PROCEEDINGS OF SCIENCE

Introducing Octo-tiger/HPX: Simulating Interacting Binaries with Adaptive Mesh Refinement and the Fast Multipole Method

Dominic C. Marcello*
Department of Physics and Astronomy, Louisiana State University, Baton Rouge, LA 70803, USA
E-mail: dmarce504@gmail.com

Kundan Kadam
Department of Physics and Astronomy, Louisiana State University, Baton Rouge, LA 70803, USA

Geoffrey C. Clayton
Department of Physics and Astronomy, Louisiana State University, Baton Rouge, LA 70803, USA

Juhan Frank
Department of Physics and Astronomy, Louisiana State University, Baton Rouge, LA 70803, USA

Hartmut Kaisar
The Center for Computation & Technology, Louisiana State University, Baton Rouge, LA 70803, USA

Patrick M. Motl
The School of Sciences, Indiana University Kokomo, Kokomo, IN 46904, USA

We introduce a new computer code, Octo-tiger/HPX. Octo-tiger/HPX simulates classically self-gravitating hydrodynamic fluids on an adaptive mesh refinement (AMR) grid. It is particularly suited for modeling interacting binary star systems. To that end, Octo-tiger/HPX uses special techniques to ensure the conservation of linear momentum, angular momentum, and energy to machine precision. Octo-tiger/HPX makes use of the new distributed C++ runtime system, High Performance Para‖eX, to enable execution on massively parallel computers.

Accretion Processes in Cosmic Sources - APCS2016 -
5-10 September 2016,
Saint Petersburg, Russia

*Speaker.

PoS(APCS2016)055

1. Introduction

Here we present Octo-tiger/HPX, a computer code for simulating self-gravitating astrophysical fluids. Octo-tiger/HPX is designed to address issues particular to the self-consistent modeling of interacting binary star systems. It combines grid based adaptive mesh refinement (AMR) methods for simulating hydrodynamical fluids with the fast multipole method (FMM) for computation of the gravitational field.

Interacting binary systems evolve in a state of near equilibrium, driven closer by systemic angular momentum losses until they reach a dynamical phase in which their final fate between tidal disruption and merger or survival as a secularly expanding binary is decided by the interplay of mass and angular momentum transfers, thermal and dynamical adjustments in the sizes and shapes of the binary components, and tidal interactions [1]. Clearly an accurate accounting of the angular momentum through this phase is important in deciding which binaries will merge and which will survive. Although typical numerical methods for simulating hydrodynamical systems conserve linear momentum to machine precision, angular momentum is only conserved for special geometries or at the expense of linear momentum conservation [2]. Established FMM methods [e.g. 3] can be implemented in a manner that conserves linear momentum, but until now, not in a manner that conserves angular momentum. The evolution of interacting binary systems is heavily influenced by the balance of angular momentum, thus violating the conservation of angular momentum. This can degrade the accuracy of numerical simulations of such systems. Actual rates of angular momentum violation depend on the particular numerical methods used and what is being modeled. [4] found a gain rate of $\sim 10^{-4}$/orbit for polytropic binaries of mass ratio 1.0 and 0.8436. [5] found a loss rate of $\sim 10^{-6}$/orbit for an interacting polytropic binary of mass ratio 0.7. [6] found a $\sim 10^{-3}$/orbit violation rate using SPH to simulate 84 orbits of an interacting 0.8 $M_\odot$ accretor and 0.2 $M_\odot$ donor.

To prevent angular momentum violations, Octo-tiger/HPX uses the recently developed technique of [7] to enable machine precision conservation of angular momentum in the hydrodynamical solver. This is combined with a modification to the FMM we have developed that allows the simultaneous conservation of linear and angular momenta through the gravitational field. First, we briefly describe the High Performance Para‖eX (HPX) runtime system (§2). Then we present the model equations that Octo-tiger/HPX solves (§3), followed by the numerical methods employed by Octo-tiger/HPX (§4). Finally, we direct the reader to movies of simulations performed by Octo-tiger/HPX (§5) available on the World Wide Web (WWW).

2. Parallelization With HPX

Octo-tiger/HPX is implemented for execution on distributed computing clusters using HPX [8]. HPX is a distributed C++ runtime system under development by the Ste‖ar Group, headquartered at the Center for Computation & Technology at Louisiana State University. HPX is an implementation of the Para‖eX execution model [9]. As opposed to the communicating sequential processes (CSP) model of distributed parallel execution, Para‖eX emphasizes fine grained parallelism. To this end, HPX is designed to simultaneously handle millions of light weight threads of execution in user space. By breaking the workload into small pieces, HPX is able to hide net-

work and other latencies. The ultimate goal of HPX is to accommodate exascale computing on the distributed computing platforms of the near and distant future.

HPX relies on C++ “futures" to accomplish synchronization across both local and remote computational resources. The C++ class, “std::future", allows the user to specify an asynchronous unit of work to be performed and encapsulates the result of that work in an object (the “future") that can be retrieved when needed later. HPX takes this idea further by allowing the specified work to occur remotely (see Figure 1). With Octo-tiger/HPX, each computational node on a distributed

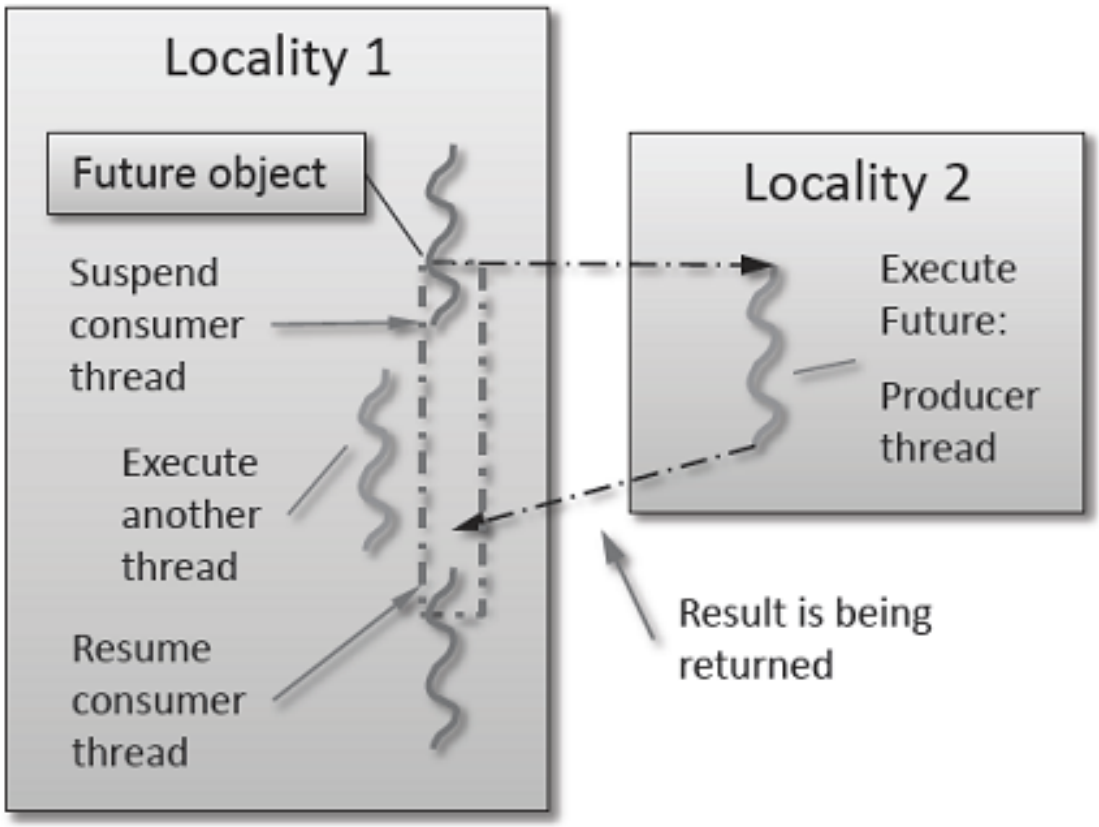


Figure 1: A depiction of how hpx::future operates. Time runs from top to bottom, and threads of execution are represented by squiggly lines. In the upper left panel, a future object is created. Creation of this object takes, as inputs, a function and the arguments to a function. Responsibility for executing this function and producing the result are then (potentially) delegated to another computational resource, potentially on a remote locality. The thread that created the future object can then proceed to do other work. When it requires the result of the future object to continue, it will suspend if that result is not yet available, freeing the processor it was running on for execution of other threads. When the result is available, the original thread resumes execution.

system may have dozens to hundreds of sub-grids in its memory. Octo-tiger/HPX specifies to HPX the work to be performed and the data dependencies between units of work, while HPX handles the actual scheduling of that work and the transfer of data across the network to and from remote computational nodes. When a data dependency prevents a particular thread of execution

from progressing, HPX seamlessly suspends that thread and loads another thread that is ready to perform useful work.

3. Model Equations

Octo-tiger/HPX uses the differential equations that describe inviscid fluid flow under the influence of its own gravitational field to evolve, in time, the mass, momentum, and energy densities of an astrophysical system. The differential equations that describe the fluid flow are

$$\frac{\partial}{\partial t}\rho + \nabla \cdot \rho \mathbf{v} = 0, \tag{3.1}$$

$$\frac{\partial}{\partial t}\mathbf{s} + \nabla \cdot \mathbf{v}\mathbf{s} + \nabla p = \rho \mathbf{g} + \Omega \times \mathbf{s}, \tag{3.2}$$

and

$$\frac{\partial}{\partial t}\left(E + \tfrac{1}{2}\rho\phi\right) + \nabla \cdot (\mathbf{v}E + \mathbf{u}p + \mathbf{u}\rho\phi) = \tfrac{1}{2}\left(\rho\frac{\partial}{\partial t}\phi - \phi\frac{\partial}{\partial t}\rho\right). \tag{3.3}$$

where ρ is the mass density, $\mathbf{v}$ is the rotating frame fluid velocity, $\mathbf{s}$ is the inertial frame linear momentum density, p is the gas pressure, $\mathbf{g} := -\nabla\phi$ is the gravitational acceleration, ϕ is the gravitational potential, $\Omega := \Omega\hat{k}$ is the angular frequency, and E is the inertial frame gas energy density (including kinetic, internal, and possibly degenerate gas energies), The inertial frame fluid velocity, $\mathbf{u}$, is defined as $\mathbf{u} := \frac{\mathbf{s}}{\rho}$. It is related to $\mathbf{v}$ by the equation $\mathbf{u} := \mathbf{v} + \mathbf{x} \times \Omega$, where $\mathbf{x}$ is the distance from the origin. The pressure, p, is a function of specific internal energy, ε, and density, $p := p(\rho, \varepsilon)$. Octo-tiger/HPX allows a choice between an ideal equation of state and the equation of state given by [10]. For the ideal equation of state, $p_{\text{ideal}} := (\gamma - 1)\rho\varepsilon$, where γ is the ratio of specific heats.

The specific internal energy is computed according to the dual energy formalism of [11]. The dual energy formalism corrects for the numerical difficulties encountered when internal gas energy density is much smaller than total gas energy. In this regime, computing the internal gas energy using $\frac{1}{\rho}E - \frac{1}{2}v^2$ can lose numerical accuracy. The internal energy could even become negative, causing negative pressure and numerical instability as a result. The dual energy formalism requires the evolution of an additional variable, the "entropy tracer" (as used in [4]), τ, where $\tau := (\rho\varepsilon)^{\frac{1}{\gamma}}$. The entropy trace is evolved according to

$$\frac{\partial}{\partial t}\tau + \nabla \cdot \mathbf{v}\tau = 0. \tag{3.4}$$

The specific internal energy is then computed using

$$\varepsilon := \begin{cases} \frac{1}{\rho}E - \frac{1}{2}v^2 & E - \frac{1}{2}\rho v^2 \geq 0.01E \\ \frac{1}{\rho}\tau^{\gamma} & \text{else} \end{cases} \tag{3.5}$$

The gravitational field is computed by integration,

$$\phi[\mathbf{x}] := -\int_V \frac{G\rho'}{|\mathbf{x} - \mathbf{x}'|} dV'. \tag{3.6}$$

In order to conserve total energy, the discretized version of equation (3.3) also requires a solution to the explicit time derivative of ϕ,

$$\frac{\partial}{\partial t}\phi\left[\mathbf{x}\right] := -\int_V \frac{G\frac{\partial}{\partial t}\rho'}{\left|\mathbf{x}-\mathbf{x}'\right|}dV'. \tag{3.7}$$

4. Numerical Method

Octo-tiger/HPX combines grid based hydrodynamics methods on an octree AMR grid with a specialized version of the FMM. First we describe the AMR structure and communication patterns between AMR sub-grids. Then we describe the FMM and the modification we have developed to conserve angular momentum to machine precision. Next we describe the hydrodynamics method, which adopts the techniques described by [7] for the conservation of angular momentum to machine precision. Finally, we describe Octo-tiger/HPX's method for producing initial models of interacting binary stars.

4.1 Adaptive Mesh Refinement

Octo-tiger/HPX discretizes the computational domain on an octree AMR structure. Each node in the structure is a $N \times N \times N$ Cartesian sub-grid, and each node may be further refined into eight child nodes, each containing their own $N \times N \times N$ sub-grid with twice the resolution of the parent sub-grid. N must be an even number and is constant for a given simulation. Typical values used are 8, 10, or 12. Each sub-grid is either entirely refined into eight child nodes, or not refined at all. Octo-tiger/HPX does not refine in time, each refinement level is advanced in sync with all the others. The AMR is structure is "properly nested", meaning there is no more than one jump in refinement level across adjacent leaf nodes. A partial octree structure is depicted in Figure 2.

In the present implementation, the refinement criteria are based solely on ρ and its fractions. Octo-tiger/HPX dynamically refines and derefines, checking the refinement criteria every 15 time steps to see if refinement or derefinement is required. This is the minimal number of time-steps required for a feature of the flow to propagate two cells, so the boundary width used to check for grid refinement is two cells wide. Newly created sub-grids are initialized from their parents using the minmod limited slopes of the primitive variables, ρ, $\mathbf{v}$, ε, τ, and $\frac{\mathbf{z}}{\rho}$. These slopes are transformed into slopes for the state vector and the resulting first order equation is integrated for each child cell. The equation is

$$w^f := w^C + \frac{1}{4}\left(\pm m_x \pm m_y \pm m_z\right), \tag{4.1}$$

where w^f is the resulting value for the refined cell, w^C is the value of its parent cell, and m_x, m_y, and m_z are the limited slopes from the parent cell. Whether $+$ or $-$ is chosen for the $\pm$ signs depends on which child cell is being computed. Note that the potential energy, $\frac{1}{2}\rho\phi$, is added to E before the interpolation and subtracted from it after, to enforce conservation of total energy. Special treatment is also given to to the spin momenta, $\mathbf{z}$ (defined in §4.3), to ensure it is properly accounted for in the linear momenta, $\mathbf{s}$, in the interpolated child cells.

The piecewise parabolic method (PPM) reconstruction used by the hydrodynamics solver requires three ghost cells, therefore the evolved state vector, $\mathscr{U}$, is stored in an array with $(N+6)^3$

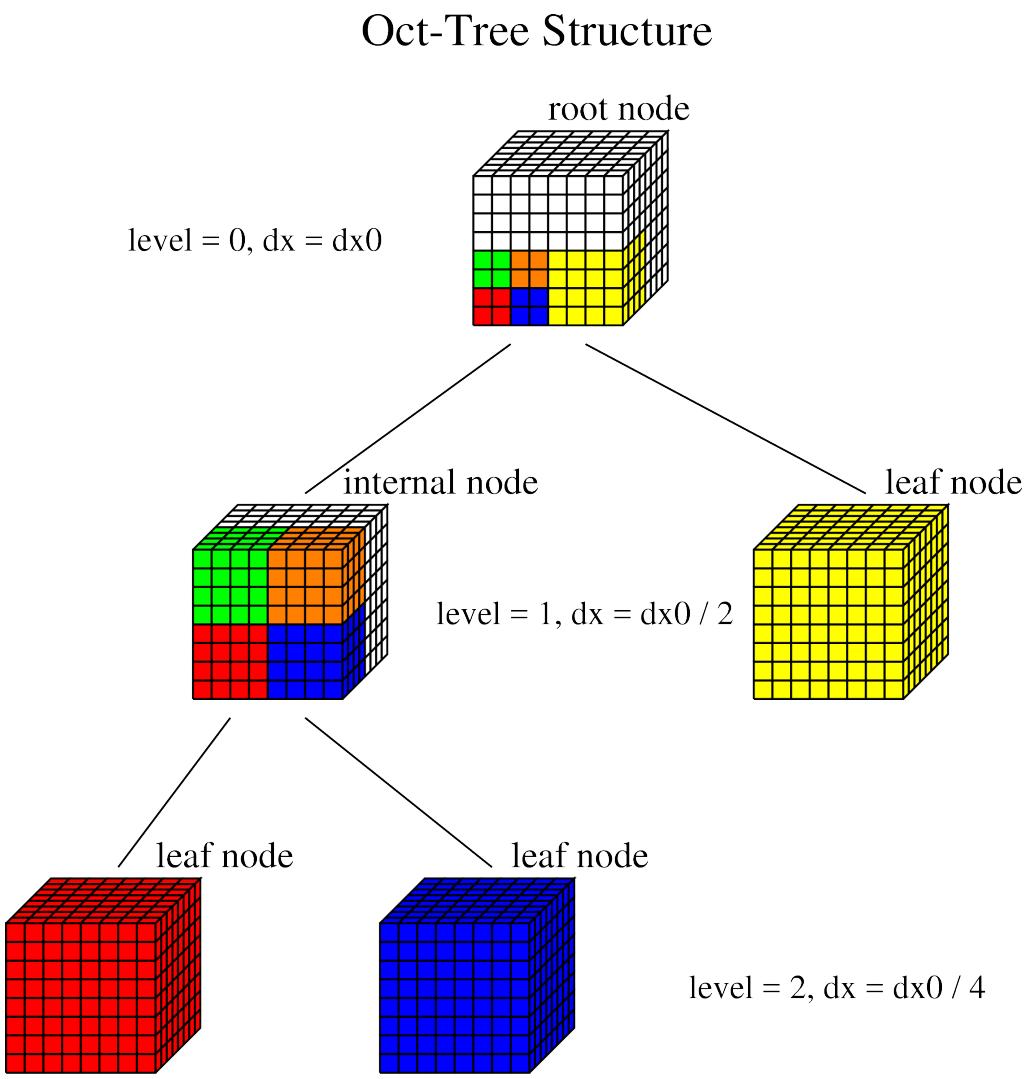


Figure 2: The octree AMR structure - Each Cartesian $8 \times 8 \times 8$ sub-grid can either be entirely refined into 8 child grids (root and internal nodes) or not refined at all (leaf nodes). Note that only two of eight children for the root and internal nodes are actually depicted.

entries. When a sub-grid is adjacent to another sub-grid on the same refinement level, the boundary zones are filled each time sub-step with data from the interior of its neighbor (Figure 3a). When the boundary of a sub-grid has no neighbor at the same level, its boundary data is filled by interpolating the data from its parent's boundary data (Figure 3c). This interpolation is accomplished in the same manner as newly formed child cells are initialized from parent cells. At the end of each time sub-step, the values of parent cells are updated by summing the values in the child cells (Figure 3d). Again, special treatment is needed for the energy, linear, and spin angular momentum to ensure conservation of these quantities is preserved in the integration. In addition to communication of boundary data and restriction of child data onto parent cells, the octree structure is also used to transmit the CFL time-step limit to all nodes in the octree structure ((Figure 3b). The time-step limit is computed for each leaf node and transmitted to its parent. Each parent sub-grid passes the

lowest time-step it receives to its parent, until the root node is reached. Here, the global time-step limit is known and transmitted down the entire octree structure. The basic forms of inter sub-grid communication that Octo-tiger/HPX uses are depicted in Figure 3, except for flux correction. Flux correction at the AMR boundaries is required to enforce conservation because the flux computed at cell faces on an AMR boundary will differ between coarse and fine sub-grids.To account for this, the flux at a cell face on the coarse interface is computed by summing the four fluxes from the cell faces of the corresponding fine cells.

The FMM (described in §4.2) requires data communication patterns similar to that of the hydrodynamics part of the code. Multipole moments are assembled from child multipoles and passed to their parents. Interactions between cells on the same level require boundary communication with neighboring sub-grids. Taylor expansion coefficients are transformed and passed down to children. These communication patterns are shown in Figure 4. To accommodate an opening criterion (described in §4.2) as small as $\frac{1}{3}$ requires a boundary width of 4 cells, so the FMM multipoles are stored in arrays with $(N+8)^3$ entries.

4.2 Gravity

We depart from typical grid based simulations of astrophysical fluids by using an N-body based solver for the gravitational field. The fast multipole method (FMM), first developed by [12], is unique among indirect gravity solvers in that it conserves linear momentum to machine precision. We have taken the Cartesian FMM of [3] and modified it to conserve angular momentum in addition to linear momentum. The FMM relies on the fact that for distant collections of masses, a multipole approximation can provide sufficient accuracy for computation of the gravitational field. There are three phases to the computation: (1) compute the multipole moments for each cell at each level in the AMR structure, (2) compute expansion coefficients for interactions between cells at the same level, and (3) transform theses expansion coefficients from parent grids to child grids.

In the following equations, we denote a multipole of order N, we will use the notation $M^{(N)}$ to refer to the entire multipole, and the notation $M^{(N)}_{i_1 \ldots i_N}$ to refer to a particular element of the multipole, where $i_1 \ldots i_N$ are the indices of that element. When referring to a collection of multipoles, we use $M^{(N)}_{l,i_1 \ldots i_N}$, where l is the index of the multipole and $i_1 \ldots i_N$ is the index of an element in the lth multipole. Taylor expansion coefficients are similarly referred to, except using the letters "L" or "$\mathscr{L}$" in place of "M." The mth component of a position vector is denoted x_m. When referring to a collection of locations within a cell, we use $x_{l,m}$ to refer to the mth component of the lth vector.

At leaf nodes in the AMR structure, each cell is taken as a single monopole, $M^{(0)}_l := \rho_l \left(\Delta x\right)^3$, where Δx is the uniform grid cell spacing at the cell's level. At interior nodes, the multipole moments for each grid cell are computed using the moments from its eight child cells. Octo-tiger/HPX uses multipole moments up to octupole,

$$M^{(0)} := \sum_{l=1}^{8} M^{(0)}_l, \tag{4.2}$$

$$M^{(1)}_m := \sum_{l=1}^{8} \left(M^{(0)}_l x_{l,m} + M^{(1)}_{l,m} \right), \tag{4.3}$$

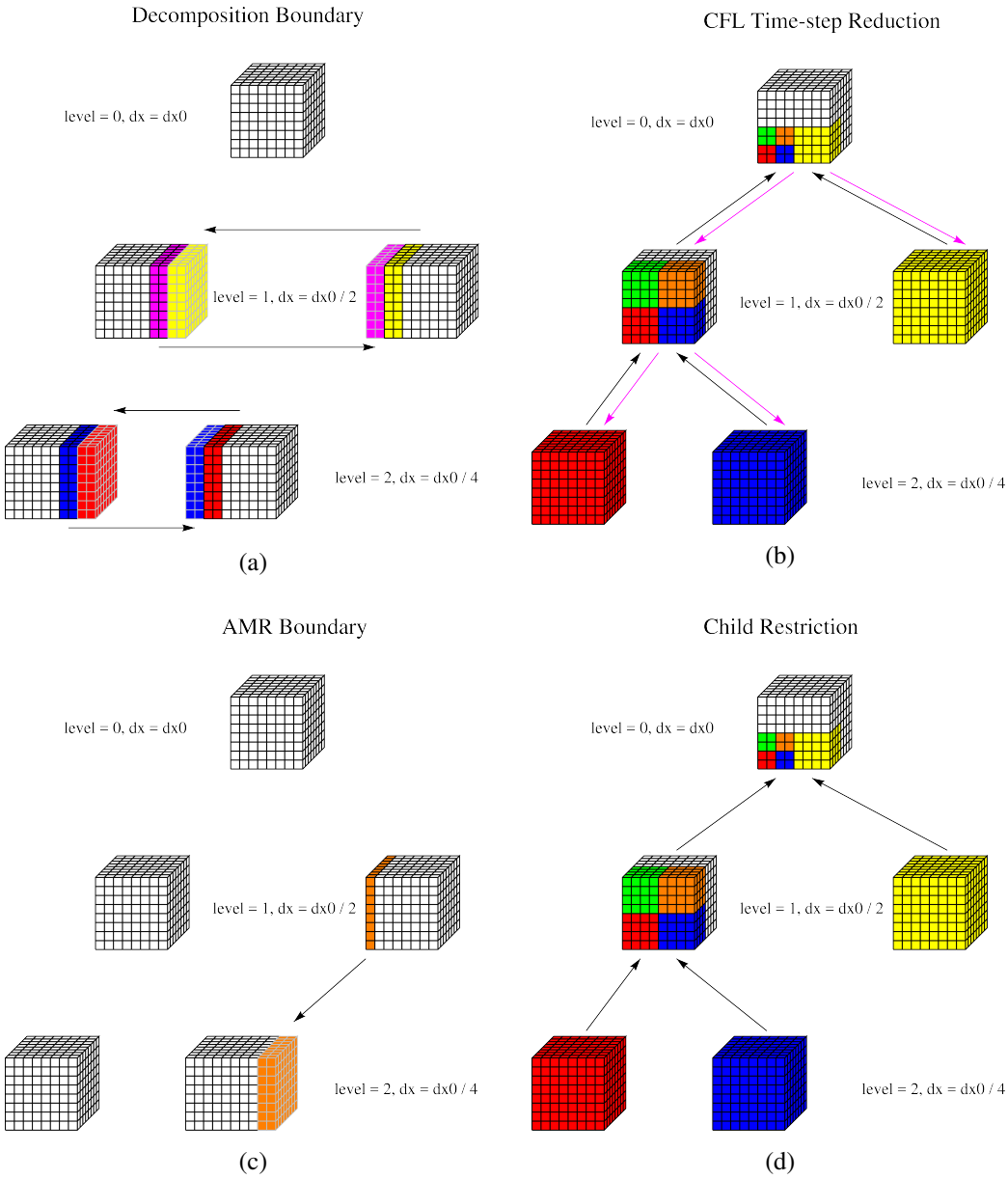


Figure 3: Four forms of inter-nodal communications for hydrodynamics - (a) Adjacent sub-grids on the same refinement level exchange boundary data into ghost cells (cells with gray boundaries). (b) The minimum allowable time-step for each node is communicated to parent nodes (black arrows). At the root node the global minimum time-step is found and communicated to all other nodes (purple arrows). (c) The boundary data for sub-grids with adjacent sub-grids on a more refined level is interpolated into the ghost cells of the more refined sub-grid. (d) Child sub-grids restrict their data onto their parent sub-grid. (Note the actual boundary width is 3 cells).

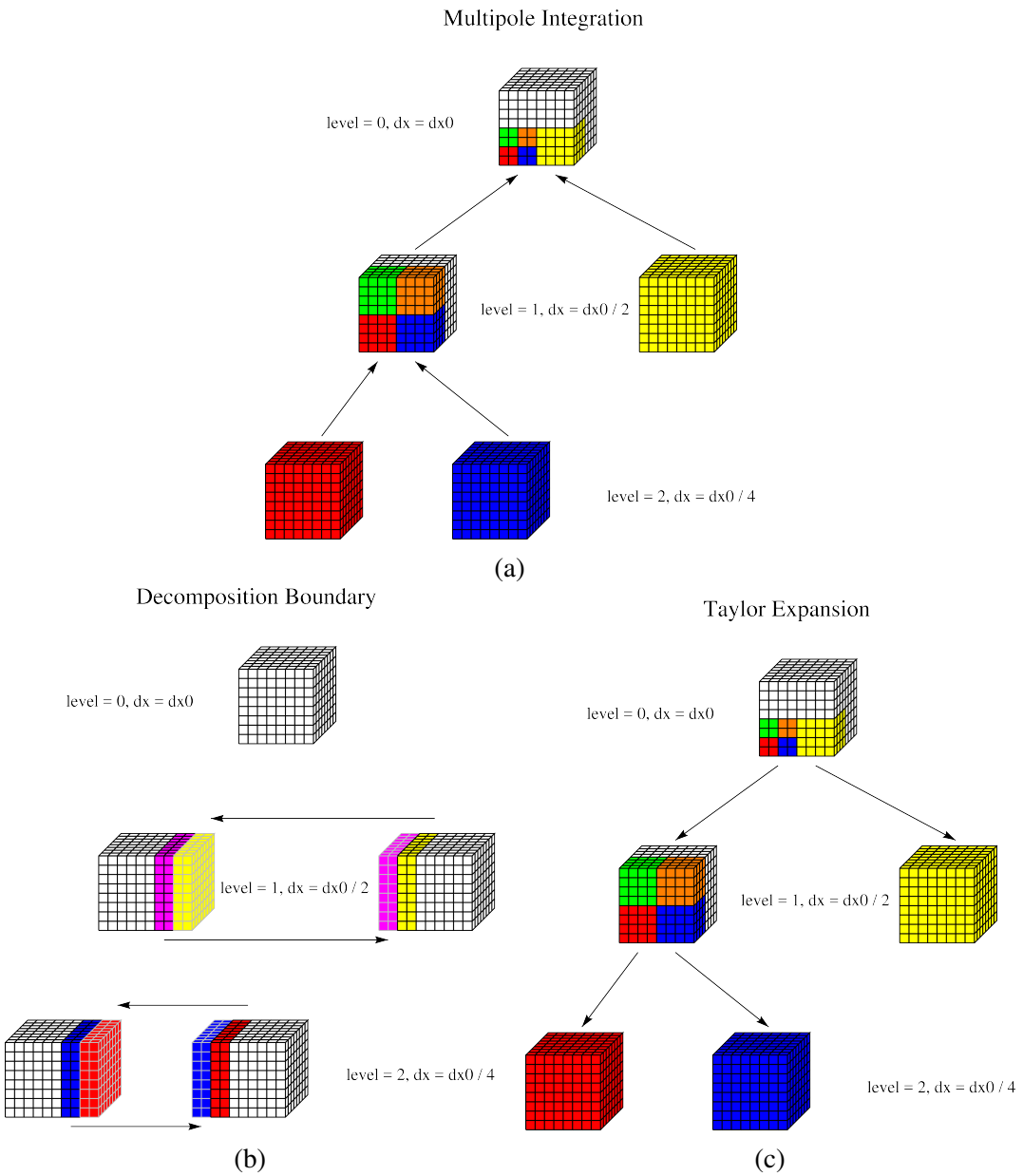


Figure 4: Three forms of inter-nodal communications for FMM gravity - (a) multipole moments are computed by integration of the multipole moments of child cells. (b) Adjacent sub-grids on the same refinement level exchange boundary data into ghost cells (cells with gray boundaries). (c) The values for the gravitational potential and acceleration are computed by Taylor expansion from coarser to finer sub-grids.(Note the actual boundary width is 4 cells).

$$M^{(2)}_{mn} := \sum_{l=1}^{8} \left(M^{(0)}_{l} x_{l,m} x_{l,n} + M^{(1)}_{l,m} x_{l,n} + M^{(2)}_{l,mn} \right), \tag{4.4}$$

and

$$M^{(3)}_{mnp} := \sum_{l=1}^{8} \left(M^{(0)}_{l} x_{l,m} x_{l,n} x_{l,p} + M^{(1)}_{l} x_{l,m} x_{l,n} + M^{(2)}_{l,m} x_{l,n} + M^{(3)}_{l,mnp} \right), \tag{4.5}$$

where $M^{(N)}$ are the multipole moments in the cell of a given interior node, $M^{(N)}_{l}$ are the moments of the eight child cells within that given cell, and x_l is position of the child cell expansion centers relative to the position of the coarse cell's expansion center. When cell expansion centers are chosen to coincide with the cell centers of mass, the dipole moment, $M^{(1)}$, vanishes.

Once the multipole moments are computed, the expansion coefficients for the interactions at each level may be computed. Whether or not a pair of cells in a given level interact with each other is determined by,

$$\frac{1}{\left| \mathbf{Z}^f_1 - \mathbf{Z}^f_2 \right|} \leq \Theta < \frac{1}{\left| \mathbf{Z}^C_1 - \mathbf{Z}^C_2 \right|}, \tag{4.6}$$

where $\mathbf{Z}^f_1$ and $\mathbf{Z}^f_2$ are the coordinate centers of two cells in the given level, and $\mathbf{Z}^C_1$ and $\mathbf{Z}^C_2$ are the coordinate centers of their respective parent coarse cells. The "opening criterion", Θ, can be no larger than $\frac{1}{2}$. At leaf nodes, only the right inequality of equation (4.6) is used to determine if pairs of cells interact and at the root node, only the left inequality of equation (4.6) is used. Smaller values of Θ result in higher accuracy at the expense of more computations and boundary data exchanges. Octo-tiger/HPX allows for a choice of Θ between $\frac{1}{3}$ and $\frac{1}{2}$. Computing the interaction coefficients require gradients of the Green's function for the gravitational potential, $\nabla^{\mathbf{n}} g(R) = -\nabla^{\mathbf{n}} \frac{1}{R}$ These are

$$D^{(0)}(\mathbf{R}) := -\frac{1}{R}, \tag{4.7}$$

$$D^{(1)}_{i}(\mathbf{R}) := \frac{R_i}{R^3}, \tag{4.8}$$

$$D^{(2)}_{ij}(\mathbf{R}) := -\frac{3R_iR_j - \delta_{ij}R^2}{R^5}, \tag{4.9}$$

and

$$D^{(3)}_{ijk}(\mathbf{R}) := \frac{15R_iR_jR_k - 3\left(\delta_{ij}R_k + \delta_{jk}R_i + \delta_{ki}R_j\right)R^2}{R^7}. \tag{4.10}$$

The Taylor expansion coefficients at a given level result from intra-level interactions in addition to the transformation of expansions at coarser levels. The expansion coefficients at a given level are computed using

$$L^{f,(0)}_{q} := \sum_{l} \left[M^{(0)}_{l} D^{(0)}\left(\mathbf{R}_{l,q}\right) + M^{(1)}_{l,m} D^{(1)}_{m}\left(\mathbf{R}_{l,q}\right) + M^{(2)}_{l,mn} D^{(2)}_{mn}\left(\mathbf{R}_{l,q}\right) + M^{(3)}_{l,mnp} D^{(3)}_{mnp}\left(\mathbf{R}_{l,q}\right) \right] + L^{C,(0)}_{q} - L^{C,(1)}_{q,m} x_m + L^{C,(2)}_{q,mn} x_n x_m - L^{C,(2)}_{q,mnp} x_n x_m x_p \tag{4.11}$$

$$L^{f,(1)}_{q,m} := \sum_{l} \left[M^{(0)}_{l} D^{(1)}_{m}\left(\mathbf{R}_{l,q}\right) + M^{(1)}_{l,n} D^{(2)}_{mn}\left(\mathbf{R}_{l,q}\right) + M^{(2)}_{l,np} D^{(3)}_{mnp}\left(\mathbf{R}_{l,q}\right) \right] + L^{C,(1)}_{q,m} - L^{C,(2)}_{q,mn} x_n + L^{C,(2)}_{q,mnp} x_n x_p \tag{4.12}$$

$$L^{f,(2)}_{q,mn} := \sum_l \left[M^{(0)}_l D^{(1)}_{mn}\left(\mathbf{R}_{l,q}\right) + M^{(1)}_{l,p} D^{(2)}_{mnp}\left(\mathbf{R}_{l,q}\right) + \right] + L^{C,(1)}_{q,mn} - L^{C,(2)}_{q,mnp} x_p \tag{4.13}$$

and

$$L^{f,(3)}_{q,mnp} := \sum_l M^{(0)}_l D^{(1)}_{mnp}\left(\mathbf{R}_{l,q}\right) + L^{C,(1)}_{q,mnp}, \tag{4.14}$$

where $\mathbf{R}_{l,q} := \mathbf{R}_l - \mathbf{R}_q$ is the distance between the expansion centers of cell l and cell q, $L^{f,(N))}_q$ are Taylor coefficients at cell q, $L^{C,(N))}_q$ are Taylor expansion coefficients from the parent cell of cell q, and $\mathbf{x}$ is the distance from the expansion center of cell q to the expansion center of it's parent cell. The summation $\sum_l$, denotes summation over all cells at the given level which satisfy equation (4.6) for the cell q at the same level. Summation is also performed over indexes m, n, and p.

As shown by [3], the sum of all gravitational forces computed using equations (4.2) - (4.14) is zero, meaning linear momentum is conserved to machine precision. As we show in [13], angular momentum is not similarly conserved. It is possible to add a corrective term to the $L^{(1)}$ expansion coefficients that results in conservation of both linear and angular momentum to machine precision. This term is

$$\mathscr{L}^{f,(1)}_{q,m} := \tfrac{1}{2} \sum_l \left[\left(M^{(2)}_{l,npr} - \frac{M^{(0)}_l}{M^{(0)}_q} M^{(2)}_{q,npr} \right) D'^{(4)}_{mnpr}\left(\mathbf{R}_{l,q}\right) \right] + \mathscr{L}^{C,(1)}_{q,m}, \tag{4.15}$$

where $\mathscr{L}^{f,(1)}_q$ is the corrective force on cell q, $\mathscr{L}^{C,(1)}_{q,m}$ is the corrective force in the parent cell of cell q, and

$$D'^{(4)}_{mnpr}\left(\mathbf{R}\right) := + \frac{15\left(\delta_{ij}R_kR_l + \delta_{ik}R_jR_l + \delta_{il}R_jR_k\right)}{R^7} - \frac{3\left(\delta_{ij}\delta_{kl} + \delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}\right)}{R^5}. \tag{4.16}$$

4.3 Hydrodynamics

The numerical solution to equations (3.1) - (3.4) are cast into semi-discrete form using the advection scheme of [14] (KT scheme). High resolution grid based hydrodynamics methods such as the KT scheme conserve linear momentum to machine precision, however, they do not similarly conserve angular momentum. The discretization of the flux of linear momentum components parallel to cell faces does not reproduce the symmetry present in the fluid stress-energy tensor, causing a violation of the conservation of angular momentum. [7] address this issue by evolving the spin angular momentum, $\mathbf{z}$, of each cell along with the usual linear momenta, $\mathbf{s}$, and using $\mathbf{z}$ as the anti-symmetric part of the velocity gradient for computing velocities at cell faces. Evolving $\mathbf{z}$ alone merely tracks the error in angular momentum conservation. Applying $\mathbf{z}$ to the reconstruction of velocities causes the error in angular momentum conservation to feed back into the linear momenta.

The semi-discrete form of the evolution equations are

$$\frac{d}{dt}\mathscr{U}_{ijk} + \frac{1}{\Delta x}\left(\mathscr{F}_{i+\frac{1}{2}jk} - \mathscr{F}_{i-\frac{1}{2}jk} + \mathscr{F}_{ij+\frac{1}{2}k} - \mathscr{F}_{ij-\frac{1}{2}k} + \mathscr{F}_{ijk+\frac{1}{2}} - \mathscr{F}_{ijk-\frac{1}{2}} \right) = \mathscr{S}_{ijk}. \tag{4.17}$$

The evolved state vector, $\mathscr{U}$, is defined as

$$\mathscr{U} := \begin{bmatrix} \rho \\ s_x \\ s_y \\ s_z \\ E \\ \tau \\ z_x \\ z_y \\ z_z \end{bmatrix}. \tag{4.18}$$

The evolved state vector is reconstructed on the right and left sides of each cell face, $\mathscr{U}^{\mathbf{R}}{}_{i+\frac{1}{2}jk}$, $\mathscr{U}^{\mathbf{R}}{}_{ij+\frac{1}{2}k}$, $\mathscr{U}^{\mathbf{R}}{}_{ijk+\frac{1}{2}}$, $\mathscr{U}^{\mathbf{L}}{}_{i+\frac{1}{2}jk}$, $\mathscr{U}^{\mathbf{L}}{}_{ij+\frac{1}{2}k}$, and $\mathscr{U}^{\mathbf{L}}{}_{ijk+\frac{1}{2}}$. To accomplish this, the primitive variables are computed from the cell centered state vector. With the exception of velocities parallel to a cell face, these variables are then reconstructed at cell faces using the PPM of [15] .

Velocities parallel to cell faces are determined by adapting the technique described in [7]. First, the velocity gradient, $(\nabla \mathbf{v})_1$ is computed using a minmod limiter,

$$\text{minmod}\,(a,b) := \tfrac{1}{2}\left(\text{sgn}\,(a) + \text{sgn}\,(b)\right)\min\left(|a|\,,|b|\right). \tag{4.19}$$

The components of $(\nabla \mathbf{v})_1$ are

$$\left(\frac{d\mathbf{v}}{dx}\right)_{1,ijk} := \text{minmod}\left(\mathbf{v}_{i+1jk} - \mathbf{v}_{ijk}, \mathbf{v}_{ijk} - \mathbf{v}_{i-1jk}\right), \tag{4.20}$$

$$\left(\frac{d\mathbf{v}}{dy}\right)_{1,ijk} := \text{minmod}\left(\mathbf{v}_{ij+1k} - \mathbf{v}_{ijk}, \mathbf{v}_{ijk} - \mathbf{v}_{ij-1k}\right), \tag{4.21}$$

and

$$\left(\frac{d\mathbf{v}}{dz}\right)_{1,ijk} := \text{minmod}\left(\mathbf{v}_{ijk+1} - \mathbf{v}_{ijk}, \mathbf{v}_{ijk} - \mathbf{v}_{ijk-1}\right). \tag{4.22}$$

The symmetric part of the second set of slopes, $(\nabla \mathbf{v})_2$, is determined using the slopes from equations (4.20) - (4.22), while the anti-symmetric part is computed using $\mathbf{z}$,

$$(\nabla \mathbf{v})_2 := \frac{1}{2}\begin{bmatrix} 2\left(\frac{dv_x}{dx}\right)_1 & \left(\frac{dv_x}{dy}\right)_1 + \left(\frac{dv_y}{dx}\right)_1 & \left(\frac{dv_x}{dz}\right)_1 + \left(\frac{dv_z}{dx}\right)_1 \\ \left(\frac{dv_x}{dy}\right)_1 + \left(\frac{dv_y}{dx}\right)_1 & 2\left(\frac{dv_y}{dy}\right)_1 & \left(\frac{dv_y}{dz}\right)_1 + \left(\frac{dv_z}{dy}\right)_1 \\ \left(\frac{dv_x}{dz}\right)_1 + \left(\frac{dv_z}{dx}\right)_1 & \left(\frac{dv_x}{dz}\right)_1 + \left(\frac{dv_y}{dz}\right)_1 & 2\left(\frac{dv_z}{dz}\right)_1 \end{bmatrix} + \frac{6}{\Delta x}\begin{bmatrix} 0 & -z_z & z_y \\ z_z & 0 & -z_x \\ -z_y & z_x & 0 \end{bmatrix}. \tag{4.23}$$

We have omitted the ijk subscripts on every variable for brevity. The final set of slopes, $(\nabla \mathbf{v})_3$, is obtained by applying another minmod limiter to the first two sets of slopes,

$$(\nabla \mathbf{v})_3 := \text{minmod}\left((\nabla \mathbf{v})_2, 2\,(\nabla \mathbf{v})_1\right). \tag{4.24}$$

The purpose of this transformation is to reproduce the asymmetrical part of the velocity gradient implied by $\mathbf{z}$ as closely as possible without compromising the stability of the numerical solution.

We now define $\mathbf{z}'$ as the difference between the evolved spin angular momenta, $\mathbf{z}$, and the spin angular momenta as measured by the computed slopes, $(\Delta x)\,(\hat{x}\times(\nabla\mathbf{v})_3)$,

$$z_x^{'} := z_x - \tfrac{\Delta x}{12}\left(\left(\frac{dv_y}{dz}\right)_3 - \left(\frac{dv_z}{dy}\right)_3\right), \tag{4.25}$$

$$z_y^{'} := z_y - \tfrac{\Delta x}{12}\left(\left(\frac{dv_z}{dx}\right)_3 - \left(\frac{dv_x}{dz}\right)_3\right), \tag{4.26}$$

and

$$z_z^{'} := z_z - \tfrac{\Delta x}{12}\left(\left(\frac{dv_x}{dy}\right)_3 - \left(\frac{dv_y}{dx}\right)_3\right). \tag{4.27}$$

The quantity, $\mathbf{z}$, is therefore a measure of spin angular momentum that is unable to appear in the computed velocity slopes without causing numerical instability.

With the primitive variables computed at cell faces, we may also compute $\mathscr{U}$. The flux, $\mathscr{F}$, also requires we know ϕ at cell faces. Because the gravitational potential is a smooth function, these values are taken as continuous across cell faces and are computed using only Equation 1.9 from [15]. The flux, $\mathscr{F}$, is the sum of the physical flux, $\mathscr{G}$, and the Kurganov Tadmor flux (KT flux), $\mathscr{H}$,

$$\mathscr{F} := \mathscr{G} + \mathscr{H}. \tag{4.28}$$

The physical flux, $\mathscr{G}$, is computed using the divergence terms on the left hand side (LHS) of equations (3.1) - (3.4) for the first six components. The flux of $\mathbf{z}$ is taken to be the physical advection of the angular momentum error not represented in the velocity reconstruction, $\mathbf{v}\mathbf{z}'$. The resulting physical flux definition is

$$\mathscr{G}_x := \begin{bmatrix} v_x\rho \\ v_x s_x + p \\ v_x s_y \\ v_x s_z \\ v_x E + u_x\,(p+\rho\phi) \\ v_x\tau \\ v_x z_x^{'} \\ v_x z_y^{'} \\ v_x z_z^{'} \end{bmatrix}, \quad \mathscr{G}_y := \begin{bmatrix} v_y\rho \\ v_y s_x \\ v_y s_y + p \\ v_y s_z \\ v_y E + u_y\,(p+\rho\phi) \\ v_y\tau \\ v_y z_x^{'} \\ v_y z_y^{'} \\ v_y z_z^{'} \end{bmatrix}, \quad \mathscr{G}_z := \begin{bmatrix} v_z\rho \\ v_z s_x \\ v_z s_y \\ v_z s_z + p \\ v_z E + u_z\,(p+\rho\phi) \\ v_z\tau \\ v_z z_x^{'} \\ v_z z_y^{'} \\ v_z z_z^{'} \end{bmatrix}, \tag{4.29}$$

The physical flux, $\mathscr{G}$, is generally a function of $\mathscr{U}$ and ϕ. When discretized, it is taken as the average of left and right fluxes,

$$\mathscr{G}_{x,i+\frac{1}{2}jk} := \tfrac{1}{2}\left[\mathscr{G}_x\left(\mathscr{U}^{\mathbf{R}}{}_{i+\frac{1}{2}jk}, \phi_{i+\frac{1}{2}jk}\right) + \mathscr{G}_x\left(\mathscr{U}^{\mathbf{L}}{}_{i+\frac{1}{2}jk}, \phi_{i+\frac{1}{2}jk}\right)\right], \tag{4.30}$$

$$\mathscr{G}_{y,ij+\frac{1}{2}k} := \tfrac{1}{2}\left[\mathscr{G}_y\left(\mathscr{U}^{\mathbf{R}}{}_{ij+\frac{1}{2}k}, \phi_{ij+\frac{1}{2}k}\right) + \mathscr{G}_y\left(\mathscr{U}^{\mathbf{L}}{}_{ij+\frac{1}{2}k}, \phi_{ij+\frac{1}{2}k}\right)\right], \tag{4.31}$$

and

$$\mathscr{G}_{z,ijk+\frac{1}{2}} := \tfrac{1}{2}\left[\mathscr{G}_z\left(\mathscr{U}^{\mathbf{R}}{}_{ijk+\frac{1}{2}}, \phi_{ijk+\frac{1}{2}}\right) + \mathscr{G}_z\left(\mathscr{U}^{\mathbf{L}}{}_{ijk+\frac{1}{2}}, \phi_{ijk+\frac{1}{2}}\right)\right] \tag{4.32}$$

The KT flux, $\mathscr{H}$, is computed at cell faces using

$$\mathscr{H}_{i+\frac{1}{2}jk} := \tfrac{1}{2} a_{i+\frac{1}{2}jk} \left(\mathscr{W}^{\mathbf{R}}{}_{i+\frac{1}{2}jk} - \mathscr{W}^{\mathbf{L}}{}_{i+\frac{1}{2}jk} \right), \tag{4.33}$$

$$\mathscr{H}_{ij+\frac{1}{2}k} := \tfrac{1}{2} a_{ij+\frac{1}{2}k} \left(\mathscr{W}^{\mathbf{R}}{}_{ij+\frac{1}{2}k} - \mathscr{W}^{\mathbf{L}}{}_{ij+\frac{1}{2}k} \right), \tag{4.34}$$

$$\mathscr{H}_{ijk+\frac{1}{2}} := \tfrac{1}{2} a_{ijk+\frac{1}{2}} \left(\mathscr{W}^{\mathbf{R}}{}_{ijk+\frac{1}{2}} - \mathscr{W}^{\mathbf{L}}{}_{ijk+\frac{1}{2}} \right). \tag{4.35}$$

The conserved state vector, $\mathscr{W}$, is defined in the same locations as $\mathscr{U}$ and differs from $\mathscr{U}$ in only the energy component. It is

$$\mathscr{W} := \mathscr{U} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \rho\phi \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{4.36}$$

Globally, the sum of gas and potential energies, $E + \frac{1}{2}\rho\phi$, is conserved. As discussed in [5], however, the quantity $E + \rho\phi$ is the "locally conserved" quantity. The difference, $-\frac{1}{2}\rho\phi$, comes into play because of changes in the global gravitational field. Therefore the KT flux is computed using $E + \rho\phi$.

The maximum signal speed at cell faces is computed using

$$a_{i+\frac{1}{2},j,k} := \max\left(\left| v^R_{x,i+\frac{1}{2}jk} \right| + c^R_{i+\frac{1}{2}jk}, \left| v^L_{x,i+\frac{1}{2}jk} \right| + c^L_{i+\frac{1}{2}jk} \right), \tag{4.37}$$

$$a_{i,j+\frac{1}{2},k} := \max\left(\left| v^R_{y,ij+\frac{1}{2}k} \right| + c^R_{ij+\frac{1}{2}k}, \left| v^L_{y,ij+\frac{1}{2}k} \right| + c^L_{ij+\frac{1}{2}k} \right), \tag{4.38}$$

and

$$a_{i,j,k+\frac{1}{2}} := \max\left(\left| v^R_{z,ijk+\frac{1}{2}} \right| + c^R_{ijk+\frac{1}{2}}, \left| v^L_{z,ijk+\frac{1}{2}} \right| + c^L_{ijk+\frac{1}{2}} \right). \tag{4.39}$$

The sound speed, c, is defined as the only non-zero eigenvalue of the Jacobian matrix, $\frac{\partial \mathscr{F}}{\partial \mathscr{U}}$, when $\mathbf{v}$ is set to zero. For the ideal gas case, $c_{\text{ideal}} := \sqrt{\gamma(\gamma-1)\varepsilon}$.

The source vector, $\mathscr{S}$, is defined as

$$\mathscr{S} := \begin{bmatrix} 0 \\ \rho g_x - \Omega s_y \\ \rho g_y + \Omega s_x \\ \rho g_z \\ \frac{1}{2}\left(\rho \frac{\partial}{\partial t}\phi - \phi \frac{\partial}{\partial t}\rho\right) - \frac{1}{2}\frac{\partial}{\partial t}\rho\phi \\ 0 \\ \mathscr{F}_{y,s_z} - \mathscr{F}_{z,s_y} \\ -\mathscr{F}_{x,s_z} + \mathscr{F}_{z,s_x} \\ \mathscr{F}_{x,s_y} - \mathscr{F}_{y,s_x} \end{bmatrix}. \tag{4.40}$$

The gravitational quantities $\mathbf{g}$, ϕ, and $\frac{\partial}{\partial t}\phi$ are computed using our modified version of the the fast multipole method (FMM). The final term in the energy source term, $\frac{1}{2}\frac{\partial}{\partial t}\rho\phi$, is computed directly using the differences between $\rho\phi$ at successive time sub-steps. The source terms for $\mathbf{z}$ should physically be zero, however, as mentioned above, the discretization of the flux violates the symmetry in the stress-energy tensor and these terms are non-zero. The discretized source is

$$\mathscr{S}_{ijk} := \begin{bmatrix} 0 \\ \rho_{ijk} g_{x,ijk} - s_{y,ijk}\Omega \\ \rho_{ijk} g_{y,ijk} + s_{x,ijk}\Omega \\ \rho_{ijk} g_{z,ijk} \\ \frac{1}{2}\left(\rho_{ijk}\left(\frac{d}{dt}\phi\right)_{ijk} - \phi_{ijk}\left(\frac{d}{dt}\rho\right)_{ijk}\right) - \frac{1}{2}\left(\frac{d}{dt}\rho\phi\right)_{ijk} \\ 0 \\ \frac{1}{2}\mathscr{F}_{y,s_z,ij+\frac{1}{2}k} + \frac{1}{2}\mathscr{F}_{y,s_z,ij-\frac{1}{2}k} - \frac{1}{2}\mathscr{F}_{z,s_y,ijk+\frac{1}{2}} - \frac{1}{2}\mathscr{F}_{z,s_y,ijk-\frac{1}{2}} \\ -\frac{1}{2}\mathscr{F}_{x,s_z,i+\frac{1}{2}jk} - \frac{1}{2}\mathscr{F}_{x,s_z,i-\frac{1}{2}jk} + \frac{1}{2}\mathscr{F}_{z,s_x,ijk+\frac{1}{2}} + \frac{1}{2}\mathscr{F}_{z,s_x,ijk-\frac{1}{2}} \\ \frac{1}{2}\mathscr{F}_{x,s_y,i+\frac{1}{2}jk} + \frac{1}{2}\mathscr{F}_{x,s_y,i-\frac{1}{2}jk} - \frac{1}{2}\mathscr{F}_{y,s_x,ij+\frac{1}{2}k} - \frac{1}{2}\mathscr{F}_{y,s_x,ij-\frac{1}{2}k} \end{bmatrix}. \tag{4.41}$$

The value of ϕ_{ijk} and $\mathbf{g}_{ijk}$ needed to compute $\mathscr{S}_{ijk}$ is determined using the FMM. Since the evolution equation for ρ has a zero source term, $\frac{d}{dt}\rho_{ijk}$ is known once the fluxes are known, allowing the computation of $\frac{d}{dt}\phi_{ijk}$ with the FMM. The expansion centers for both of these computations are taken at the centers of mass at each cell. For a given set of expansion centers, the value calculated by the FMM in a particular cell can be written as a linear combination of the source values in each of the cells throughout the grid. By keeping the same expansion centers for both the computation of ϕ_{ijk} and $\frac{d}{dt}\phi_{ijk}$, we ensure that the first energy source term, $\rho_{ijk}\frac{d}{dt}\phi_{ijk} - \phi_{ijk}\frac{d}{dt}\rho_{ijk}$, sums to zero globally.

The equation (4.17) is advanced in time using the 2nd order Runge-Kutta integrator of [16]. The maximum allowable time-step is computed according to the Courant Friedrichs Lewy (CFL) condition at the start of the first Runga-Kutta sub-step,

$$\Delta t := \frac{f_{\text{CFL}}\Delta x}{\max_{\text{all}ijk}\left(c^R_{i+\frac{1}{2}jk}, c^R_{ij+\frac{1}{2}k}, c^R_{ijk+\frac{1}{2}}, c^L_{i+\frac{1}{2}jk}, c^L_{ij+\frac{1}{2}k}, c^L_{ijk+\frac{1}{2}}\right)}. \tag{4.42}$$

The CFL factor, f_{CFL}, is set to $\frac{2}{15}$. This ensures both stability and the positivity of ρ and τ.

The entropy tracer, τ, is updated according to the dual energy formalism,

$$\tau_{ijk} \rightarrow \begin{cases} \left(E_{ijk} - \frac{1}{2}\rho_{ijk}v_{ijk}^2\right)^{\frac{1}{\gamma}} & E_{ijk} - \frac{1}{2}\rho_{ijk}v_{ijk}^2 \geq 0.1E_{\text{neighborhood},ijk} \\ \tau_{ijk} & \text{else} \end{cases}, \tag{4.43}$$

where $E_{\text{neighborhood},ijk} := \max\left(E_{ijk}, E_{i+\frac{1}{2}jk}, E_{ij+\frac{1}{2}k}, E_{ijk+\frac{1}{2}}, E_{i-\frac{1}{2}jk}, E_{ij-\frac{1}{2}k}, E_{ijk-\frac{1}{2}}\right)$. Note that unlike equation (3.5), which is used to compute the specific internal gas energy for the equation of state, equation (4.43) requires information from neighboring cells. Equation (4.43) also results in an update to τ itself, while equation (3.5) does not alter the value of τ stored in $\mathscr{U}$.

4.4 Initial Models - the Self Consistent Field (SCF) Technique

To generate an initial equilibrium binary star model, Octo-tiger/HPX uses the self consistent field (SCF) technique [17, 18]. The SCF technique relies on the fact that the hydrostatic balance equation in the rotating frame,

$$\nabla H + \nabla \Psi = 0, \tag{4.44}$$

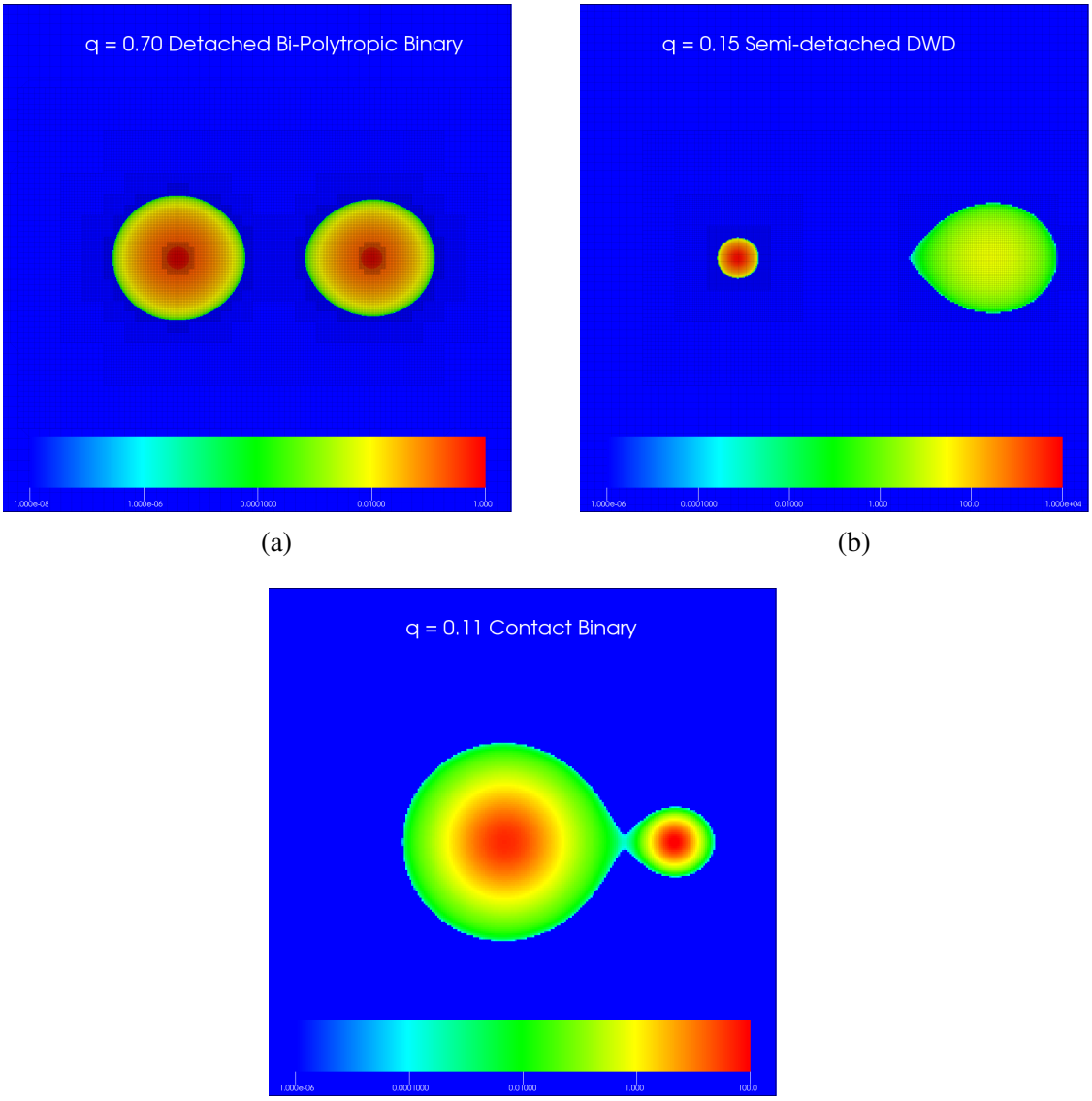
where H is the isentropic enthalpy, $H[\rho] := \int_0^{P=P[\rho]} \frac{dP'}{\rho'}$, and $\Psi := \phi - \frac{1}{2}\Omega^2\left(x^2+y^2\right)$ is the "effective potential", can be written as an algebraic equation,

$$H + \Psi = \Psi_0, \tag{4.45}$$

where Ψ_0 is a constant. Each star may have a different value for Ψ_0. By setting the Ψ_0 based on the value of the effective potential at the L1 Lagrange point, Ψ_{L1}, initial models can be created that are detached, semi-detached or in contact. For detached models, $\Psi_0 < \Psi_{L1}$ for both stars, for semi-detached models, $\Psi_0 < \Psi_{L1}$ for the accretor and $\Psi_0 = \Psi_{L1}$ for the donor, and for contact models $\Psi_0 \geq \Psi_{L1}$ for both stars (but less than the potential at the L2 Lagrange point). Octo-tiger/HPX provides for a choice between three structural equations of state: (1) a polytropic equation of state, (2) a bi-polytropic equation of state [19], and (3) a cold white dwarf equation of state. Figure 5 depicts the density profiles through the equatorial slices of three such initial conditions.

5. Example Models

Here we present two models of interacting binaries as an example of Octo-tiger/HPX's capabilities. The first begins with mass ratio $q = 0.7$ bi-polytropic binary in near equilibrium. The polytropic indexes are $n = 3$ for the core and $n = \frac{3}{2}$ for envelope of each star. This approximates a star with a radiative interior and convective envelope. The density jumps by a factor of two at the core-envelope interface. An extra level of resolution is added to the cores to better resolve this contact discontinuity. The equation of state for the evolution of the binary is that of an ideal gas. When the simulation starts, a small amount of angular momentum is artificially removed from the binary in order to initiate mass transfer. The sub-grids are $8 \times 8 \times 8$ cells. The number of sub-grids varies as the simulation progresses. The typical number of sub-grids throughout the simulation is $40,000$, containing a total of $20,000,000$ computational cells. This configuration is unstable to mass transfer and results in tidal disruption and merger after about 13 orbits. There are two movies of this



(a) (b)

Figure 5: Equatorial density slices of initial binary star conditions produced with Octo-tiger/HPX. (a) A detached binary of mass ratio $q = 0.7$. The stars in this model use a bi-polytropic structural equation of state, Note the core fractions are each given an extra level of resolution. (b) A semi-detached binary of mass ratio $q = 0.15$. This model was produced using a degenerate equation of state. The donor star exactly fills its Roche lobe. (c) A contact binary. This model was produced using a bi-polytropic equation of state. The polytropic constants of each star's envelope are set to be equal. Both stars slightly overfill their Roche lobes.

simulation available on the WWW. The first, at https://www.youtube.com/watch?v=Ho6LoPxa0kI, depicts the the equatorial slice of mass density for the entire computational grid. The second, at https://www.youtube.com/watch?v=4jvEvvVfe4Y, shows a close up of the same view. This simulation is discussed and analyzed by [20]. Note that the rotational velocity of the AMR grid, Ω, is set to the rotational velocity of the binary at $t = 0$ and keep constant throughout.

The second example simulation is of a double white dwarf (DWD) of mass ratio $q = 0.2$. The accretor is a CO white dwarf of $1M_{\odot}$ and the donor is a He white dwarf of $0.2M_{\odot}$. The initial orbital period is 242 seconds. This simulation was conducted using an earlier version of Octo-tiger/HPX than is presented here. This earlier version conserved z-angular momentum at the expense of violating conservation of x and y linear momentum, as described by [2]. We present the results from this earlier version to demonstrate the usefulness of machine precision angular momentum conservation when simulating interacting binaries that are (potentially) stable to mass transfer. Mass transfer is expected to be stable for DWDs of this mass ratio [21], and they are expected to form AM CvN systems [e.g. 22, 23] when mass transfer ensues. The simulation ran for over 525 orbital periods. To our knowledge, no fully three dimensional simulation of binary stars with self-consistent hydrodynamic modeling of each of the components has run for this many orbital periods. A movie of the simulation can be found at https://www.youtube.com/watch?v=X5xIBzEyFl8. Unlike in the first example movie, Ω is varied throughout the evolution so as to keep the binary's line of centers coincident with the x-axis. As shown in the movie, the mass transfer rate initially rises, peaking at around 40 orbits. After that it decreases for the entire remainder of the simulation as the binary separates.

6. Conclusions

Here we have presented angular momentum conserving numerical methods that allow for the numerical modeling of classically self-gravitating astrophysical fluids. We have implemented these methods in Octo-tiger/HPX for execution on distributed computing systems. We have presented two models of interacting binary systems, one of which simulates stable mass transfer for over 500 orbital periods.

Octo-tiger/HPX is available on the WWW as a github.com repository at https://github.com/STEllAR-GROUP/octotiger. The HPX runtime system is available at https://github.com/STEllAR-GROUP/hpx [24].

7. Acknowledgments

We wish to acknowledge the support from the National Science Foundation through INSPIRE grant AST- 1240655.

Portions of this research were conducted with high performance computing resources provided by Louisiana State University (http://www.hpc.lsu.edu).

Portions of this research were conducted with high performance computational resources provided by the Louisiana Optical Network Initiative (http://www.loni.org).

The authors would like to thank the organizing committee of Accretion Processes in Cosmic Sources 2016 for the opportunity to present our work at the conference and in this publication.

References

[1] K. A. Postnov and L. R. Yungelson, *The Evolution of Compact Binary Star Systems*, *Living Reviews in Relativity* **17** (2014) 3, [1403.4754].

[2] Z. D. Byerly, B. Adelstein-Lelbach, J. E. Tohline and D. C. Marcello, *A Hybrid Advection Scheme for Conserving Angular Momentum on a Refined Cartesian Mesh*, *Astrophysical Journal Supplement* **212** (2014) 23, [1404.5942].

[3] W. Dehnen, *A Very Fast and Momentum-conserving Tree Code*, *Astrophysical Journal Letters* **536** (2000) L39–L42, [arXiv:astro-ph/0003209].

[4] P. M. Motl, J. E. Tohline and J. Frank, *Numerical Methods for the Simulation of Dynamical Mass Transfer in Binaries*, *Astrophysical Journal Supplement* **138** (2002) 121–148, [arXiv:astro-ph/0107308].

[5] D. C. Marcello and J. E. Tohline, *A Numerical Method for Studying Super-Eddington Mass Transfer in Double White Dwarf Binaries*, *Astrophysical Journal Supplement* **199** (2012) 35.

[6] M. Dan, S. Rosswog, J. Guillochon and E. Ramirez-Ruiz, *Prelude to A Double Degenerate Merger: The Onset of Mass Transfer and Its Impact on Gravitational Waves and Surface Detonations*, *Astrophysical Journal* **737** (2011) 89, [1101.5132].

[7] B. Després and E. Labourasse, *Angular momentum preserving cell-centered Lagrangian and Eulerian schemes on arbitrary grids*, *Journal of Computational Physics* **290** (2015) 28–54.

[8] H. Kaiser, T. Heller, D. Bourgeois and D. Fey, *Higher-level parallelization for local and distributed asynchronous task-based programming*, in *Proceedings of the First International Workshop on Extreme Scale Programming Models and Middleware*, ESPM '15, (New York, NY, USA), pp. 29–37, ACM, 2015. DOI.

[9] M. Anderson, M. Brodowicz, H. Kaiser and T. Sterling, *An Application Driven Analysis of the ParalleX Execution Model*, *ArXiv e-prints* (2011) , [1109.5201].

[10] L. Segretain, G. Chabrier and R. Mochkovitch, *The Fate of Merging White Dwarfs*, *Astrophysical Journal* **481** (1997) 355.

[11] G. L. Bryan, M. L. Norman, J. M. Stone, R. Cen and J. P. Ostriker, *A piecewise parabolic method for cosmological hydrodynamics*, *Computer Physics Communications* **89** (1995) 149–168.

[12] L. Greengard and V. Rokhlin, *A Fast Algorithm for Particle Simulations*, *Journal of Computational Physics* **135** (1997) 280–292.

[13] D. C. Marcello, *A very fast and angular momentum conserving tree code*, *Under Revision for Astrophysical Journal* (2017) .

[14] A. Kurganov and E. Tadmor, *New High-Resolution Central Schemes for Nonlinear Conservation Laws and Convection-Diffusion Equations*, *Journal of Computational Physics* **160** (2000) 241–282.

[15] P. Colella and P. R. Woodward, *The Piecewise Parabolic Method (PPM) for Gas-Dynamical Simulations*, *Journal of Computational Physics* **54** (1984) 174–201.

[16] C. Shu and S. Osher, *Efficient Implementation of Essentially Non-oscillatory Shock-Capturing Schemes*, *Journal of Computational Physics* **77** (1988) 439–+.

[17] I. Hachisu, *A versatile method for obtaining structures of rapidly rotating stars. II - Three-dimensional self-consistent field method*, *Astrophysical Journal Supplement* **62** (1986) 461–499.

[18] W. Even and J. E. Tohline, *Constructing Synchronously Rotating Double White Dwarf Binaries*, *Astrophysical Journal Supplement* **184** (2009) 248–263, [0908.2116].

[19] K. Kadam, P. M. Motl, J. Frank, G. C. Clayton and D. C. Marcello, *A numerical method for generating rapidly rotating bipolytropic structures in equilibrium*, *Monthly Notices of the Royal Astronomical Society* **462** (2016) 2237–2245.

[20] Kadam, K., *Numerical simulations of mass transfer in close and contact binaries using bipolytropes*. PhD thesis, Louisiana State University, 2017.

[21] T. R. Marsh, G. Nelemans and D. Steeghs, *Mass transfer between double white dwarfs*, *Monthly Notices of the Royal Astronomical Society* **350** (2004) 113–128, [arXiv:astro-ph/0312577].

[22] M. Kilic, J. J. Hermes, A. Gianninas, W. R. Brown, C. O. Heinke, M. A. Agüeros et al., *Found: the progenitors of AM CVn and supernovae .Ia*, *Monthly Notices of the Royal Astronomical Society* **438** (2014) L26–L30, [1310.6359].

[23] G. Nelemans, L. R. Yungelson, M. V. van der Sluys and C. A. Tout, *The chemical composition of donors in AM CVn stars and ultracompact X-ray binaries: observational tests of their formation*, *Monthly Notices of the Royal Astronomical Society* **401** (2010) 1347–1359, [0909.3376].

[24] H. Kaiser, T. Heller, A. Berge, B. Adelstein-Lelbach and et.al., *HPX V0.9.99: A general purpose C++ runtime system for parallel and distributed applications of any scale*, http://dx.doi.org/10.5281/zenodo.58027.