PROJECT NARRATIVE (The narrative should not exceed 15 pages)

Visual materials, such as charts, graphs, pictures, etc., are included in the 15-page limit. References **do not** count in the 15-page limit and should be listed after the Project Narrative. URLs that provide information related to the proposal should not be included. **The 15-page limit will be strictly enforced.** The Project Narrative should address the following points:

1 SIGNIFICANCE OF RESEARCH

(Placeholder text. This section should be about **1 page**.)

This first section should be broad enough for a general audience comparing our proposal with those in other disciplines.

Broad science overview paragraph, including bold (literally and figuratively) thesis statement.

Brief statement of the problem.

Brief description of the proposed simulations (solution).

Brief description of the impact of the project on the community.

Brief description of previous awards and their relation to this proposal.

1.1 Scientific Motivation

(Placeholder text. This section should be about **2 pages**.)

This is where we really dig in to the science. The level should be appropriate for peer review in astrophysics.

Outline the science background. I like to start with an interesting observation, followed by a problem, followed by our solution to the problem. Possible structure:

- Background: recent observational studies of the CGM (COS and friends)
- Theoretical puzzle: how to explain the cool gas in galaxy halos? Observations point towards small-scale structures, well below the resolution limit of current cosmological simulations.
- Theoretical solution: small volume-filling cool structures seeded by turbulence + thermal instability in the CGM. Possibly very small - $c_{s}t_{\mathrm{cool}}$ of order a few to tens of parsecs (see Figure **??**).
- Logical numerical study: simulations of patches of the CGM with driven turbulence and astrophysical cooling, with $c_{s}t_{\mathrm{cool}}$ resolved. For driving scales of order 1 kpc, this will require very high resolution boxes.
- What will be gained from the study.
- Additional benefits to the community (e.g. better physical understanding of CGM phase structure to inform observations, subgrid model for cosmological simulations, etc..

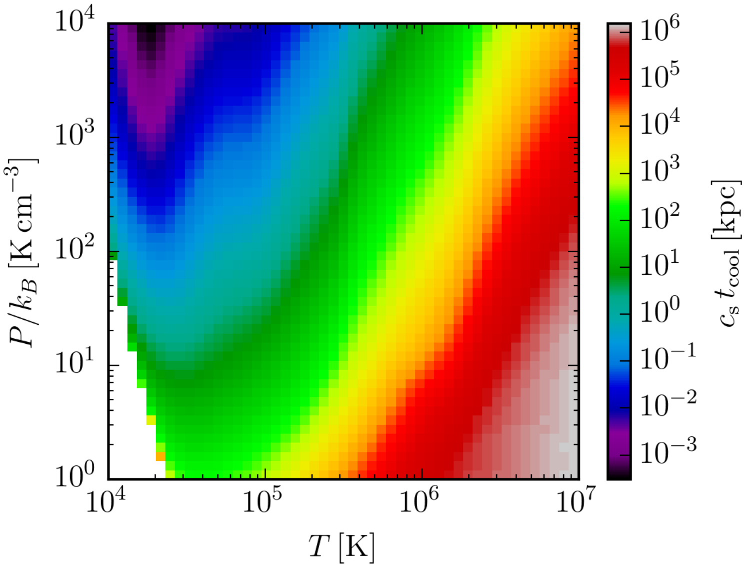

Figure 1. This figure justifies our resolution requirements. Would be nice to have the minimum length scale in one panel (as shown), and a figure justifying the box size in another panel (if we can think of an appropriate figure).

1.2 Results from Previous INCITE Awards and Relation to Current Proposal

(Placeholder text. This section should be **< 1 page**.)

(Evan will add more here.)

Previous awards include DD projects AST107 "Scaling the GPU-enabled Hydrodynamics Code Cholla to the Power of Titan" and AST119 "Extending the Physics of the GPU-Enabled CHOLLA Code to the Power of Titan" (co-I Schneider) and INCITE project AST125 "Revealing the Physics of Galactic Winds with Petascale GPU Simulations" (co-PI Schneider). Results from project AST107 included the first demonstration of `Cholla` at petascale. AST119 allowed the development of GPU-accelerated radiative cooling and associated research into the cloud-wind problem, through which we demonstrated that the cool gas in the CGM cannot be directly explained by ram pressure acceleration of dense disk material [5]. The ongoing INCITE project AST125 has produced the most detailed numerical models of multiphase galactic winds ever, but the simulations only extend out to 10 kpc, and thus cannot be used to determine what happens to the gas in the wind once it reaches the CGM. There are strong indications that starburst winds are a driver of the large-scale turbulence to be investigated via this project, see also [1]).

2 RESEARCH OBJECTIVES AND MILESTONES

(Placeholder text. This section should be about **6 pages**.)

This section needs a lot of filling out. Basically, this is where we describe the set of simulations we plan to do, and why we're doing them. I've come up with a baseline set of simulations that I think we could do based on the computational time, and a baseline set of objectives and milestones. In describing the simulations, we should include density and temperature projections (I've put some that I generated in as placeholders at the moment). I think some of Drummond's density / temperature pdfs would be good in this section as well, to help describe the analysis we will do and how it relates to our research objectives.

Table 1: Research Objectives

RO.A	Determine the effects of resolution on current numerical attempts to characterize the CGM.
RO.B	Describe the physical nature (phase structure, cloud size scales, velocity coherence length etc.) of gas in the circumgalactic medium for a range of physical pressures.
RO.C	Develop a connection between the state of the CGM and other relevant galaxy properties, both simulated and observable (e.g. halo mass, star-formation rate, etc.).

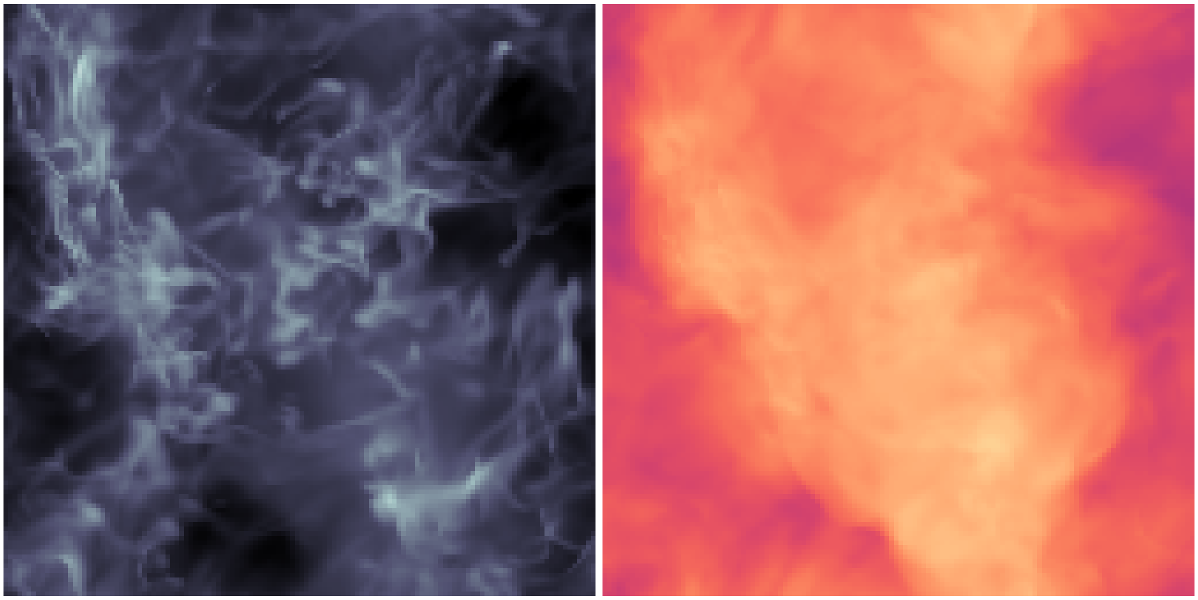

Figure 2. Placeholder figures.

2.1 Heading 2 (optional)

Insert paragraph(s).

2.1.1 Heading 3 (optional)

Insert paragraph(s).

3 COMPUTATIONAL READINESS

Placeholder text. This section, including the following subsections, is typically about **5 pages**.

3.1 Use of Resources Requested

Our program for simulating the CGM requires two classes of simulations - petascale simulations where we fully resolve the minimum length scale set by thermal instability, as well as a set of moderate-resolution simulations that will allows us to sample the parameter space of CGM conditions. We note that while we refer to them as moderate resolution, these simulations will still be orders of magnitude better resolution than any previous simulations of the CGM, and therefore will provide highly valuable scientific content. Since all of our calculations will use the GPU-native code `Cholla`, each of these simulations will require large numbers of GPUs (4096–16,384) as provided by the Titan XK7 system.

(Evan will add more here.)

Table 2: Research Milestones

Milestone		**Objective**
Quarter 1		
RM.A	Develop "fiducial" turbulent CGM model and run resolution study, including first petascale simulation of the CGM.	RO.A , RO.B
Quarter 2		
RM.B	Run suite of moderate resolution turbulent box simulations to explore relevant parameter space.	RO.B, RO.C
Quarter 3		
RM.C	Run additional extreme resolution models (as informed by parameter study) to produce a state-of-the-art suite of numerical simulations to be used for comparison to observational and theoretical models of the circumgalactic medium.	RO.A, RO.B, RO.C

Table 3: Research Simulations

Simulation Type and Details		**Objective / Milestone**	**Resolution**	**Titan Nodes**	**Titan Node Hours**
Quarter 1: 0.5M node hours					
RS.A	Fiducial CGM Petascale Simulation and Resolution Study	RO.A, RO.B	$N = 1024^3$	8 - 16,384	0.5M
Quarter 2: 0.7M node hours					
RS.B - M	Parameter Study, set of 12 moderate resolution simulations	RO.B, RO.C	$N = 2048^3$	4096	0.7M
Quarter 3: 1.5M node hours					
RS.N - P	Three additional petascale simulations based on results of parameter study	RO.A, RO.B, RO.C	$N = 4096^3$	16,384	1.5M

3.2 Computational Approach

Overview:
Cholla is a Godunov[2]-based, finite-volume, Eulerian grid hydrodynamics code that takes advantage of the massively parallel computing power of GPUs [3]. In order to harness this power, Cholla was designed with the operation of the GPU in mind. Cholla consists of a set of C/C++ routines that run on the CPU (the “host”) plus functions called kernels that execute on one or more GPUs (the “device”). The device kernels and the host functions that call them are written in CUDA C, an extension to the C language introduced by NVIDIA. All of the CUDA functions are contained in a separate hydro module so that they can be compiled independently with the NVIDIA nvcc compiler.

Cholla represents the state-of-the-art for astrophysical hydrodynamics simulations on a fixed Cartesian mesh. The physical modeling used by Cholla includes a range of reconstruction methods (including both the characteristics PPM model of Athena and the primitive PPM model used by, e.g., Flash), a variety of exact and approximate Riemann solvers, and two unsplit integrators (Constrained Transport Upwind and Van Leer). As demonstrated in [3], the use of GPUs enables Cholla to achieves a $\sim$ 50× speed-up (1 GPU vs. 1 CPU core) over CPU-only codes performing hydrodynamics with a similar level of physical fidelity.

Given the typical power of a single GPU, small problems can easily be run on a single host/device pair. For large problems like those considered by this proposal, Cholla can be run using the MPI library to perform message passing between processes that govern separate computational subvolumes. Each subvolume is treated as a self-contained simulation volume for the duration of each simulation time step. Portions of our algorithm that require information from potentially distant cells in the global simulation volume are carried out on the host. The main host functions set initial conditions, apply boundary conditions, and perform any interprocess communications. Parts of the calculation that only require information from nearby cells are carried out on the device. Because the bulk of the computational work resides in the hydrodynamics integration module that requires a stencil containing only local cells, essentially all of the hydrodynamical computations are performed on the GPU. The steps in the Cholla algorithm are listed below.

1. Initialize the simulation by setting the values of the conserved fluid quantities for all cells in the simulation volume, and calculate the first time step.
2. Transfer the array containing the conserved variables and other fluid variables of interest (e.g. the gas thermal energy) to the GPU. This array contains all the fluid variables that are being tracked for every cell in the simulation volume.
3. Perform the hydrodynamic integration (using either the CTU or Van Leer method) on the GPU, including updating the conserved variable array and computing the next time step.
4. Transfer the updated fluid variable array back to the CPU.
5. Apply the boundary conditions. When running an MPI simulation, this step may require interprocess communication to exchange information for cells at the edges of subvolumes.
6. Output simulation data if desired.

The initialization of the simulation is carried out on the host(s). The initialization includes setting the values of the fluid variables for both the real and the ghost cells according to the conditions specified in a text input file. Ghost cells are a buffer of cells added to the boundaries of a simulation volume to calculate fluxes for real cells near the edges. The number of ghost cells reflects the size of the local stencil used to perform fluid reconstruction. Because updating the ghost cells at each time step may require information from cells that are not local in memory, the values of the ghost cells are set on the host before transferring data to the GPU.

Once the simulation volume has been initialized on the CPU, the hydrodynamical calculation begins. The host copies the fluid variable array onto the device. Because the GPU has less memory than the CPU, the fluid variable array associated with a single CPU may be too large to fit into the GPU memory at once. If so, Cholla uses a series of subgrid splitting routines to copy smaller pieces of the simulation onto the GPU and carries out the hydrodynamics calculations on each subvolume. At the end of the hydro calculation the next time step is calculated on the device using a GPU-accelerated parallel reduction. The updated fluid variables and new time step are then transferred back to the host. The host updates the values of the ghost cells using the newly calculated values of the real cells, and Steps 2-5 repeat until the desired final simulation time is reached.

The design of the massively parallel algorithm implemented by Cholla allows execution on multiple GPUs simultaneously. Cholla can thereby gain a multiplex advantage beyond the significant computation power afforded by a single GPU. This additional parallelization is implemented using the MPI library. The global simulation volume is decomposed into subvolumes, and the subvolumes are each assigned a single MPI process. In Cholla, each MPI process runs on a single CPU that has a single associated GPU, such that the number of MPI processes, CPUs, and GPUs are always equal. When the simulation volume is initialized, each process is assigned its simulation subvolume and surrounding ghost cells. Since the hydrodynamical calculation for every cell is localized to a finite stencil, only the ghost cells on the boundary of the volume may require updating from other processes via MPI communication every time step. Compared with a simulation done on a single CPU/ GPU pair, additional overheads for a multi-process simulation can therefore include MPI communications needed to exchange information at boundaries and potential inefficiencies in the GPU computation introduced by the domain decomposition. While domain decomposition influences communications overheads in all MPI-parallelized codes by changing the surface area-to-volume ratio of computational subvolumes, domain decomposition additionally affects the performance of a GPU-accelerated code by changing the ratio of ghost to real cells in memory that must be transferred to the GPU. Since memory transfers from the CPU to the GPU involve considerable overhead, domain decompositions that limit the fraction of ghost cells on a local process are favorable.

3.3 Parallel Performance

The ability to run extremely high resolution static grid hydrodynamic simulations was the primary motivation for creation of the Cholla code, and as such, weak scaling performance has been a high priority at all stages of development. Our DD Project AST107, "Scaling the GPU-enabled Hydrodynamics Code Cholla to the Power of Titan", allowed us to test the parallel performance of the code in the petascale regime, with the excellent results shown in the left panel of Figure 3. These tests followed the adiabatic propagation of an acoustic wave across the grid for a constant number of time steps. The total sizes of the simulations were scaled such that each GPU was assigned $\approx 322^3$ cells. Figure 3 displays the scaling of the total simulation runtime, as well as the breakdown between the hydrodynamics integration, all of which is computed on the GPU, and the necessary communication between MPI processes to exchange boundary cell information. The jaggedness of each scaling reflects the fact that some domain decompositions have more favorable surface-to-volume ratios than others, however we emphasize the the total scaling remains roughly flat out to 16,384 GPUs (simulating > 0.5T cells), on roughly 90% of the Titan system.

We have implemented a novel GPU-accelerated radiative cooling scheme into Cholla that uses the texture memory on the GPU to perform linear interpolation on pre-computed cooling tables as a function of the gas density and temperature. The right panel of Figure 3 compares the timing of adiabatic (solid) and radiatively-cooling (dotted) sound wave propagation tests using 256^3 cells per GPU on the Titan XK7 system, and demonstrates that the weak scaling performance of the Cholla code is nearly identical with or without radiative cooling. These tests are very similar to the calculations that will be performed in our turbulent CGM simulations, giving us additional confidence in our estimated run time for the petascale simulations that require optically-thin radiative cooling.

3.4 Developmental Work

Most of the developmental work for this program has either already been completed, or will be complete by the start of 2019. However, we highlight below some of the additional physics modules in Cholla that will be used in this work, particularly those that we may improve over the coming months.

3.4.1 Cooling: COMPLETED

Since the publication of [3], a primary developmental task for the Cholla codebase was the now-completed implementation of a model for optically-thin cooling. This task formed much of the motivation for our DD Project AST119 granted in January 2016. This developmental work is complete and is described in [5]. Algorithmically, energy losses due to radiative cooling are precomputed using the Cloudy code [4] assuming solar metallicity and a photoionizing cosmic UV background. The cooling rates are stored in a two-dimensional table, and interpolated as a function of gas density and temperature. This

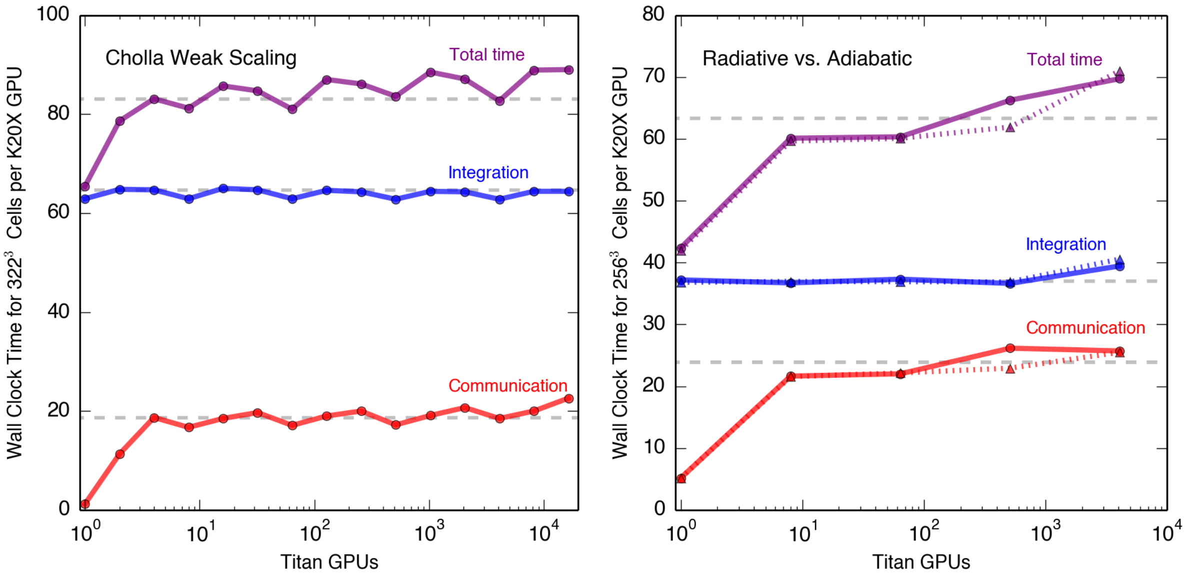

Figure 3. Weak scaling tests with `Cholla` on Titan. Shown is the wall clock execution time for an acoustic wave propagation test simulation when the number of cells per GPU is kept fixed (322^3, left panel; 256^3, right panel). The code exhibits excellent weak scaling over four orders of magnitude in GPU (node) number, up to 16,384 Titan nodes tested. The largest simulation run in this test contained $>$ 0.5T cells. The total simulation (purple), hydro algorithm (blue), and communications plus boundary conditions (red) timings are shown separately for comparison. In each case, the ideal scaling would be constant (gray dashed lines). Our novel GPU hardware-accelerated implementation of radiative cooling (right panel) allows for radiative simulations (dotted lines, triangles) to match otherwise identical adiabatic simulations (solid lines; circles) in computational efficiency and weak scaling, as tested to 4096 GPUs.

interpolation is GPU-accelerated by storing the table in texture memory on the GPU, which allows for interpolation to be performed natively by the hardware using the CUDA `cudaTextureObject_t` and `tex2D` capabilities. We also use timestep subcycling combined with a forward-Euler integration to account for short cooling timescales in high density gas and radiative shocks. Even when several subcycles are required, this method of calculating radiative cooling remarkably adds negligible computational cost, allowing the superior weak scaling of `Cholla` to be maintained (see Figure 3).

3.4.2 Turbulence Generator: In Progress

As demonstrated in Figure2, a turbulence generator has already been built for `Cholla`. The current model imposes a series of “kicks" to the velocities of each cell, with the energy injection rate normalized to offset energy losses due to radiative cooling. The velocity perturbations are applied over the course of a single time step at discreet intervals, typically 1/10th of the turbulent crossing time, $\Delta t = 0.1L/c_s$. In addition to the kick model, this work will make use of a "constant energy injection" mode of turbulence driving that is currently under construction. In the constant energy model, kinetic energy is re-added to the grid at the end of each time step in an amount that exactly balances the global radiative losses. Co-PI Fielding has experience implementing this sort of turbulence generator, and will implementing it in `Cholla`.

3.4.3 Conduction: In Progress, Supplemental

While not strictly required for the simulations outlined in this proposal, conduction is an additional physical process that may have a large effect on the physical state of gas in the circumgalactic medium. As such, we have begun development work on a conduction module for `Cholla`, which will allow us to further explore the nature of thermal instability. This development is being done in conjunction with our current INCITE project (AST125), and as such, will be complete by the end of 2018. We expect to use conduction in a subset of the parameter study simulations for comparison purposes.

4 REFERENCES

[1] Fielding, D., et al. “The Impact of Star Formation Feedback on the Circumgalactic Medium." *MNRAS*, **466**, 3810 (2017).

[2] Godunov, S., “A Difference Scheme for Numerical Solution of Discontinuous Solution of Hydrodynamic Equations.” *Math. Sbornik*, **47**, 271 (1959)

[3] Schneider, E. and Robertson, B., “`Cholla`: A New Massively Parallel Hydrodynamics Code for Astrophysical Simulation”, *ApJS*, **217**, 24 (2015).

[4] Ferland, G., et al. “The 2013 Release of Cloudy.” *RMAA*, **49**, 137 (2013)

[5] Schneider, E. and Robertson, B., “Hydrodynamical Coupling of Mass and Momentum in Multiphase Galactic Winds”, *ApJ*, **834**, 144 (2017).

[6] Steidel, C., et al. “The Structure and Kinematics of the Circumgalactic Medium from Far-ultraviolet Spectra of z =2-3 Galaxies.” *ApJ*, **717**, 289 (2010)