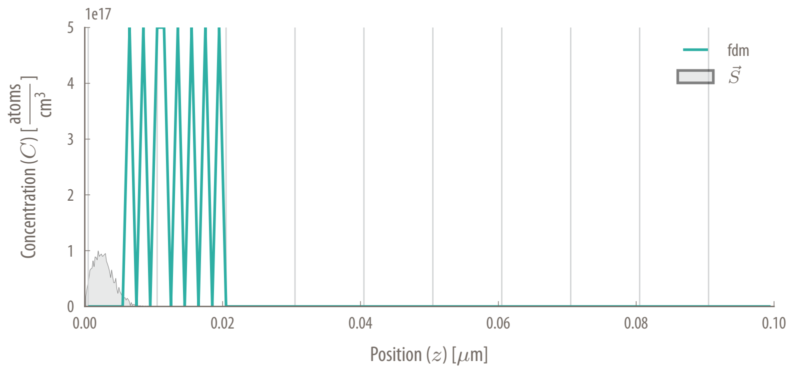



Figure 8: Solution of diffusion equation using finite difference method with $1$ $\mu$s steps

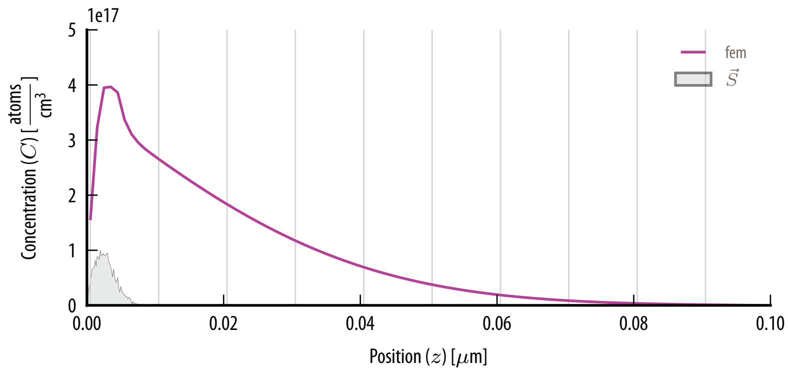



Figure 9: Solution of diffusion equation using finite element method with $1$ $\mu$s steps

$\Rightarrow$ Finite difference method completely diverges when large time steps are attempted

$\Rightarrow$ Finite element method simply provides an inaccurate answer when too large time steps are attempted

$\Rightarrow$ Source term was added "lumped" at each time step, and that may have been too large to diffuse away

# Results and Method Analysis: Computational Effort

## Finite Element Method

- $\Rightarrow$ Under "best" conditions, required $24.37$ s to solve a mesh with 1000 elements and 10000 time steps
- $\Rightarrow$ Gauss-Seidel matrix solver is iterative and could be faster
- $\Rightarrow$ Scales favorably with more elements: $\mathcal{O}\left(> 3n\right)$
- $\Rightarrow$ Scales linearly with more time steps: $\mathcal{O}\left(n\right)$

## Finite Difference Method

- $\Rightarrow$ Under "best" conditions, required $0.14$ s to solve a mesh with 100 elements and 1000 time steps
- $\Rightarrow$ Scales very poorly with more elements: $\mathcal{O}\left(n^3\right)$
- $\Rightarrow$ Scales linearly with more time steps: $\mathcal{O}\left(n\right)$

# Proposed Enhancements to Finite Element Diffusion Code:

$\Rightarrow$ Compare results to commercial FEM package, i.e. COMSOL

$\Rightarrow$ Use PARADISO to solve matrix equation

$\Rightarrow$ Determine “stability” for FEM time step

$\Rightarrow$ Provide the ability to perform on a non-uniform mesh