# Moving Boundaries in Common Envelope Evolution

Logan J. Prust*
*Department of Physics, University of Wisconsin-Milwaukee*
*Milwaukee, WI, USA*

## Abstract

The common envelope phase in binary star systems is simulated using the 3-D moving-mesh hydrodynamic code MANGA. The companion object is modeled as a moving boundary condition in our simulations. We outline the methodology of implementing moving boundary conditions into MANGA, which are reactive to hydrodynamic and gravitational forces. The generation of initial conditions for our simulations is discussed, which must be modified near the location of the companion. We show that the size of the companion has a significant effect on the dynamics of the spiral-in, particularly with regard to the orbital separation and eccentricity. We note that our results are sensitive to the spatial resolution of our simulations. We conclude that the finite size of the companion object must be taken into account in future simulations of common envelope evolution.

## 1. Introduction

Common envelope evolution (CEE) is a phase during the life of a binary star system in which a giant star shares its gaseous envelope with a smaller companion object: a small star or a stellar remnant (for a review see Ivanova et al. 2013). The transfer of orbital energy and angular momentum from the giant's core and companion to the gas drives partial or complete ejection of the envelope from the system. This, in turn, causes the separation between the core and companion to decrease. This process is important for the formation of X-ray binaries, double white dwarfs, and double neutron stars, and may also be responsible for the merging black hole systems observed by Advanced LIGO (Ivanova et al. 2013). Despite its importance, CEE is not well understood, and this is due in part to the number of different physics that must be included as well as the range of time-scales involved. Furthermore, there is uncertainty as to whether the contributions from recombination and radiation energy should be included, as well as whether the finite size of the companion is negligible. Because of these challenges, our current understanding of CEE is limited at best.

In the presence of the different physics and time-scales, astronomers have mainly employed conserved quantities – energy and angular momentum – to roughly model CEE. These are known as the energy formalism (Webbink 1984) and the angular momentum (or $\gamma$) formalism (Nelemans et al. 2000). Because of the uncertainties in theoretical modeling of CEE, we bring numerical tools to bear. To perform 3-D simulations of CEE, two main approaches are used. The first is smooth particle hydrodynamics (SPH), where fluid quantities are determined from a finite sampling of nearby particles. SPH is computationally inexpensive and obeys conservation laws well, but the smoothing means that it has difficulty in handling discontinuities (i.e. shock waves). The other approach is Eulerian grid-based solvers, which are superior at capturing shocks but suffer from grid effects and violation of conservation laws (Springel 2010). The defining numerical studies are those carried out by Passy et al. (2012), Ricker & Taam (2012) and Nandez et al. (2015). Nandez et al. (2015) use a SPH code, Passy et al. (2012) use both a SPH and an Eulerian code and Ricker & Taam (2012) use an adaptive mesh refinement Eulerian code.

*The author is supported supported by the NASA Astrophysical Theory Program (ATP) through NASA grant NNH17ZDA001N-ATP. We use the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation (NSF) grant No. ACI-1053575. We acknowledge the Texas Advanced Computing Center (TACC) at The University of Texas at Austin for providing HPC resources that have contributed to the research results reported within this paper (URL: http://www.tacc.utexas.edu). Computational resources supporting this work were also provided by the NASA High-End Computing (HEC) Program through the NASA Advanced Supercomputing (NAS) Division at the Ames Research Center. This material is based upon work supported by NASA under Award No. RFP20_6.1 issued through the Wisconsin Space Grant Consortium and the National Space Grant College and Fellowship Program. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Aeronautics and Space Administration. We thank Philip Chang for useful and detailed discussions.

In recent years, a hybrid of these methods has been developed in an attempt to capture the best characteristics of both. This is the arbitrary Lagrangian-Eulerian (ALE) scheme, and software that use ALE schemes are known as moving-mesh codes. In an ALE scheme, the mesh moves along with the fluid, combining the superior shock-capturing of grid-based solvers with the conservation properties of SPH. Springel (2010) described one such scheme that has proven successful, which is implemented in the code AREPO. The scheme constructs an unstructured mesh from an arbitrary distribution of points using a Voronoi tessellation. This guarantees that the mesh will be well-defined, unique and continuously deformable; thus, finite-volume methods can be applied in a manner similar to that of an Eulerian code. In addition, the lack of Galilean invariance in Eulerian codes is rectified if the mesh cells move along with the local flow. A moving-mesh hydrodynamic solver has also been developed for the N-body simulation code ChaNGa (Charm N-body GrAvity solver) (Jetley et al. 2008, 2010; Menon et al. 2015). This moving-mesh solver is known as MANGA, which is is described in detail by Chang et al. (2017). In Prust & Chang (2019, hereinafter PC19), we incorporated individual time-steps into MANGA and showed that this results in a speedup of a factor of 4 to 5 for a CEE simulation.

In fluid dynamics it is often desirable to set boundary conditions on a surface. For example, in modeling the flow around an airfoil the edges of the simulation box can be treated as inflow or outflow boundaries and the airfoil itself as a reflective boundary, which allows no matter to pass through it. Indeed, Wan & Zhong (2019) use a 2-D ALE scheme with moving boundaries to model a pitching NACA 0012 airfoil. Such schemes are common in fields such as aerodynamics, but also have interesting and useful applications in astrophysics. Here, we use a moving boundary to model the companion object in CEE simulations.

All CEE simulations to date have modeled the companion object as a dark matter particle, which interacts only gravitationally with the gas particles. While this is appropriate in the case of a black hole companion, it is unclear how well it approximates a larger object such as a neutron star, white dwarf, or small star. In particular, stars have hydrodynamically relevant surfaces. As material accretes onto a star or just passes by it, it can create a shock, and this shock can backreact on the material. It is impractical numerically to model the companion star as a hydrostatically supported object. However, because of the great density contrast between the companion star and the extended envelope of the primary, the surface of the companion can instead be treated as a hard spherical boundary. It is therefore desirable to model the companion as a moving boundary which interacts both gravitationally and hydrodynamically with the primary. To this end, we have implemented moving boundary conditions into MANGA which react to gravitational and hydrodynamic forces. We summarize the implementation here, but we refer the reader to Prust (2020, hereinafter P20) for a detailed discussion.

The implementation of boundary conditions into Eulerian grid-based solvers is straightforward. However, because Eulerian grid codes have a static mesh, it is at best very difficult to have a boundary which moves with respect to the mesh. On the other hand, moving-mesh codes are ideal for moving boundaries since the mesh-generating points can move with respect to one another. In ALE schemes, the velocities of the mesh-generating points are typically taken to be equal or similar to the fluid velocity to gain the advantages offered by a Lagrangian scheme. However, the mesh-generating points can in principle move at any velocity relative to the fluid.

Springel (2010) carried out a simple 2-D simulation using moving boundaries in AREPO in which a curved boundary (thought of as a "spoon") stirs a fluid. They generate their boundary using two sets of points on either side of and equidistant from the desired boundary surface (see their Fig. 38), on which they set reflective boundary conditions. They also require that both sets of points move together as a rigid body, and move their spoon at a constant speed along a pre-determined circular path. We adopt a similar strategy for our implementation of moving boundaries into MANGA, requiring the motion of the cells adjacent to the boundary to be identical to the motion of the boundary itself. However, we also set the velocity of the boundary such that its motion is reactive to external forces. The aerodynamic drag force can be computed by integrating the gas pressure over the surface of the boundary, and the gravitational force by integrating over the volume of the boundary.

We have organized this paper as follows. In section 2, we summarize the algorithm of MANGA and describe the initial conditions for our simulations. We then outline our implementation of moving boundary conditions into MANGA in section 3. In section 4, we describe the results of our simulations of a 2 $M_\odot$ red giant and a 1 $M_\odot$ companion. We conclude and discuss future directions in section 5.

## 2. Methodology

**2.1. Summary of the MANGA algorithm** The ALE algorithm that is implemented in MANGA is briefly summarized as follows. We refer the reader to Chang et al. (2017) and PC19 for detailed discussions. MANGA solves the Euler equations, which written in conservative form are

$$\frac{\partial \rho}{\partial t} + \boldsymbol{\nabla} \cdot \rho \boldsymbol{v} = 0, \tag{1}$$

$$\frac{\partial \rho \boldsymbol{v}}{\partial t} + \boldsymbol{\nabla} \cdot \rho \boldsymbol{v}\boldsymbol{v} + \boldsymbol{\nabla} P = -\rho \boldsymbol{\nabla} \Phi \tag{2}$$

and

$$\frac{\partial \rho e}{\partial t} + \boldsymbol{\nabla} \cdot (\rho e + P)\, \boldsymbol{v} = -\rho \boldsymbol{v} \cdot \boldsymbol{\nabla} \Phi, \tag{3}$$

where $\rho$ is the density, $\boldsymbol{v}$ is the fluid velocity, $\Phi$ is the gravitational potential, $e = \epsilon + v^2/2$ is the specific energy, $\epsilon$ is the internal energy and $P(\rho, \epsilon)$ is the pressure. Eq. 1 to 3 can be written in a compact form by introducing a state vector $\boldsymbol{\mathcal{U}} = (\rho, \rho\boldsymbol{v}, \rho e)$:

$$\frac{\partial \boldsymbol{\mathcal{U}}}{\partial t} + \boldsymbol{\nabla} \cdot \boldsymbol{\mathcal{F}} = \boldsymbol{\mathcal{S}}, \tag{4}$$

where $\boldsymbol{\mathcal{F}} = (\rho\boldsymbol{v}, \rho\boldsymbol{v}\boldsymbol{v} + P, (\rho e + P)\boldsymbol{v})$ is the flux function and $\boldsymbol{\mathcal{S}} = (0, -\rho\boldsymbol{\nabla}\Phi, -\rho\boldsymbol{v} \cdot \boldsymbol{\nabla}\Phi)$ is the source function. To solve Eq. 4, we adopt the same finite-volume strategy as Springel (2010). For each cell, the integral over the volume of the $i$th cell $V_i$ defines the charge of the $i$th cell to be

$$\boldsymbol{U}_i = \int_{V_i} \boldsymbol{\mathcal{U}} dV = \boldsymbol{\mathcal{U}}_i V_i. \tag{5}$$

We then use Gauss' theorem to convert the volume integral over the divergence of the flux in Eq. 4 to a surface integral

$$\int_{\partial V_i} \boldsymbol{\nabla} \cdot \boldsymbol{\mathcal{F}} dV = \int_{\partial V_i} \boldsymbol{\mathcal{F}} \cdot \hat{\boldsymbol{n}} dA, \tag{6}$$

where $\partial V_i$ is the boundary of the cell. We now take advantage of the fact that the volumes are Voronoi cells with a finite number of neighbours to define an integrated flux

$$\sum_{j \in \text{neighbors}} \boldsymbol{F}_{ij} A_{ij} = \int_{\partial V_i} \boldsymbol{\mathcal{F}} \cdot \hat{\boldsymbol{n}} dA, \tag{7}$$

where $\boldsymbol{F}_{ij}$ and $A_{ij}$ are the average flux and area of the common face between cells $i$ and $j$. The discrete time evolution of the charges in the system is given by

$$\boldsymbol{U}_i^{n+1} = \boldsymbol{U}_i^n + \Delta t \sum_j \hat{\boldsymbol{F}}_{ij} A_{ij} + \Delta t \boldsymbol{S}_i, \tag{8}$$

where $\hat{\boldsymbol{F}}_{ij}$ is an estimate of the half time-step flux between the initial $\boldsymbol{U}_i^n$ and final states $\boldsymbol{U}_i^{n+1}$ and $\boldsymbol{S}_i^{(n+1/2)} = \int_i \boldsymbol{\mathcal{S}} dV$ is the time-averaged integrated source function.

**2.2. Simulation setup** We now outline the development of appropriate initial conditions for CEE simulations in MANGA. The reader is referred to PC19 for a detailed discussion. We first use the open-source stellar evolution code MESA (Paxton et al. 2011, 2013, 2015, 2018, 2019) to evolve a 2 M$_\odot$ star with metallicity $Z = 0.02$ from the pre-main sequence to the red giant phase. We stop when the star reaches 52 R$_\odot$ with a helium core mass of 0.36 M$_\odot$. From the MESA output, we take the entropy and hydrogen fraction. For the core, we take the total mass at a density that is 50 times greater than the mean density of the red giant, giving a core mass $M_\text{c} = 0.379$ M$_\odot$. This corresponds to a core radius $R_\mathrm{c} = 1.99$ R$_\odot$, which we use as the core gravitational softening length $h$. For all particles present regardless of type, MANGA uses spline softening to soften the gravitational forces within radius $h$ of the particle. Because of the great difference in density between the helium core and the hydrogen envelope, we model the core as a dark matter particle with mass $M_\mathrm{c}$ and softening radius of $R_\mathrm{c}$. We then take the entropy profile and construct a star of mass $M - M_\mathrm{c}$, with an entropy profile which matches that of the original star and contains a dark matter particle core. This yields a radial profile of density, temperature and hydrogen fraction that can be mapped to a particle (mesh-generating point) profile.

We construct an appropriate particle mesh for the star from a pre-computed glass distribution of points embedded in a 3-D cube. We periodically replicate this glass distribution to produce a sufficient number of particles. We assume that each particle is of equal volume and rescale them to the appropriate mass based on the computed $M(r)$ from MESA. These particles are also endowed with the radially interpolated temperature and hydrogen fraction. The total number of particles representing the star is $1.8 \times 10^5$. Outside of the star, we include a low density atmosphere of $10^{-13}$ g cm$^{-3}$ with temperature $10^5$ K that extends out to the total box size of $3.5 \times 10^{14}$ cm (5000 R$_\odot$), with periodic boundary conditions at its edges. The total number of particles in the simulation box is $1.1 \times 10^6$. To lower the computational cost, we use a mesh refinement algorithm to decrease the number of gas particles in the atmosphere far from the star. We define a scale factor $\mathcal{S}(r) = (r/R_*)^n$ where $R_*$ is the radius of the star, $r$ is the spherical radius and $n$ is an adjustable parameter which we have set to $n = 2/3$ in this case. Starting with the same uniform glass distribution as for the star, the linear spacing between particles is increased by $\mathcal{S}$ and their mass is increased by $\mathcal{S}^3$, preserving the external density.

We place the 1 M$_\odot$ companion in an initially circular Keplerian orbit just outside of the red giant, at $a = 60$ R$_\odot$. Although this neglects the evolution of the binary prior to CEE, we compensate by altering the dynamics of the giant. We have implemented the corotation between the envelope rotation and orbital motion into our simulations following the scheme of MacLeod et al. (2018). Within the envelope, we assume rigid body rotation and initialize the velocity as

$$v_\phi = f_\mathrm{cr}\Omega_\mathrm{orb}R_\mathrm{cyl}, \tag{9}$$

but give the atmosphere a velocity

$$v_\phi = \frac{f_\mathrm{cr}\Omega_\mathrm{orb}R_*^2\sin^2(\theta)}{R_\mathrm{cyl}}. \tag{10}$$

Here, $f_\mathrm{cr}$ is an adjustable parameter which we have set to unity, $\phi$ is the azimuthal angle, $\theta$ is the polar angle, $\Omega_\mathrm{orb}$ is the orbital frequency of the red giant and companion, $R_*$ is the radius of the giant and $R_\mathrm{cyl}$ is the cylindrical radius from the rotation axis of the giant. We note that Eq. 9 and 10 ensure that the velocity is continuous at the surface of the giant.

## 3. Implementation of Moving Boundary Conditions

We now discuss our changes to the MANGA algorithm to implement moving boundary conditions. We refer the reader to P20 for a more detailed discussion.

**3.1. Flux Across a Reflective Boundary** Let us consider a face on which we would like to set a boundary condition and boost our state vectors and flux functions into the rest frame of the face. If we rotate the face such that its normal lies along the $x$-direction, as is done in MANGA, then we can write the flux normal to the face as

$$\boldsymbol{\mathcal{F}}_x = (\rho v_x,\ \rho v_x^2 + P,\ \rho v_x v_y,\ \rho v_x v_z,\ (\rho e + P)v_x). \tag{11}$$

Here we have expanded the momentum flux tensor $\rho\boldsymbol{vv}$ into its three components. The flux of mass, momentum and energy that pass through the face are determined by the values of the state vector $\boldsymbol{\mathcal{U}}_\mathrm{L}, \boldsymbol{\mathcal{U}}_\mathrm{R}$ and the flux function $\boldsymbol{\mathcal{F}}_{x,\mathrm{L}}$, $\boldsymbol{\mathcal{F}}_{x,\mathrm{R}}$ on either side of the face. We see that under the transformation $v_x \to -v_x$ all terms in $\boldsymbol{\mathcal{F}}_x$ change sign with the exception of the $x$-momentum flux $\rho v_x^2 + P$. Conversely, when this transformation is applied to the state vector $\boldsymbol{\mathcal{U}} = (\rho, \rho v_x, \rho v_y, \rho v_z, \rho e)$, the *only* term that changes sign is the $x$-momentum $\rho v_x$. Consider now the Harten-Lax-van

Leer (HLL) approximate Riemann solver (Einfeldt 1988), which computes the flux across the face as

$$\hat{\boldsymbol{F}}_{\mathrm{HLL}} = \frac{\alpha_+ \boldsymbol{\mathcal{F}}_{x,\mathrm{L}} + \alpha_- \boldsymbol{\mathcal{F}}_{x,\mathrm{R}} - \alpha_+ \alpha_- (\boldsymbol{\mathcal{U}}_{\mathrm{R}} - \boldsymbol{\mathcal{U}}_{L})}{\alpha_+ + \alpha_-}, \tag{12}$$

where $\alpha_\pm = \max[0, \pm\lambda_\pm(\boldsymbol{\mathcal{U}}_{\mathrm{L}}), \pm\lambda_\pm(\boldsymbol{\mathcal{U}}_{\mathrm{R}})]$, the maximum and minimum eigenvalues of the Jacobians of each state are $\lambda_\pm = v_x \pm c_{\mathrm{s}}$ and $c_{\mathrm{s}}$ is the sound speed. We now impose the condition that the state vector and flux on either side are equal except for the fluid velocities normal to the face, which are opposite: $v_x = v_{x,\mathrm{L}} = -v_{x,\mathrm{R}}$. That is,

$$\boldsymbol{\mathcal{F}}_{x,\mathrm{R}}(v_x) = \boldsymbol{\mathcal{F}}_{x,\mathrm{L}}(-v_x) \quad \text{and} \quad \boldsymbol{\mathcal{U}}_{\mathrm{R}}(v_x) = \boldsymbol{\mathcal{U}}_{\mathrm{L}}(-v_x). \tag{13}$$

This implies $\alpha_+ = \alpha_- = c_{\mathrm{s}} + |v_x|$ and simplifies the HLL flux to the form

$$\hat{\boldsymbol{F}}_{\mathrm{HLL}} = \frac{\boldsymbol{\mathcal{F}}_{x,\mathrm{L}}(v_x) + \boldsymbol{\mathcal{F}}_{x,\mathrm{L}}(-v_x) - \alpha_+[\boldsymbol{\mathcal{U}}_{\mathrm{L}}(-v_x) - \boldsymbol{\mathcal{U}}_{\mathrm{L}}(v_x)]}{2}. \tag{14}$$

If we now plug in $\boldsymbol{\mathcal{F}}$ and $\boldsymbol{\mathcal{U}}$ in component form as in (11), we obtain

$$\hat{\boldsymbol{F}}_{\mathrm{HLL}} = (0,\ \rho v_x^2 + P + \rho v_x(c_{\mathrm{s}} + |v_x|),\ 0,\ 0,\ 0). \tag{15}$$

We see that all terms in $\hat{\boldsymbol{F}}_{\mathrm{HLL}}$ vanish except for the term corresponding to the flux of $x$-momentum – that is, the component of momentum normal to the face. Thus, the reflective boundary condition (13) prevents the transport of any quantity besides momentum normal to the face for the HLL Riemann solver; mass, energy and entropy cannot be transported. This momentum flux corresponds to the contact pressure exerted on the face. The derivation for the Harten-Lax-van Leer-Contact (HLLC) solver (Toro et al. 1994) proceeds in a similar fashion. The implementation of reflective boundary conditions requires modifications to the Riemann solver and – for schemes that are accurate to second order or higher in space – the gradient estimation. The alternate condition $\boldsymbol{\mathcal{F}}_{x,\mathrm{L}}(v_x) = \boldsymbol{\mathcal{F}}_{x,\mathrm{R}}(v_x)$ and $\boldsymbol{\mathcal{U}}_{x,\mathrm{L}}(v_x) = \boldsymbol{\mathcal{U}}_{x,\mathrm{R}}(v_x)$ corresponds to an outflow boundary condition. However, we focus on reflective conditions in this paper for the reason that inflow or outflow conditions in a moving-mesh code would necessitate the creation or destruction of mesh-generating points, which is not yet implemented in MANGA.

**3.2. Initialization** In the initialization step at the beginning of the simulation, we declare some user-specified region as the initial boundary. Cells with centres within this region are defined to be a part of the boundary. We will refer to these cells as “boundary cells” and to all other cells as “gas cells.” The boundary cells will comprise the boundary for the entirety of the simulation, as boundary cells are never added or removed. We define the boundary particles to have equal mass so that the density of the boundary is constant, provided that the density of mesh-generating points is constant within the boundary. The total mass of the boundary is an input parameter in our simulations.

To accurately produce the desired boundary shape, it is necessary to have a sufficiently high density of mesh-generating points near the edges of the boundary. To this end, it is sometimes beneficial to initialize the simulation with a larger resolution in the region containing the boundary than in the rest of the simulation box. In our CEE simulations, we refine the mesh in a spherical region centred on and enclosing the boundary, with a radius equal to twice that of the boundary. We begin by specifying the number of cells to comprise the boundary (500 in all simulations presented here). We then require that the number density of mesh-generating points immediately outside of the boundary matches that inside. Then, the number density decreases with distance from the boundary in the same manner as the atmosphere particles decrease with distance from the star, as described in section 2.2. This local increase in resolution results in the addition of roughly 2000 mesh cells, which is small compared to the $10^6$ particles in the simulation as a whole. This method is a step forward from P20, where we simply used a finer mesh in a cube centred on the companion, with no variations in resolution within the cube.

**3.3. Edge Cells** We define all gas cells which share a face with a boundary cell as “edge cells.” These cells function as normal gas cells for the most part; however, the motion of their mesh-generating points always matches that of the boundary. The result of this is a layer of gas cells which do not move with respect to the boundary, as in Springel (2010). This is important because the surface of the boundary is comprised of the shared faces between the edge cells and boundary cells, which we refer to as “boundary faces.” The boundary faces are redrawn with every Voronoi computation, which happens several times during each time-step. Therefore, if the mesh-generating points were allowed to flow past the boundary, the surface of the boundary would deform, which is undesirable. Although the edge cells are the only gas cells which have direct contact with the boundary, they communicate its effects to the rest of the gas through their properties and motion.

**3.4. Time-Stepping** In PC19, we discuss the implementation of individual time-steps into MANGA. PC19 finds that individual time-stepping decreases the computation time by a factor of 4 to 5 for a CEE simulation. When a moving boundary is present, we place the boundary and edge cells on the highest rung (smallest time-step) of all cells present. Besides improving the stability of the code, this mitigates the risk of gas cells penetrating into the boundary or other pathological behaviors. Because the number of edge cells is typically small relative to the total number of cells, we find that this does not substantially slow down the simulation code.

**3.5. Motion of Boundary** The motion of our boundaries is reactive to both hydrodynamic and gravitational forces. The hydrodynamic force is calculated during the half time-step Riemann solution, and is the sum of all forces on the boundary faces. We combine this with the gravitational acceleration to get the total acceleration of the boundary. At the drift step, this acceleration is used to update the velocities of all boundary and edge cells. Unlike normal gas cells, the boundary and edge cells do not receive kicks in the direction of the fluid velocity.

**3.6. Test Cases** Our moving boundary module has been successfully applied to several test cases (P20), including a Sod shock tube, a Sedov-Taylor blast wave, and a sphere moving at supersonic speed through air. We have also carried out a simple CEE simulation where the companion object is modeled as a spherical moving boundary, using the same initial conditions as PC19. Though low resolution, this simulation qualitatively shows the expected behavior of a common envelope phase, demonstrating that our module is feasible for studying CEE.

## 4. Results

We carry out three CEE simulations using the above initial conditions, modeling the companion as a spherical moving boundary with radius 1 $R_\odot$, 2 $R_\odot$ and 3 $R_\odot$. For the 2 $R_\odot$ case, we show density projections at $t$ = 1, 10, 20, 30, 50 and 100 days in Fig. 1.

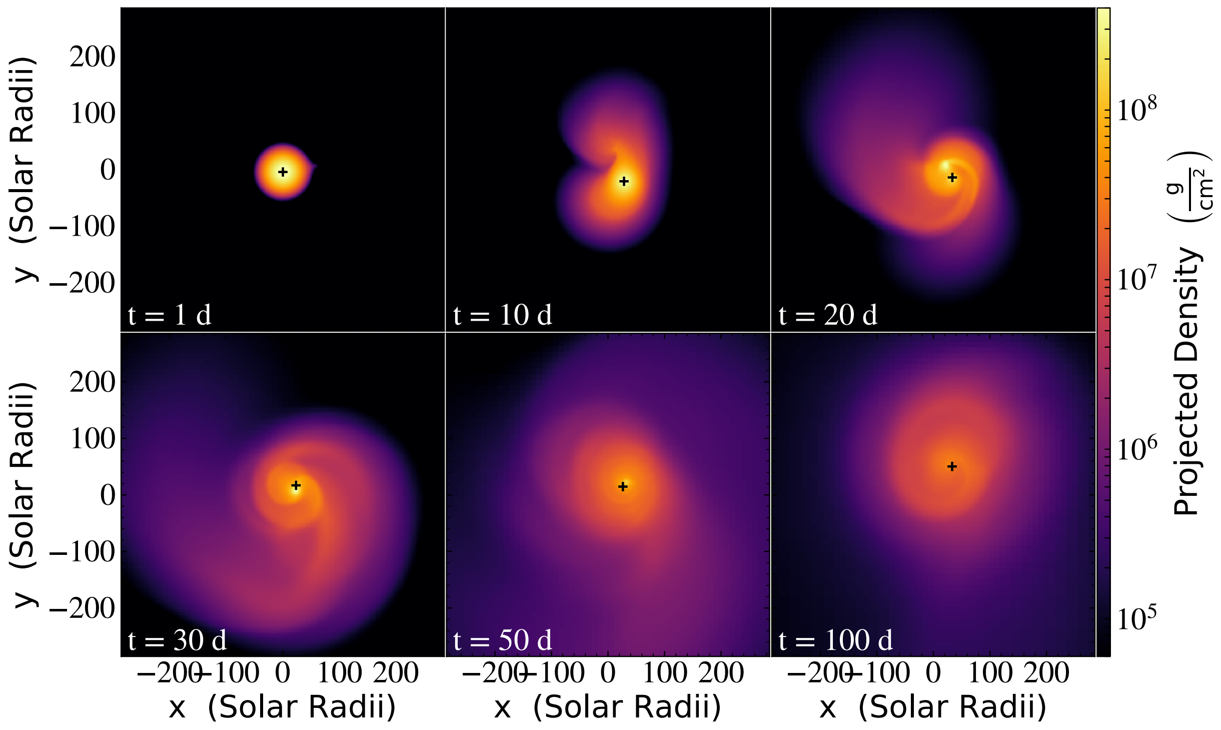

Figure 1: Density projections in our simulation with a companion of radius 2 $R_\odot$. The + sign marks the red giant core.

In all cases, the companion experiences an initial plunge into the envelope of the giant that lasts about 18 days, throwing off a large tidal tail and greatly decreasing the separation between the companion and core. The companions then continue to spiral in as their orbital energy is transferred to the gas; the spiral shocks facilitating this transfer can be seen in the projections (Fig. 1 upper right and lower left). The simulations end well before the outflows reach the edge of the simulation box.

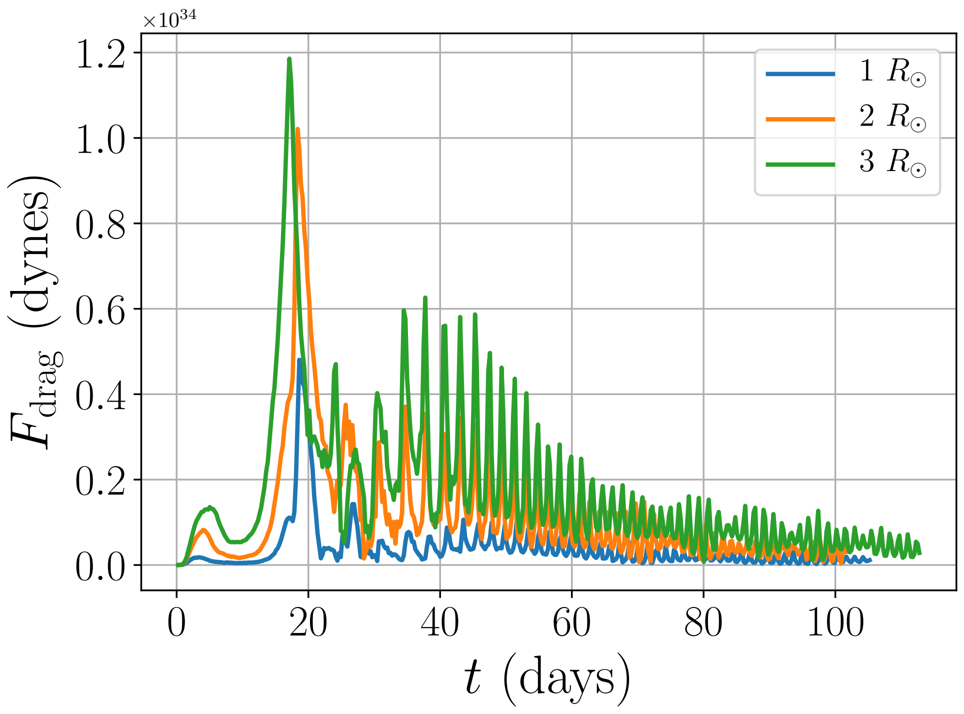

Figure 2: Aerodynamic drag experienced by the companion object in each case. The spikes occur at periapse passage, when the orbital velocity is at a maximum.

In Fig. 2, we have plotted the aerodynamic drag experienced by the companion object in each case as a function of time. Many spikes are seen in the drag force, corresponding to the times of periapse passage. Unsurprisingly, the size of the companion has a dramatic effect on the drag force. These results agree with order of magnitude estimates of the ram pressure on a spherical object moving through a gas ($10^{33}$ dynes).

The separations between the stellar core and companion are shown in Fig. 3. For ease of reading, we show the separations on a logarithmic scale and smooth them over a period of 15 d in Fig. 4. That is, each separation $a_i$ is taken to be the average of all separations within an interval of 15 d centered on $a_i$. After one orbit, all binaries have reduced their orbital separation to less than half the initial separation, and they continue to spiral in at a slower rate. We see that these separations vary between the three cases, especially at late times. In particular, the 3 R$_\odot$ case ends with a larger orbital separation than the other two. This is likely due to the larger aerodynamic drag in this case; this slows the companion, preventing it from ejecting the stellar envelope efficiently. Conversely, we would expect the 1 R$_\odot$ case to have the smallest orbital separation. This is not borne out by the results, but it appears that this and the 2 R$_\odot$ cases had not settled into stable orbits by the end of the simulation period.

The eccentricity $e$ (Fig. 5) can be found from the periapsis and apoapsis distances,

$$e = \frac{r_{\rm apo} - r_{\rm peri}}{r_{\rm apo} + r_{\rm peri}}, \tag{16}$$

where $r_{\rm apo}$ and $r_{\rm peri}$ denote the distances of closest and furthest approach. Here $r_{\rm apo}$ and $r_{\rm peri}$ are determined by interpolating the maxima and minima of the separation. The alert reader will notice that this definition is meaningful in this situation because the potential is not Keplerian owing to the distribution and nonaxisymmetry of the unbound gas. However, this method allows the determination of the eccentricity from only the orbital separations. All binaries circularize their orbits over time, seeing drops in eccentricity from $e \approx 0.5$ down to $e = 0.2 - 0.3$ by the end of the simulation period. Significant differences are seen between the three cases, though not in a consistent manner. So although it is not possible to determine the precise effect of companion size on eccentricity, it is clear that such an

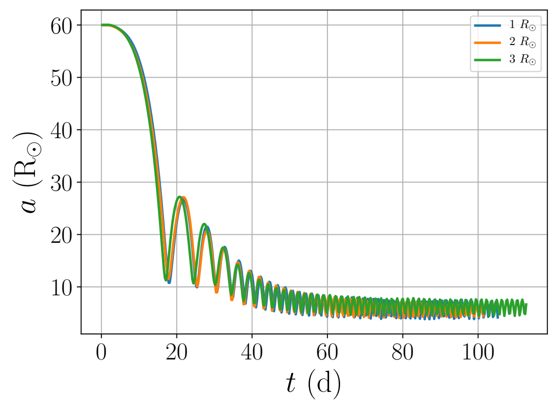

Figure 3: Separation $a$ between the stellar core and companion in each case. Significant differences are observed between the 3 cases.

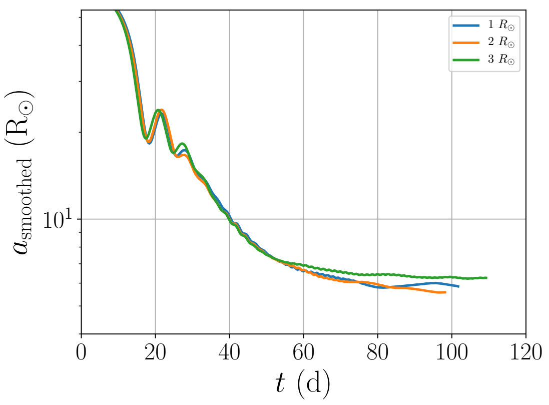

Figure 4: Separation $a$ between the stellar core and companion in each case, smoothed over time and shown on a logarithmic scale for ease of reading. Here we can see that the simulation with the largest companion size settles into an orbit with the largest final separation.

effect exists and is significant.

The simulations presented here were run with a resolution low enough that all simulations could be completed over the course of one summer. However, as it is possible that our results are sensitive to resolution, we have conducted a resolution test using the 2 R$_\odot$ case. Here we have doubled the spatial resolution, using $3.5 \times 10^5$ mesh cells for the star and $1.9 \times 10^6$ for the atmosphere, for a total of $2.3 \times 10^6$ cells. The orbital separation for both resolutions is shown in Fig. 6. Although the separations are similar at early times, we see a marked difference after about 30 days. This indicates that resolution is indeed a factor in our results, and in the future we will run higher-resolution simulations of all three cases to obtain more accurate results.

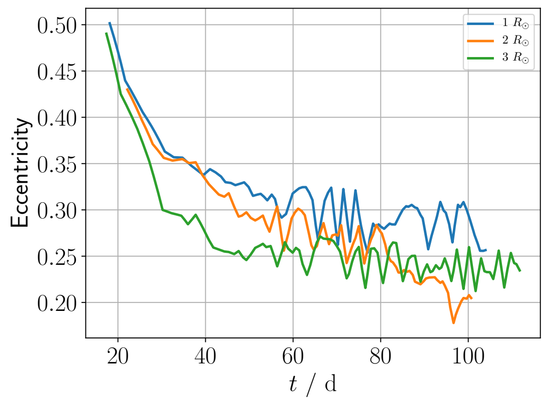

Figure 5: Orbital eccentricity in each case, as determined by 16. Significant differences are seen among the three cases, though not with a clear trend.

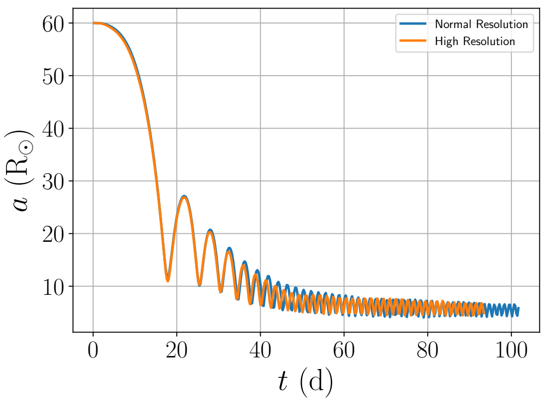

Figure 6: Separation $a$ between the stellar core and companion in the resolution test of the 2 R$_\odot$ case. Although similar at early times, differences begin to emerge after roughly 30 days.

## 5. Discussion

In this work, we investigate the effects of companion size on CEE using the moving-mesh code MANGA. We use the moving boundary conditions module for MANGA, described in P20, to model the companion object as a spherical moving boundary. We generate initial conditions for our simulations by using MESA to evolve a pre-main sequence star to the red giant phase before placing a companion on its surface in a circular orbit. We use a newly-refined algorithm to refine the mesh near the initial position of the boundary. We run simulations of a 2 M$_\odot$ giant with a 1 M$_\odot$

companion for 100 days for three values of the companion radius: 1, 2, and 3 $R_{\odot}$.

We find that the aerodynamic drag on the companion depends strongly on its radius, as predicted by theory. This has ramifications for the dynamics of the spiral-in: both the orbital separation and eccentricity differ markedly among the three cases. Although we have not found an explicit relationship between companion size and the orbital properties, the results clearly show that the companion radius has a large effect. This indicates that future numerical studies of CEE will be unable to ignore the finite size of the companion in the case that the companion is a star. We have also found that our results are sensitive to resolution, indicating the need for future studies using a finer mesh and a longer integration time.

We note here that other processes may also alter the orbital dynamics. Much recent work have been focused on energy and momentum injection from jets produced by accretion onto the companion (for example, López-Cámara et al. 2018). In addition, the transition from laminar flow to accretion flows around the companion star can be complicated and also provide another mechanism by which orbital energy is dissipated (MacLeod et al. 2017). Finally, we mention that it is also suggested that dust formation during CEE can drive winds to expel the envelope (Glanz & Perets 2018).

## Acknowledgments

We thank Philip Chang for useful and detailed discussions. We are supported by the NASA Astrophysical Theory Program (ATP) through NASA grant NNH17ZDA001N-ATP. We use the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation (NSF) grant No. ACI-1053575. We acknowledge the Texas Advanced Computing Center (TACC) at The University of Texas at Austin for providing HPC resources that have contributed to the research results reported within this paper (URL: http://www.tacc.utexas.edu). We use the YT software platform for the analysis of the data and generation of plots in this work (Turk et al. 2011). Computational resources supporting this work were also provided by the NASA High-End Computing (HEC) Program through the NASA Advanced Supercomputing (NAS) Division at the Ames Research Center. This material is based upon work supported by NASA under Award No. RFP20_6.1 issued through the Wisconsin Space Grant Consortium and the National Space Grant College and Fellowship Program. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Aeronautics and Space Administration.

## References

Chang, P; Wadsley, J; Quinn, T. "A Moving Mesh Hydrodynamics Solver for ChaNGa," *accepted to MNRAS*, 2017

Einfeldt, B. "On Godunov-Type Methods for Gas Dynamics," *SIAM Journal on Numerical Analysis*, v. 25(2), 1988, p. 294–318. https://ui.adsabs.harvard.edu/abs/1988SJNA...25..294E

Glanz, H; Perets, HB. "Efficient common-envelope ejection through dust-driven winds," *MNRAS*, v. 478, 2018, p. L12–L17. https://ui.adsabs.harvard.edu/abs/2018MNRAS.478L..12G

Ivanova, N; Justham, S; Chen, X; De Marco, O; Fryer, CL; Gaburov, E; Ge, H; Glebbeek, E; Han, Z; Li, XD; Lu, G; Marsh, T; Podsiadlowski, P; Potter, A; Soker, N; Taam, R; Tauris, TM; van den Heuvel, EPJ; Webbink, RF. "Common envelope evolution: where we stand and how we can move forward," *A&A Rev.*, v. 21, 2013, p. 59. http://ukads.nottingham.ac.uk/abs/2013A%26ARv..21...59I

Jetley, P; Gioachin, F; Mendes, C; Kale, LV; Quinn, TR. ""massively parallel cosmological simulations with changa"," *Proceedings of IEEE International Parallel and Distributed Processing Symposium*, 2008

Jetley, P; Wesolowski, F; Gioachin, F; Kale, LV; Quinn, TR. ""scaling hierarchical n-body simulations on gpu clusters"," *Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing*, 2010

López-Cámara, D; De Colle, F; Moreno Méndez, E. "Self-regulating jets during the Common Envelope phase," preprint (`arXiv:1806.11115`), 2018, p. arXiv:1806.11115. https://ui.adsabs.harvard.edu/abs/2018arXiv180611115L

MacLeod, M; Antoni, A; Murguia-Berthier, A; Macias, P; Ramirez-Ruiz, E. "Common Envelope Wind Tunnel: Coefficients of Drag and Accretion in a Simplified Context for Studying Flows around Objects Embedded within Stellar Envelopes," *ApJ*, v. 838, 2017, p. 56. https://ui.adsabs.harvard.edu/abs/2017ApJ...838...56M

MacLeod, M; Ostriker, EC; Stone, JM. "Runaway Coalescence at the Onset of Common Envelope Episodes," *ApJ*, v. 863, 2018, p. 5. https://ui.adsabs.harvard.edu/#abs/2018ApJ...863....5M

Menon, H; Wesolowski, L; Zheng, G; Jetley, P; Kale, L; Quinn, T; Governato, F. "Adaptive techniques for clustered N-body cosmological simulations," *Computational Astrophysics and Cosmology*, v. 2, 2015, p. 1. http://adsabs.harvard.edu/abs/2015ComAC...2....1M

Nandez, JLA; Ivanova, N; Lombardi, JC. "Recombination energy in double white dwarf formation," *MNRAS*, v. 450, 2015, p. L39–L43. http://adsabs.harvard.edu/abs/2015MNRAS.450L..39N

Nelemans, G; Verbunt, F; Yungelson, LR; Portegies Zwart, SF. "Reconstructing the evolution of double helium white dwarfs: envelope loss without spiral-in," *A&A*, v. 360, 2000, p. 1011–1018. http://adsabs.harvard.edu/abs/2000A%26A...360.1011N

Passy, JC; De Marco, O; Fryer, CL; Herwig, F; Diehl, S; Oishi, JS; Mac Low, MM; Bryan, GL; Rockefeller, G. "Simulating the Common Envelope Phase of a Red Giant Using Smoothed-particle Hydrodynamics and Uniform-grid Codes," *ApJ*, v. 744, 2012, p. 52. http://adsabs.harvard.edu/abs/2012ApJ...744...52P

Paxton, B; Bildsten, L; Dotter, A; Herwig, F; Lesaffre, P; Timmes, F. "Modules for Experiments in Stellar Astrophysics (MESA)," *ApJS*, v. 192, 2011, p. 3. http://adsabs.harvard.edu/abs/2011ApJS..192....3P

Paxton, B; Cantiello, M; Arras, P; Bildsten, L; Brown, EF; Dotter, A; Mankovich, C; Montgomery, MH; Stello, D; Timmes, FX; Townsend, R. "Modules for Experiments in Stellar Astrophysics (MESA): Planets, Oscillations, Rotation, and Massive Stars," *ApJS*, v. 208, 2013, p. 4. http://adsabs.harvard.edu/abs/2013ApJS..208....4P

Paxton, B; Marchant, P; Schwab, J; Bauer, EB; Bildsten, L; Cantiello, M; Dessart, L; Farmer, R; Hu, H; Langer, N; Townsend, RHD; Townsley, DM; Timmes, FX. "Modules for Experiments in Stellar Astrophysics (MESA): Binaries, Pulsations, and Explosions," *ApJS*, v. 220, 2015, p. 15. http://adsabs.harvard.edu/abs/2015ApJS..220...15P

Paxton, B; Schwab, J; Bauer, EB; Bildsten, L; Blinnikov, S; Duffell, P; Farmer, R; Goldberg, JA; Marchant, P; Sorokina, E; Thoul, A; Townsend, RHD; Timmes, FX. "Modules for Experiments in Stellar Astrophysics (MESA): Convective Boundaries, Element Diffusion, and Massive Star Explosions," *ApJS*, v. 234(2), 2018, p. 34. https://ui.adsabs.harvard.edu/abs/2018ApJS..234...34P

Paxton, B; Smolec, R; Schwab, J; Gautschy, A; Bildsten, L; Cantiello, M; Dotter, A; Farmer, R; Goldberg, JA; Jermyn, AS; Kanbur, SM; Marchant, P; Thoul, A; Townsend, RHD; Wolf, WM; Zhang, M; Timmes, FX. "Modules for Experiments in Stellar Astrophysics (MESA): Pulsating Variable Stars, Rotation, Convective Boundaries, and Energy Conservation," *ApJS*, v. 243(1), 2019, p. 10. https://ui.adsabs.harvard.edu/abs/2019ApJS..243...10P

Prust, LJ. "Moving and reactive boundary conditions in moving-mesh hydrodynamics," *MNRAS*, v. 494(4), 2020, p. 4616–4626. https://ui.adsabs.harvard.edu/abs/2020MNRAS.494.4616P

Prust, LJ; Chang, P. "Common envelope evolution on a moving mesh," *MNRAS*, v. 486(4), 2019, p. 5809–5818. https://ui.adsabs.harvard.edu/abs/2019MNRAS.486.5809P

Ricker, PM; Taam, RE. "An AMR Study of the Common-envelope Phase of Binary Evolution," *ApJ*, v. 746, 2012, p. 74. http://adsabs.harvard.edu/abs/2012ApJ...746...74R

Springel, V. "E pur si muove: Galilean-invariant cosmological hydrodynamical simulations on a moving mesh," *MNRAS*, v. 401, 2010, p. 791–851. http://adsabs.harvard.edu/abs/2010MNRAS.401..791S

Toro, EF; Spruce, M; Speares, W. "Restoration of the contact surface in the HLL-Riemann solver," *Shock Waves*, v. 4(1), 1994, p. 25–34. https://ui.adsabs.harvard.edu/abs/1994ShWav...4...25T

Turk, MJ; Smith, BD; Oishi, JS; Skory, S; Skillman, SW; Abel, T; Norman, ML. "yt: A Multi-code Analysis Toolkit for Astrophysical Simulation Data," *The Astrophysical Journal Supplement Series*, v. 192, 2011, p. 9. http://adsabs.harvard.edu/abs/2011ApJS..192....9T

Wan, Y; Zhong, C. "An ALE-type discrete unified gas kinetic scheme for low-speed continuum and rarefied flow simulations with moving boundaries," *arXiv e-prints*, 2019, p. arXiv:1906.01813. https://ui.adsabs.harvard.edu/abs/2019arXiv190601813W

Webbink, RF. "Double white dwarfs as progenitors of R Coronae Borealis stars and Type I supernovae," *ApJ*, v. 277, 1984, p. 355–360. http://adsabs.harvard.edu/abs/1984ApJ...277..355W