Abstract

Diode-Pumped Alkali Laser (DPAL) systems have proven instrumental in wide-ranging applications that require reliable high-power distribution at reasonable cost. To maintain and improve DPAL functionality, it is imperative that pumped electrons be relaxed to lower energy states through collisions with buffer atoms without generating excessive waste heat. Ideally, alkali electrons maintain a population inversion at \( n^2 \), and \( n^2P_\text{1/2} \). In order to accurately predict and model particle interactions within alkali chambers, researchers utilize powerful technology like HPC and ANSYS Fluent to conduct simulations which depend on grid generation.

Background

DPALs contain an active vaporous medium of Group 1a atoms (traditionally Rb, Cs, or K) and one or more buffer elements (He). Pump sources deliver energy to the system, exciting the alkali atoms. As the atoms return to the ground state, they release energy in the form of photons (Fig. 1) which pass through a gain cell before exiting the output mirror as a steady laser beam (Fig. 2).

Grid generation allows us to perform simulations and model the interactions between the alkali and buffer gas atoms, providing data which can be used to develop solutions to minimize heat-related energy loss and maximize power output.

Methods

- Use HPC in tangent with Pointwise and/or Python to generate multiple, unevenly-spaced structured grids
- Run grids through ANSYS Fluent’s CFD simulation software using Navier-Stokes and other relevant physics equations
- Interpolate grids to obtain simulation results for both CFD and electromagnetic activity
- Compare simulation results to determine the optimal grid dimensions and spacing for a reliable model
- Generate final grid(s) and perform additional FLUENT simulations to obtain a comprehensive DPAL model

Application

In order to capture the physics of a model along certain boundaries, a grid must be finely spaced near the border. However, fine spacing throughout the grid unnecessarily taxes computer resources. To balance computational accuracy with resource demand, the user must define a function that spaces points closely at the boundary and more coarsely toward the center of the grid. The grids below use a sinh function to produce uneven spacing along the y-axis for a defined distance from each endpoint and even spacing in the center region:

- Selected spacing function: \( y(x) = \sinh(x) = \frac{e^x - e^{-x}}{2} \)

DFT and Heisenberg’s Uncertainty Principle

Prior to bulk grid generation, fourier transform code was selected as an appropriate exercise to test HPC capabilities. For this purpose, a Python program generating discrete fourier transform (DFT) results for a selected wave function will soon serve as the benchmark of the transition to direct interaction with the HPC system while providing visual descriptions of particle behavior in position and momentum domains.

The DFT’s application to DPAL gain cells lies in the phenomenon of population inversion, a necessary distribution of electrons between the ground and excited states which allows the laser to produce a steady beam. The probability that an electron is present in a given energy level can be described by a wave function, where \( \psi \) represents the position of a particle in the wave and the area under the curve represents the probability of finding the electron in that position.

The following example demonstrates the transform of a gaussian function which describes the motion of a particle as a function of position:

\[
\Psi(x) = \frac{1}{\sqrt{2\pi \sigma^2}} e^{-\frac{(x-x_0)^2}{2\sigma^2}} e^{ik_0(x-x_0)}
\]

Conducting a discrete fourier transform of this wave function produces a function in the momentum domain which reveals the probability that the electron is traveling at a certain velocity:

\[
\Psi(k) = \int_{-\pi}^{\pi} \psi(x) e^{-ikx} \, dx
\]

According to Heisenberg’s principle, the more accurately one measures a particle’s position, the less accurately one predicts its velocity, as demonstrated in Fig. 7-8. The wider the probability curve in \( x \), the narrower it becomes in \( k \) and vice versa. Fig. 9-10 reveal a function and its transform. At a height of \( \frac{1}{\sqrt{2\pi \sigma^2}} \), the uncertainty \( \Delta x \Delta k \) exceeds the corresponding \( \Delta x \), implying a greater certainty of finding particle velocity than position.

Future Work

- Develop a script that uses Pointwise and Fluent in batch mode to generate and test a large quantity of simulation grids
- Use original DFT code to plot and analyze HPC runtime for various task distribution configurations

References


Sponsored by:

DISTRIBUTION A. Approved for public release: distribution unlimited.