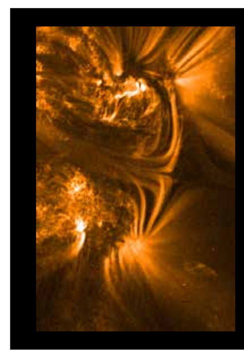
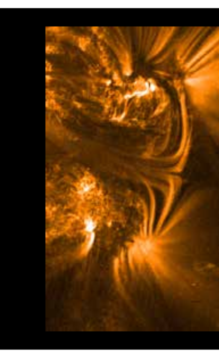


Simulation Emission Spectra at the Solar Limb



Calvin Huang[1,2], Juan Martinez-Sykora[1,3]

1. Lockheed Martin Solar and Astrophysics Laboratory 2. Gunn High School, Palo Alto 3. Bay Area Environmental Research Institute

Abstract

We performed synthetic imaging and spectra from a 3D radiative MHD simulation and compared them with observations. The simulation was created using Bifrost code with ionization equilibrium conditions. We reduced the numerical limitations which restricts us to small domains of the 3D MHD models by attaching various snapshots from the same simulation at different instances, until we get a line-of-sight (LOS) comparable to solar observations. From the snapshots we calculate, using forward modeling, the radiative transfer with optically thin approximation and ionization equilibrium conditions. We took advantage of graphics processing units (GPUs) in order to speed up our calculations. We did a parametric study of the synthetic observations taking into account a single domain, versus putting together many domains, with and without absorption, and with and without curvature. Our results reveal that the absorption and the curvature play an important role in the intensity scale height, and other observables such as Doppler shift and line width variation, with the height at the limb. In addition, we are able to tell from comparison with observations that the 3D radiative MHD models need to be improved in order to reproduce the dynamics in the chromosphere. In addition, the magnetic field configuration needs to be studied in detail in order to get closer to the Sun's magnetic field distribution.

Introduction

• IRIS (Interface Region Imaging Spectrograph) is a satellite launched in 2013 which attempts to understand the energizing of the solar atmosphere.

• The IRIS investigation combines high resolution, high throughput multi-channel UV imaging spectrograph at high cadence with advanced numerical modeling.

• The interface layer between the Sun surface (photosphere) and the million degree kelvin corona (chromosphere and transition region) is filled with dynamic fibrils and different types of spicules.

• Being such a dynamic and complicated region (with hugely varying density, temperatures, magnetic fields, etc), it is very difficult to simulate accurately

• Understanding this region is crucial to understanding coronal heating and solar wind.

• A crucial tool to interpret the observations are state-of-the-art simulations combined with synthetic imaging and profiles. The similarities, differences and discrepancies provide information about the different processes in the chromosphere.

Spectral Analysis

• Analysis of asymmetries in profiles, Doppler shift and broadening can be performed to extract meaningful features from the data

• Again, each pixel (with its own spectral line) is assigned to a GPU core

• We created tools that allow us to effectively calculate Doppler shift, line width, and characterize profile asymmetry

• Applicable to both outputs from br_ioni simulations as well as data taken from IRIS

Calculating Intensity

• Synthesized total intensity integrated along a line of sight l (assuming optically thin radiation and ionization equilibrium conditions:

$$I = \int_l A_b n_e n_h G(T, n_e) dl$$

Where A_b, n_e, n_h and $G(T, n_e)$ represent the abundance of the emitting element, the electron and the hydrogen densities, and the contribution function—interpolated from tables in the CHIANTI v7.0 database—respectively.

• Synthesized total intensity at a certain wavelength:

$$I_\nu = \frac{h\nu}{4\pi} \int_l \phi_\nu n_e n_h G(T, n_e) dl$$

$$\phi_\nu = \frac{1}{\pi^{1/2} \Delta\nu_D} e^{\left[-\left(\frac{\Delta\nu - \nu\vec{u}\cdot\vec{n}/c}{\Delta\nu_D}\right)^2\right]}$$

$$\Delta\nu_D = \frac{\nu_0}{c}\sqrt{\frac{2kT}{m_A}}$$

Where φ represents the shape of the line profile, Δν represents the deviation from the lin's rest frequency, u·⃗n is the component of the plasma velocity along the LOS, and $\Delta\nu_D$ is the width of the profile as given by Doppler broadening.

• Taking opacity into account, which is the absorption of emissions by various elements in the sun:

$$I = \int_l A_b n_e n_h G(T, n_e) e^{-\tau} dl$$

$$I_\nu = \frac{h\nu}{4\pi} \int_l \phi_\nu n_e n_h G(T, n_e) e^{-\tau} dl$$

Where τ is the integral of the attenuation coefficient α, accumulated along the line of sight.

Using the GPU

• All forms of emission modeling that we explore require some variant of ray tracing integration along the LOS.

• To simplify this, we created a CUDA and Python framework that simplifies ray tracing through a box from an arbitrary viewpoint. This framework allows full control over the CUDA kernel calculations, as well as allowing output size/ resolution, scale factor, step size, and point of view to be easily manipulated. Output size and integration step size improve accuracy and detail of the render, at the cost of proportionally increased processing time.

• Rendering of the emissions is a problem suitable for use with parallel computing, as the emissions along each ray are completely independent of rays at different positions. Therefore, we can assign a single GPU core to each ray, giving a large speedup factor over CPU processing.

• Simulation density, internal energy, and plasma velocity within a simulated 3D domain is stored as three-dimensional tables, and linearly interpolated between during calculations.

• Depending on the size of the simulation to be processed, the density, energy, and velocity arrays may take up lots of memory, preventing them from being loaded onto the GPU all at once. As such, to improve scalability and memory usage, our renderer is structured so as to split up the simulated domain along the x-axis into numerous smaller sections if necessary.

• The tool created (br_ioni) is also able to use output from bifrost simulations taking time dependent ionization into account

• Br_ioni also allows small simulated domains to be stacked on top of each other, to more accurately represent the appearance of the much larger sun.

• A graphical user interface was created to easily manipulate render settings, and to quickly visualize a simulated domain