



The Santa Barbara Binary–disk Code Comparison

Paul C. Duffell¹ , Alexander J. Dittmann^{2,3} , Daniel J. D’Orazio⁴ , Alessia Franchini^{5,6,7} , Kaitlin M. Kratter⁸ , Anna B. T. Penzlin^{9,10} , Enrico Ragusa^{11,12} , Magdalena Siwek¹³ , Christopher Tiede⁴ , Haiyang Wang^{14,15,16} , Jonathan Zrake¹⁷ , Adam M. Dempsey¹⁸ , Zoltan Haiman¹⁹ , Alessandro Lupi^{6,7,20} , Michal Pirog^{21,22,23} , and Geoffrey Ryan²⁴

¹ Department of Physics and Astronomy, Purdue University, 525 Northwestern Avenue, West Lafayette, IN 47907, USA; pduffell@purdue.edu

² Department of Astronomy and Joint Space-Science Institute, University of Maryland, College Park, MD 20742-2421, USA; dittmann@umd.edu

³ Theoretical Division, Los Alamos National Laboratory, Los Alamos, NM 87545, USA

⁴ Niels Bohr International Academy, Niels Bohr Institute, Blegdamsvej 17, 2100 Copenhagen, Denmark; daniel.dorazio@nbi.ku.dk

⁵ Universität Zürich, Institut für Astrophysik, Winterthurerstrasse 190, CH-8057 Zürich, Switzerland

⁶ Dipartimento di Fisica “G. Occhialini,” Università degli Studi di Milano-Bicocca, Piazza della Scienza 3, I-20126 Milan, Italy

⁷ INFN, Sezione di Milano-Bicocca, Piazza della Scienza 3, I-20126 Milano, Italy

⁸ University of Arizona, 933 N. Cherry Avenue, Tucson, AZ 85721, USA

⁹ Institut für Astronomie und Astrophysik, Universität Tübingen, Auf der Morgenstelle 10, D-72076 Tübingen, Germany

¹⁰ Astrophysics Group, Department of Physics, Imperial College London, Prince Consort Road, London, SW7 2AZ, UK

¹¹ Dipartimento di Matematica, Università degli Studi di Milano, Via Salдинi 50, 20133, Milano, Italy

¹² ENS de Lyon, CRAL UMR5574, Université Claude Bernard Lyon 1, CNRS, Lyon, F-69007, France

¹³ Center for Astrophysics, Harvard University, Cambridge, MA 02138, USA

¹⁴ Fudan University, Department of Physics, Shanghai 200433, People’s Republic of China

¹⁵ Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Wilberforce Road, Cambridge CB3 0WA, UK

¹⁶ TAPIR, California Institute of Technology, Pasadena, CA 91125, USA

¹⁷ Department of Physics and Astronomy, Clemson University, Clemson, SC 29634, USA

¹⁸ X Computational Physics Division, Los Alamos National Laboratory, Los Alamos, NM 87545, USA

¹⁹ Columbia University, Departments of Astronomy and Physics, 550 West 120th Street, New York, NY 10027, USA

²⁰ Dipartimento di Scienza e Alta Tecnologia, Università degli Studi dell’Insubria, via Valleggio 11, I-22100 Como, Italy

²¹ Wilkes Honors College, Florida Atlantic University, Jupiter, FL 33458, USA

²² Department of Physics and Astronomy, West Virginia University, Morgantown, WV 26506, USA

²³ Center for Gravitational Waves and Cosmology, West Virginia University, Chestnut Ridge Research Building, Morgantown, WV 26505, USA

²⁴ Perimeter Institute for Theoretical Physics, 31 Caroline Street North, Waterloo, ON N2L 2Y5, Canada

Received 2024 February 17; revised 2024 June 18; accepted 2024 June 18; published 2024 July 29

Abstract

We have performed numerical calculations of a binary interacting with a gas disk, using 11 different numerical methods and a standard binary–disk setup. The goal of this study is to determine whether all codes agree on a numerically converged solution and to determine the necessary resolution for convergence and the number of binary orbits that must be computed to reach an agreed-upon relaxed state of the binary–disk system. We find that all codes can agree on a converged solution (depending on the diagnostic being measured). The zone spacing required for most codes to reach a converged measurement of the torques applied to the binary by the disk is roughly 1% of the binary separation in the vicinity of the binary components. For our disk model to reach a relaxed state, codes must be run for at least 200 binary orbits, corresponding to about a viscous time for our parameters, $0.2(a^2\Omega_B/\nu)$ binary orbits, where ν is the kinematic viscosity. The largest discrepancies between codes resulted from the dimensionality of the setup (3D vs. 2D disks). We find good agreement in the total torque on the binary between codes, although the partition of this torque between the gravitational torque, orbital accretion torque, and spin accretion torque depends sensitively on the sink prescriptions employed. In agreement with previous studies, we find a modest difference in torques and accretion variability between 2D and 3D disk models. We find cavity precession rates to be appreciably faster in 3D than in 2D.

Unified Astronomy Thesaurus concepts: Circumstellar disks (235); Binary stars (154); Planetary-disk interactions (2204); Supermassive black holes (1663); Accretion (14); Galaxy accretion disks (562); Hydrodynamics (1963); Computational methods (1965)

1. Introduction

Over the past few decades, the general astrophysical hydrodynamics problem of a disk interacting with a binary has been investigated in numerous studies. It has been studied in the context of a stellar binary interacting with the protostellar disk (e.g., Bate et al. 1995; Dunhill et al. 2015; Terquem et al.

2015; Franchini et al. 2019; Martin & Lubow 2019) and in the context of supermassive black hole binaries at subparsec separations interacting in galactic nuclei (e.g., Armitage & Natarajan 2002; Hayasaki et al. 2007; MacFadyen & Milosavljević 2008; Cuadra et al. 2009; Dittmann et al. 2023; Major Krauth et al. 2023). It has been studied in 2D, in the plane of the binary orbit (e.g., Ochi et al. 2005; MacFadyen & Milosavljević 2008; Hanawa et al. 2010; D’Orazio et al. 2013, 2016; Farris et al. 2014; Young et al. 2015; Muñoz & Lai 2016; Miranda et al. 2017; Muñoz et al. 2019; Duffell et al. 2020; Dittmann & Ryan 2021, 2022; Penzlin et al. 2022; Siwek et al. 2023a, 2023b; Mahesh et al. 2023; Cimerman & Rafikov 2024;



Original content from this work may be used under the terms of the [Creative Commons Attribution 4.0 licence](https://creativecommons.org/licenses/by/4.0/). Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

Tiede & D’Orazio (2024), and in 3D (e.g., Artymowicz & Lubow 1994, 1996; Bate & Bonnell 1997; Hayasaki et al. 2007; Kratter 2011; Nixon et al. 2011; Ragusa et al. 2016; Moody et al. 2019; Heath & Nixon 2020; Ragusa et al. 2020; Franchini et al. 2022; Bourne et al. 2023; Franchini et al. 2023; Park et al. 2024), including magnetic fields (e.g., Shi et al. 2012; Bankert et al. 2015; Shi & Krolik 2015), self-gravity (e.g., Cuadra et al. 2009; Roedig et al. 2011, 2012; Franchini et al. 2021; Bourne et al. 2023), and general relativity (e.g., Farris et al. 2012; Noble et al. 2012; Gold et al. 2014; Bowen et al. 2019; Noble et al. 2021; Combi et al. 2022; Avara et al. 2023; Mignon-Risse et al. 2023). It has been studied using a locally isothermal equation of state by many of these studies and using a cooled disk with more involved thermodynamic assumptions in others (e.g., Farris et al. 2015; d’Ascoli et al. 2018; Combi et al. 2022; Westernacher-Schneider et al. 2022; Avara et al. 2023; Major Krauth et al. 2023). What all of these studies have in common is that they employ numerical codes to integrate a set of hydrodynamic equations, to compute the gravitational interaction of a disk with a binary.

Many of these studies have had results that (at least seemingly) disagreed with one another; for example, it was often thought that disk–binary interaction would always remove angular momentum from the binary (essentially exerting a “drag force”), shrinking the binary separation (e.g., D’Angelo & Lubow 2008). However, with the advent of modern high-resolution hydrodynamical simulations, several studies have found net *positive* torque on the binary, enough to cause binary expansion. Circumbinary disk-driven outspirals were first observed in Miranda et al. (2017) and later confirmed by numerous other studies (e.g., Moody et al. 2019; Muñoz et al. 2019; Duffell et al. 2020). Subsequent studies have found that binary orbital evolution depends on the disk and binary parameters, such as the disk aspect ratio h/r (Tiede et al. 2020; Dittmann & Ryan 2022, 2024; Penzlin et al. 2022), the disk viscosity (Miranda et al. 2017; Dittmann & Ryan 2022, 2024; Franchini et al. 2022; Penzlin et al. 2022), the cooling timescale (Sudarshan et al. 2022; Wang et al. 2023a, 2023b; Franchini et al. 2024), and the binary eccentricity (Miranda et al. 2017; D’Orazio & Duffell 2021; Siwek et al. 2023a). Lai & Muñoz (2023) recently presented a review summarizing the outcomes of many of these studies.

This rapid influx of results is compounded by the fact that each study uses a different code, at different resolutions, integrating for different amounts of time, using different numerical choices such as boundaries, floors, and sinks, and including different amounts of additional physics (such as 3D and magnetic fields). In order to help make sense of all of this, we present a code test for a uniform binary–disk setup, to determine just how closely our different codes agree and what numerical ingredients (such as resolution and length of time integration) are necessary for convergence (or at least for agreement) between codes.

Nearly identical initial conditions are implemented across 11 different numerical schemes, running them at different resolutions and with different numerical choices. We compute a few different measurements of the gravitational torque on the binary, the morphology of the cavity, and the accretion rate onto each binary component, in order to determine under what conditions the codes agree for different measured quantities.

The hydrodynamical setup is designed to be relatively simple while maintaining the essential ingredients of a time-dependent

binary potential interacting with a disk. In this way we will think of it as the “minimal” binary–disk problem. The obvious advantage of using a minimal setup is that it can be tested across many different codes. Additionally, performing these calculations in 2D enables codes to be run at higher resolution and for longer timescales than in 3D, which will make it possible to determine when the solution is converged. Such a criterion might then be used to inform whether 3D studies are converged. Another advantage of such a minimal problem is that it provides a straightforward way to test new codes on their ability to describe a binary–disk setup. Our code outputs have been made public on Zenodo at doi:10.5281/zenodo.10557951 (Duffell 2024) and online.²⁵

2. Codes

A wide variety of codes will be compared on this binary setup. It is worth noting that many of them are not being utilized up to their full potential; many of these codes are capable of 3D, MHD, self-gravity, general relativity, complex thermodynamics, and more. Many of these capabilities need to be “turned off” in order to make an accurate comparison in a minimal binary–disk problem. For each code, results from at least three numerical calculations at different resolutions will be presented and compared through the usage of a number of diagnostics described in Section 4.

We list the essential characteristics of each code, as they are used in this study, in Table 1. Some codes, namely *Disco* and *Athena++*, have been used in multiple configurations. For the sake of brevity, we defer comprehensive descriptions of each code to the Appendix.

3. Test Problem

3.1. Binary Orbit

We study an equal-mass binary with total binary mass M (individual component masses $M_1 = M_2 = M/2$), on a fixed, circular ($e = 0$) Keplerian orbit with semimajor axis a . The binary angular frequency is denoted Ω_B , while the disk angular frequency is Ω . For concreteness, the binary is initialized along the x -axis, orbiting in the x - y plane counterclockwise when viewed from the positive z -axis.

Gravitational softening is implemented uniformly across all codes. The binary potential can be expressed as

$$\Phi_j = -\frac{GM_j}{\sqrt{|\mathbf{z}_{ij}|^2 + \epsilon^2}}, \quad (1)$$

where $|\mathbf{z}_{ij}|$ is the distance between the i th gas cell/particle and the j th binary component and ϵ is the gravitational softening length. In the fiducial runs, we set $\epsilon = 0.05a$, but we also explore its impact on the results in Section 6.5. In that section, we demonstrate the binary torque to be insensitive to the gravitational softening for $\epsilon < 0.15a$.

Most of the codes have employed “code units” where $GM = 1$, $a = 1$ (so that $\Omega_B = 1$), and $\Sigma_0 = 1$ (Σ_0 is defined in the initial conditions). Codes that do not work in these units will normalize their results (e.g., reporting Σ/Σ_0 instead of Σ). For example, simulations employing a “live binary” have an additional mass scale (the solution can depend on the normalization of Σ_0), and therefore such codes must set

²⁵ <https://www.physics.purdue.edu/duffell/data.html>

Table 1
Overview of Various Hydrodynamics Codes and Their Configurations Used in This Study

Code	2D/3D	Numerical Scheme	Convergence Order	Excised Binary	Discretization
Arepo ^a	2D	Moving-mesh finite volume	Second	No	Voronoi tessellation
Athena++ ^b	2D	Finite volume	Third	No	Cartesian
Athena++ (Excised)	2D	Finite volume	Second	Yes	Cylindrical polar
Athena++ (3D)	3D	Finite volume	Second	No	Cartesian
Disco ^c	2D	Moving-mesh finite volume	Second	No	Cylindrical polar
Disco (Excised)	2D	Moving-mesh finite volume	Second	Yes	Cylindrical polar
Fargo3D ^d	2D	Orbital advection finite difference and volume	Second	Yes	Cylindrical polar
Gizmo ^e	3D	Mesh-free finite mass	Second	No	Particles
Mara3 ^f	2D	Finite volume	Second	No	Cartesian
Phantom ^g	3D	Smoothed particle hydrodynamics	Second	No	Particles
Pluto ^h	2D	Finite volume	Second	No	Cylindrical polar
Sailfish ⁱ	2D	Finite volume	Second	No	Cartesian

Notes. A summary of the different codes used in this study. We provide additional details for each in the [Appendix](#).

^a Springel (2010); Muñoz et al. (2013).

^b Stone et al. (2020).

^c Duffell (2016).

^d Benítez-Llambay & Masset (2016).

^e Hopkins (2015); Franchini et al. (2022).

^f Zrake & MacFadyen (2012).

^g Price et al. (2018).

^h Mignone et al. (2007).

ⁱ Westernacher-Schneider et al. (2024).

$\Sigma_0 \ll 1$ in order to find agreement with fixed-binary results in the “test fluid” limit.

3.2. Initial Conditions

It is important for all codes to start from the same initial condition (or at least as close to one as possible). This ensures that any distinctions (no matter how small) stem from our numerical methods, and not differences in the true solution, and that all codes would converge to exactly the same solution at sufficiently high resolution.

The initial disk surface density employed here is a simplified version of (Muñoz & Lithwick 2020, Equation (4))

$$\Sigma(r) = \Sigma_0[(1 - \delta_0)e^{-(R_{\text{cav}}/r)^{12}} + \delta_0]f(r), \quad (2)$$

with $R_{\text{cav}} = 2.5a$ and $\delta_0 = 10^{-5}$, as grid-based codes may have difficulty with such a deep vacuum in their initial condition (this is not a density “floor”; the cavity can get deeper than this during the calculation).

The function $f(r)$ provides a means for truncating the disk, for those codes that are not able to specify general outer boundary conditions. For an infinite disk, $f=1$. For a finite disk,

$$f(r) = 1 - \frac{1}{1 + e^{-2(r-R_{\text{out}})/a}}, \quad (3)$$

with $R_{\text{out}} = 10a$. In practice, we will find that most results do not depend on whether the finite or infinite disk is chosen (see Section 6.3 for a comparison between finite and infinite disk solutions), as previously shown by Muñoz et al. (2020).

The equation of state is locally isothermal with sound speed

$$c_s^2 = -\Phi_b(t; r, \phi)h^2 = -\Phi_b(t; r, \phi)/\mathcal{M}^2 \quad (4)$$

for the (softened) binary potential $\Phi_b = \Phi_1 + \Phi_2$. We choose a disk with aspect ratio $h = 0.1$ (equivalently an azimuthal Mach number $\mathcal{M} = 10$).

For the initial angular frequency of gas in the disk, we first start with a solution that is in equilibrium far from the binary,

$$\Omega_0^2(r) = \frac{GM}{r^3}(1 - h^2) = \frac{GM}{r^3}(1 - 1/\mathcal{M}^2), \quad (5)$$

suitably modified to flatten in the vicinity of the binary, i.e.,

$$\Omega(r) = [\Omega_0(r)^{-4} + \Omega_B^{-4}]^{-1/4}. \quad (6)$$

For 3D particle-based runs, we adopted a Gaussian vertical density profile given by

$$\rho(r, z) = \frac{\Sigma(r)}{H} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{z^2}{2H^2}\right), \quad (7)$$

where $H = hr$ is the disk scale height. Note that $\Sigma(r)/H = \rho(r, 0)$ is the density on the disk midplane and

$$\Sigma(r) = \int_{-\infty}^{+\infty} \rho(r, z) dz. \quad (8)$$

The 3D Athena++ runs adopted a vertically isothermal equation of state, in analogy to the 2D runs, and specified the density profile so as to match the surface density prescribed by Equation (2) and to satisfy vertical hydrostatic equilibrium.

We assume a constant kinematic viscosity ν rather than constant α , as different codes may have different α viscosity implementations (this is also consistent with the assumed uniform surface density profile in steady state, far from the binary). We adopt the value

$$\nu = 10^{-3} a^2 \Omega_B, \quad (9)$$

which, for $h = 1/\mathcal{M} = 0.1$, corresponds to $\alpha = 0.1$ at $r = a$ and is also convenient to achieve steady-state solutions in only a few hundred orbits, due to the correspondingly short viscous time in the disk.

A key feature of the isothermal circumbinary disk response to an equal-mass binary is an elongated, lopsided ($m=1$) central disk cavity (e.g., MacFadyen & Milosavljević 2008). In

an attempt to seed cavity eccentricity in a uniform way across codes, we implement an initial perturbation or “kick” to the cavity via an initial radial velocity profile of the form

$$v_r(r) = v_0 \sin(\phi) (r/a) \exp\left[-\left(\frac{r}{3.5a}\right)^6\right], \quad (10)$$

with $v_0 = 10^{-4}\Omega_B a$. This initial perturbation is not included in most binary–disk studies in the literature; it is not required for the disk eccentricity growth, but if growth could be seeded by our initial condition rather than numerical noise, it might aid agreement between codes. In practice, we will find that this initial seed choice is too small to affect the outcome, and particle-based codes that had additional numerical noise unfortunately did not recover the initial value of the disk eccentricity induced by this kick. We note that we did not give the fluid an initial inward velocity consistent with the viscous drift rate, but this did not make a major difference in the overall solution.

3.3. Boundary Conditions

Possibly the biggest difference between numerical setups involves the boundary conditions. We have tried to design the problem such that differences in boundary conditions would not significantly affect the solution.

3.3.1. Outer

The outer boundary is typically chosen to be large enough to not affect the solution (we will test this explicitly in Section 6.3). Because different codes have very different boundaries (e.g., square vs. circular, or no boundary at all in the case of particle-based meshless codes), we do not require a specific outer boundary condition. *Disco*, for example, employs a Dirichlet boundary condition at the outer boundary, set at $r = 30a$, where the hydrodynamic quantities are fixed by the initial conditions.

3.3.2. Inner

The inner boundary condition depends on the setup. For excised binaries, this is determined at the excision radius ($r = a$). The standard boundary condition employed for excised binaries will be a “diode” boundary condition, which is essentially a Neumann boundary condition, unless the radial velocity of the fluid is positive, in which case the radial velocity is set to zero to prevent flow of gas from within the excision radius back into the domain. While most of the nonexcised codes resolved the origin at $r = 0$, the *PLUTO* code utilized a grid extending down to a radius of $r = 0.03a$ with a reflecting inner boundary.

3.4. Sink Prescription

For nonexcised binaries, there is no explicit inner boundary condition, but codes typically require some prescription for accreting mass and momentum (“sink prescription”). This is usually not done using a formal boundary condition, but instead via some prescription for either removing gas particles from the domain or including a sink term explicitly on the right-hand side of the evolution equations for the gas. This (sometimes loosely defined) nature of sink prescriptions can pose a challenge to convergence studies.

For practical reasons, we did not attempt to employ the same sink prescription across different codes. However, all codes have some prescription for accretion near the individual point masses. After discussing our fiducial model, we will also assess the effects of different sink prescriptions, to determine their importance and whether any one sink is a more accurate or well-behaved choice than others.

Particle-based codes *Gizmo* and *Phantom* employed sink particles, whereby any fluid parcel that came within a certain distance of the sink (and met certain velocity criteria) was instantly removed from the calculation (and in the case of a live binary, its mass and momentum are added to the sink particle). Grid-based and moving-mesh techniques typically employed source terms instead; *Athena++*, *Disco*, *Mara3*, *Sailfish*, and *Pluto* all employed some version of this source term, which took the form of an additional expression on the right-hand side of the equations for mass and momentum density evolution. This source term is then ideally well resolved in both space and time, such that the size of the affected area is at least a few computational zones across, and the timescale gas is removed on constitutes a large number of time steps. Employing a well-resolved sink prescription is generally expected to improve convergence properties of the code. *Arepo* employs a sink prescription that is very similar to the grid-based codes, but instead of a slow source term that is resolved in time, it reduces the density toward a fiducial value every time step, effectively giving a sink timescale that is proportional to the time step size.

Athena++ and *Disco* each employ different versions of a “torque-free” sink (Dempsey et al. 2020). Roughly speaking, the difference between a “standard sink” and a “torque-free” sink is the choice of how much momentum to remove when removing mass from the system. We will describe the method in more detail in Section 6.2, but the idea of a torque-free sink is to effectively be a subgrid model for what happens on much smaller scales than the sink, assuming that the disk extends down to much smaller radii. Any sink choice will naturally have some effect on the solution at radii outside the sink, but the torque-free sinks were designed to have as little effect as possible, when comparing a disk that is resolved at both large and small radii with a disk where gas is removed by the sink at some finite radius.

Which sink should be considered “most accurate”? This depends on the problem one wishes to solve and the physical interpretation for the sink. If the sink is meant to describe a finite radius where gas is accreted (such as the innermost stable circular orbit (ISCO) of a black hole), then short-timescale “standard” sinks might be a good choice, as they will capture the fact that spin is being added to the point mass. However, if one wishes to model a system where the inner radius of the disk is meant to be on much smaller scales than that resolved by the code, torque-free sinks are precisely designed and calibrated for this purpose. Thus, the most accurate choice of sink depends on the physical scenario, and code users should be mindful of the implicit choices being made in the choice of sink prescription.

4. Diagnostics

In this section we discuss the different diagnostics we used to compare codes. For each diagnostic the results from each code will be presented as a function of the spatial resolution. We note that, for particle-based/Lagrangian codes, the resolution changes throughout the duration of the simulation.

The details about how the resolution was attributed for each diagnostic will be described in the corresponding plot caption.

4.1. Torque

In this study, when we describe “gravitational torque,” we typically mean the transfer of angular momentum via the gravitational force between the disk and the binary, computed as $\vec{T} = \vec{r} \times \vec{F}_g$, with $F_g = -\int(\nabla\Phi)\Sigma dA$. For a circular orbit, this is just the radius of the binary component times the ϕ -component of the gravitational force:

$$T = r_1 F_1^\phi + r_2 F_2^\phi \quad (11)$$

$$\vec{F}_i = \int \Sigma (-\vec{\nabla}\Phi_i) dA. \quad (12)$$

The sign of the torque is defined so that a positive torque corresponds to angular momentum being *received* by the binary. Thus, positive torque works to increase the binary angular momentum (hence separation), and negative torque works to shrink the binary orbit.

In our initial set of runs, we did not compute “accretion torque” or the angular momentum directly accreted onto the binary. In the limit of a sufficiently small sink radius, or more generally sink terms that do not contribute to the spin angular momentum of either binary component, the orbital angular momentum accreted by the binary is $\dot{M}_1 j_1 + \dot{M}_2 j_2$ (where M_i and j_i are the accretion rate onto and specific angular momentum of binary member i), which for an equal-mass binary is by symmetry $0.25 \dot{M}a^2\Omega_b$.

To take into account the nonnegligible sink radius, we later found it useful to compute the angular momentum removed by the sink. Unless otherwise specified, “gravitational torque” computed in this study only accounts for angular momentum transferred directly via gravity, as opposed to “accretion torque.”

Often in this study, the torques will be “normalized” in units of $\Sigma_0 a^4 \Omega_B^2$, or $\Sigma_0 G M a$. However, we have found that the most robust measurement of gravitational torque is obtained by normalizing it to the accretion rate, using the dimensionless quantity $T_{\text{grav}}/(\dot{M}a^2\Omega_B)$. This can be related to the accretion “eigenvalue” l_0 via

$$l_0 = \frac{T_{\text{grav}} + \dot{J}_{\text{acc}}}{\dot{M}a^2\Omega_B}, \quad (13)$$

where \dot{J}_{acc} is the angular momentum directly removed by the sinks. In the limit of small sink radii, $\dot{J}_{\text{acc}} \rightarrow 0.25\dot{M}a^2\Omega_B$ for an equal-mass binary, but we found it useful to compute the accreted angular momentum explicitly; as we will show later, this will result in better agreement between codes.

4.2. Cavity Eccentricity

Another important diagnostic to measure is the overall morphology of the disk solution. In the first 100 orbits of binary evolution, the cavity becomes unstable, breaking the initial symmetry of the system and developing an eccentricity (e.g., MacFadyen & Milosavljević 2008). We wish to include a diagnostic that can measure the growth and saturation of this symmetry-breaking instability. We tested a few diagnostics and

found that the eccentricity vector

$$\vec{e} = \frac{|\mathbf{v}|^2 \mathbf{r} - (\mathbf{v} \cdot \mathbf{r}) \mathbf{v}}{GM} - \hat{\mathbf{r}} \quad (14)$$

provides the most reliable measurement of the asymmetry of the system.

The x and y components of this vector are

$$e_x = \frac{rv_r v_\phi}{GM} \sin(\phi) + \left(\frac{rv_\phi^2}{GM} - 1 \right) \cos(\phi) \quad (15)$$

$$e_y = -\frac{rv_r v_\phi}{GM} \cos(\phi) + \left(\frac{rv_\phi^2}{GM} - 1 \right) \sin(\phi), \quad (16)$$

or, in Cartesian coordinates,

$$e_x = \frac{jv_y}{GM} - x/r \quad (17)$$

$$e_y = -\frac{jv_x}{GM} - y/r, \quad (18)$$

with $j = xv_y - yv_x$. Under this definition,

$$|\vec{e}| = \sqrt{e_x^2 + e_y^2} \quad (19)$$

is the orbital eccentricity, while ϖ is the longitude of pericenter such that

$$e_x = |\vec{e}| \cos \varpi \quad (20)$$

$$e_y = |\vec{e}| \sin \varpi. \quad (21)$$

We computed a mass-weighted eccentricity vector between $r=a$ and $r=6a$:

$$\langle \vec{e} \rangle = \frac{\int_a^{6a} \int_0^{2\pi} \vec{e}(r, \phi) \Sigma dA}{35\pi \Sigma_0 a^2}, \quad (22)$$

where the denominator is a normalization constant that is designed to reduce this integral to an eccentricity value less than 1. The profile of $e(r)$ is another potentially useful diagnostic; however, it would require development of a uniform angle-averaging and time-averaging procedure across all codes, which we found cumbersome for a code comparison of this size.

4.3. Accretion

Accretion rates are measured directly by asking how much mass was removed by the sinks over the course of a time step at individual points in time. These accretion rates are then time-averaged according to the prescriptions given below.

4.4. Time Averaging and Derivatives

At sufficiently late times, diagnostics at single points in time become less useful, as there may be a great deal of temporal variability (and this variability is also quite sensitive to the sink prescription, which we are not keeping consistent between codes). Therefore, we need a process to time-average quantities. The simplest method would be to take a rolling average or “boxcar” smoothing. However, such a prescription needs very long duration in order to smooth out variability over a few-orbit timescale. Ultimately, this is due to the fact that a “top-hat” smoothing kernel has power at high frequencies. An

alternative is to use a convolution integral, in particular with a Gaussian smoothing kernel:

$$F_{\text{avg}}(t) = \frac{1}{\sqrt{2\pi}\sigma} \int dt' F(t') e^{-\frac{1}{2}(t-t')^2/\sigma^2}. \quad (23)$$

By smoothing on a timescale of only $\sigma = 2$ orbits, one can significantly suppress short-timescale variability in most codes, making it possible to focus on long-term trends. We perform such convolutions over the total torque, the excised torque ($r > a$), and the accretion rate, to compare code performance at late times.

It will often be useful to take the derivative of a highly variable function with time (especially the eccentricity vector). We can similarly define our derivatives by differentiating the smoothed version of a function with time:

$$\begin{aligned} \frac{dF}{dt} &\equiv \frac{dF_{\text{avg}}}{dt} \\ &= \frac{1}{\sqrt{2\pi}\sigma^3} \int dt' F(t') (t - t') e^{-\frac{1}{2}(t-t')^2/\sigma^2}. \end{aligned} \quad (24)$$

Thus, our derivatives can also be defined via a convolution integral. This helps to prevent any one data point from skewing the derivative.

4.5. Periodograms

Temporal variability in the accretion rate is decomposed into Fourier modes to generate periodograms. Similarly to the smoothing procedures above, periodograms are computed over a window of time by multiplying the accretion rate by a Gaussian smoothing kernel:

$$c(\omega) = \frac{1}{\sqrt{2\pi}\sigma} \int dt' \dot{M}(t') e^{-\frac{1}{2}(t'-t_0)^2/\sigma^2} e^{i\omega t'}, \quad (25)$$

where $\sigma = 15$ orbits and $t_0 = 250$ orbits. The power is then computed via $|c(\omega)|^2$. This means that the accretion variability is measured in a window of time roughly between $200 < t < 300$, but without the high-frequency artifacts that would be present if we cut off the integrals abruptly within the window. The choice of $\sigma = 15$ orbits means that our periodograms are only valid on timescales shorter than this.

5. Time Evolution

We investigate the evolution of the disk for up to 300 binary orbits (with some codes running to 1000 orbits), by which time the system has settled into a relaxed (quasi-)steady state. Before this time, the solution still depends on the details of the initial conditions; we explore the evolution from this initial state to the final state at different stages. We first look in detail at the first orbit, in part as a consistency check on the details of the initial conditions.

5.1. The First Orbit

Figure 1 shows the high-resolution results obtained using the `Sailfish` code after one orbit, as an example of the calculations performed. During the first binary orbit, the initially axisymmetric disk begins to viscously spread and is strongly perturbed by the gravitational influence of the binary. Nascent circumprimary and circumsecondary accretion disks (generically “minidisks”) form, initially from the ambient gas

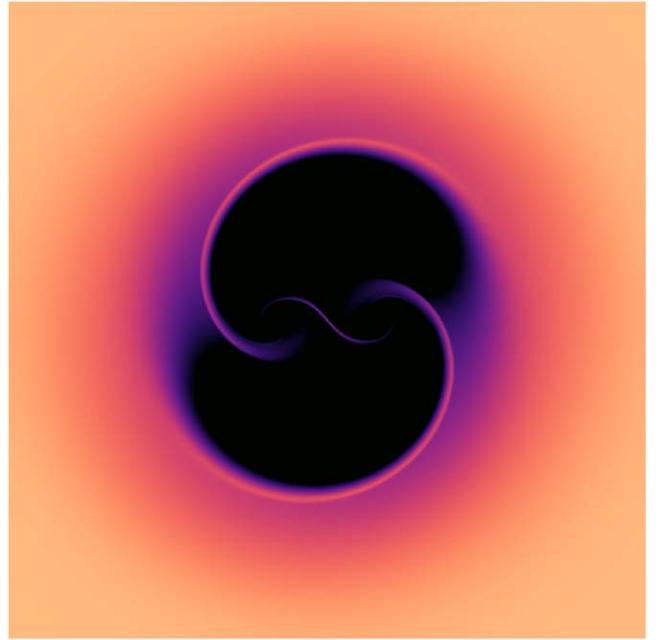


Figure 1. Logarithm of the surface density in the `Sailfish` code after one orbit, at the highest uniform resolution of all codes in the comparison study (4000×4000 , or $\Delta x = 0.005a$). For this and other 2D density maps throughout, the spatial grid ranges from $-2.5a$ to $2.5a$ in each dimension, and the color map is linear, ranging from -3 (black) to 0.5 (yellow). The color map is the same as in Figure 3.

within the cavity, but later from gas stripped from the circumbinary disk. Some of the gas that is stripped from the spreading cavity walls streams past the binary member that initially perturbed it and shocks against the gas symmetrically perturbed by the other binary member, leading to a higher-density bridgelike feature between them.

The first orbit provides an opportunity to test that all codes have implemented the setup and initial conditions in precisely the same way. Capturing the evolution of the first orbit is reasonably straightforward; in this first orbit, all codes quickly converge to the same solution. Figure 2 shows the torque as a function of time during this first orbit, again showing that a converged torque is found and agreed upon by all codes at these times. Excised torque was computed (using contributions from fluid elements with $r > a$) in part because all codes could be compared, including those that excised the binary. However, during the first orbit (and similarly the first 10 orbits), very little torque comes from within $r < a$, as very little gas has accreted onto the binary at this stage.

The middle panel of Figure 2 shows convergence of the `Disco` code at five different resolutions, demonstrating that the solution is very well converged. Increasing the resolution by a factor of four has almost no effect on the curve $T(t)$; differences between the different resolution runs are at the percent level.

The bottom panel of Figure 2 shows the excised torque at one orbit as a function of resolution for all codes. All codes converge toward the same value of this torque. The converged excised torque at one orbit is found to be $T/(\Sigma_0 G Ma) = -7.2 \times 10^{-3} \pm 10^{-4}$, where error bars are roughly set by differences between codes. The only (minor) deviations are from particle-based codes that have additional Poisson noise from the positions of the initial computational particles (or zones). These codes can still capture

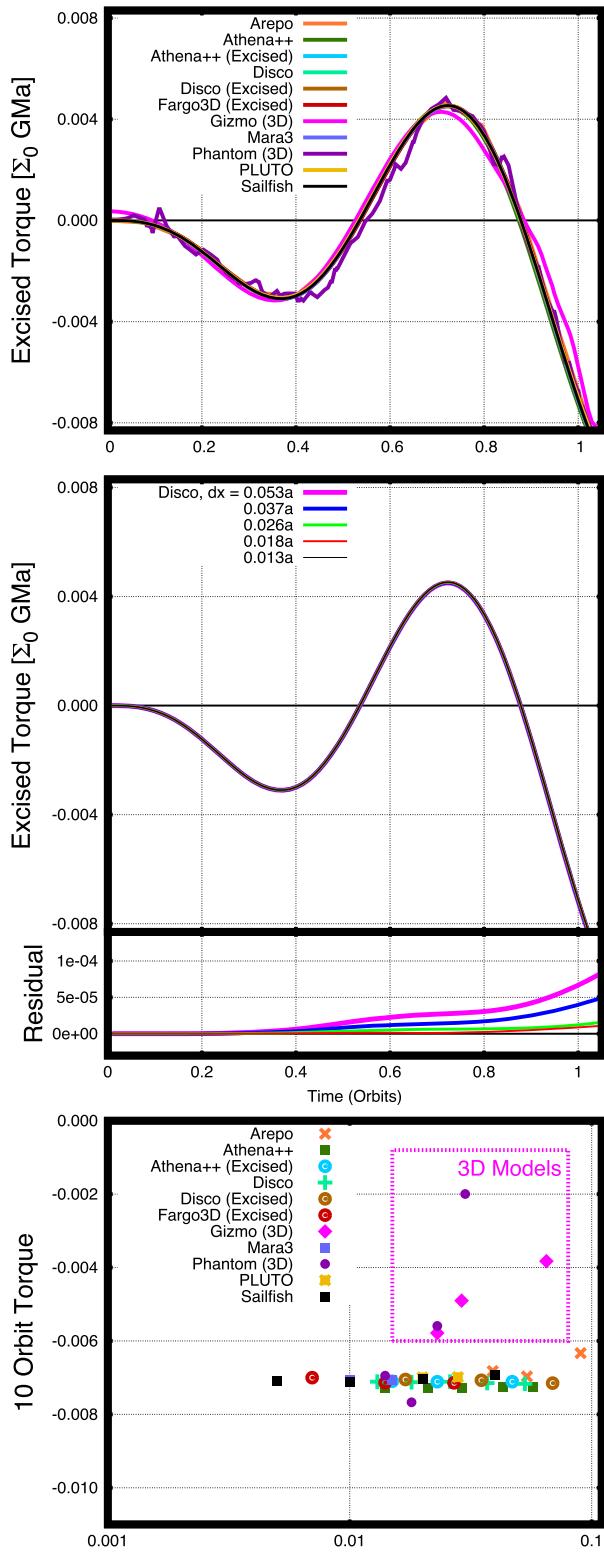


Figure 2. Torque convergence over the first orbit. All codes are plotted in the top panel, lying on essentially the same curve (consistent with the highest-resolution *Sailfish* run). A demonstration of convergence in the *Disco* code is shown in the middle panel, showing very little deviation between resolutions. Residuals are included (taking the difference between the output and the highest-resolution run), showing rapid convergence of the solution. The value of the torque at one binary orbit is plotted in the bottom panel, for all codes, as a function of the resolution length scale. The resolution is measured in terms of the zone size $\Delta x/a$ at $r = 3a$.

time-averaged torques reliably at these times, as seen in the next section.

Figure 3 shows a plot of surface density at one orbit for all numerical methods in our study. In this and future large comparative images, all codes are using the same effective resolution of $\Delta x = 0.02a$ at $r = 3a$ (though some codes have higher resolution on the minidisks and lower resolution farther out in the disk). The basic structures are captured similarly by all codes; particle-based code *Phantom* exhibits less well-defined shocks. We later demonstrate this to be a resolution-dependent effect, which can be cured by very high particle count; because particle-based codes typically resolve mass elements rather than volume elements, at these early times, when no gas is yet in the cavity, they can have very low effective spatial resolution near the origin (though in this study *Gizmo* mitigates this effect via particle splitting; see Franchini et al. 2022 for details).

5.2. The First 10 Orbits

After the first few orbits, the accretion of gas onto the binary settles into a fairly regular pattern, as some gas from the disk is captured by the binary, forming minidisks. Waves have begun to propagate outward through the circumbinary disk, and some gas is able to flow from one minidisk to the other. An example showing the calculation after 10 orbits at high resolution is given in Figure 4 (using the *Sailfish* code as an example). On a 10-orbit timescale, nonlinear effects make convergence less trivial but still reasonably attainable. The minidisks have begun to accumulate a bit of gas already, but nearly all torque still comes from outside $r > a$. The cavity still appears to be symmetric, respecting the symmetry of the binary and the initial conditions. Figure 5 shows the torque as a function of time in the *Sailfish* code at many resolutions. It is evident that *Sailfish* is able to achieve a converged torque as a function of time at sufficiently high resolution (at least for these first 10 orbits).

The middle panel of Figure 5 compares all of the codes during this time, using the bottom panel to highlight which codes attain very similar torque curves. While the outliers exhibit somewhat noisier torque curves, all codes are able to agree on the time-averaged torque over these timescales; Figure 6 shows the time average of the torque between 8 and 10 orbits, computed by a simple numerical integration over this window of time. All codes at all resolutions are plotted, demonstrating that it is possible for all codes to achieve the converged solution at sufficiently high resolution. During the first 10 orbits, the gravitational torque is usually negative. Between 8 and 10 orbits, the time-averaged gravitational torque is $T/(\Sigma_0 GMa) = -0.0046 \pm 10^{-4}$, where the uncertainty is (roughly) quantified using the spread between different codes in the plot at their highest resolutions.

Figure 7 shows the surface density at 10 orbits for all numerical methods in our study. As the minidisks accumulate mass, the particle-based codes show improved spatial resolution within the cavity, but discrepancies in minidisk morphology still remain, specifically when considering the *Phantom* code. In Figure 8, we demonstrate that *Phantom* can achieve agreement with the other codes, when the cavity is sufficiently well resolved. However, at 10 orbits the particle count in the cavity is still sufficiently low in the fiducial runs that the

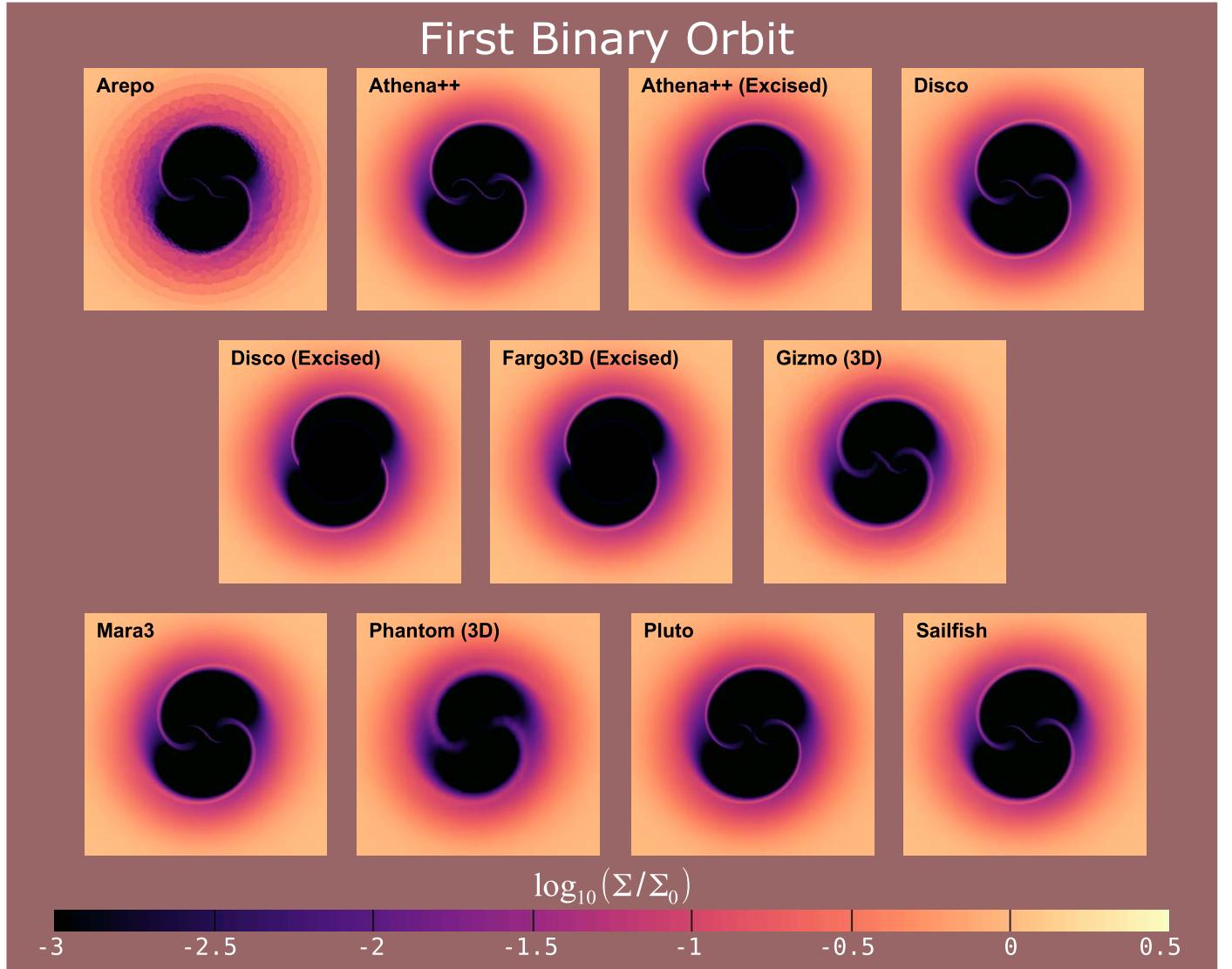


Figure 3. Surface density for all codes at one orbit. Codes are compared with similar zone spacing $\Delta x = 0.02a$ at $r = 3a$. During this first orbit, tenuous streams of gas begin to accrete onto the binary, and extremely low density minidisks begin to form. Part of each stream passes by the first binary member encountered, leading to a higher-density shock near the barycenter.

minidisks are yet unresolved. Interestingly, each code appears to find a different minidisk surface density at this time, likely reflecting discrepancies in sink prescriptions between codes. Excised codes are able to capture the outer disk morphology accurately.

Figure 7 indicates that binary–disk interaction is a challenging problem. After only 10 orbits, significant visual discrepancies in minidisk morphology develop across the different codes. This visual trend will continue at later stages. However, when measuring averaged quantities such as torque or accretion rate, we will find that the different codes can reach agreement. Thus, despite visual differences in the instantaneous snapshots, it will be possible to find convergence to a common value for bulk integrated quantities.

5.3. The First 100 Orbits

An example showing the calculation after 100 orbits at high resolution is shown in Figure 9 (using the *Sailfish* code as an example). Until this point, the disk has been nearly symmetric. During the first 100 orbits, however, this picture

gradually changes. This eccentric “instability” was first noted in MacFadyen & Milosavljević (2008), which used the FLASH code with an excised binary. It has been characterized by Shi et al. (2012) and D’Orazio et al. (2013) and has been shown to manifest itself at mass ratios as low as $q \gtrsim 0.01$, although its amplitude varies depending on disk and binary properties (e.g., D’Orazio et al. 2016; Ragusa et al. 2020; Noble et al. 2021; Dittmann & Ryan 2024). The conditions for its stability and growth rate as a function of all disk and binary parameters have not yet been completely explored.

A model of the instability was given in Shi et al. (2012). Streams carrying momentum and mass are excited by the binary and impart energy and momentum onto fluid elements at the cavity edge. A preference for one stream becoming stronger than the other creates a feedback cycle that causes stream impacts on one side of the cavity to dominate (D’Orazio et al. 2013), hence increasing the eccentricity of the inner circumbinary disk and cavity. An overdense “lump” builds up and orbits at the cavity edge. This lump orbits with the orbital period of the cavity edge, which for the 2D simulations in this work is very close to five

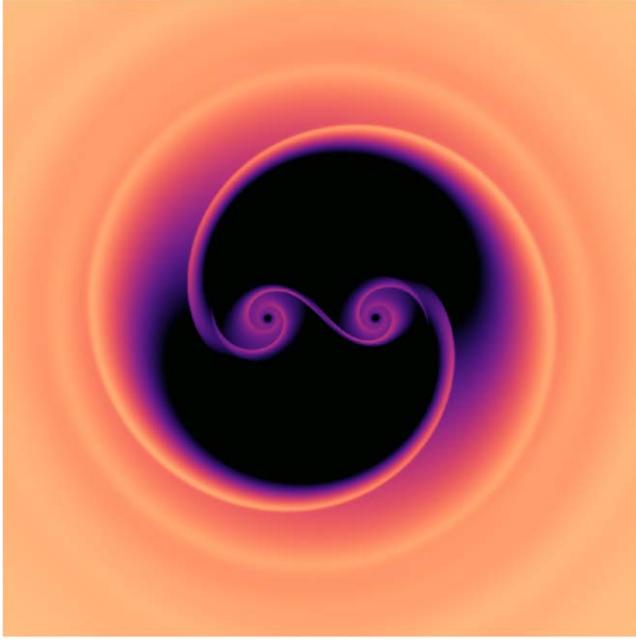


Figure 4. Logarithm of the surface density in the Sailfish code after 10 orbits. The color map is the same as in Figure 7.

binary orbital periods. However, the size of the cavity and thus its orbital period depend strongly on the properties of both the binary and the disk (e.g., D’Orazio et al. 2013; Farris et al. 2014; Noble et al. 2021; Dittmann & Ryan 2024), so this five-orbit periodicity is far from universal.

Eccentricity growth impacts all measured quantities. For example, prior to eccentricity saturation, all codes found a negative torque on the binary, which would lead to inward migration (for this test problem), and a sudden shift in the torque toward outward migration occurs around the time when the instability nears saturation. This was true for all codes, regardless of when the cavity eccentricity saturated. The two appear to be causally linked, but the details depend on the disk and binary parameters. In some cases, disks that are stable and do not achieve an eccentric cavity morphology result in inward migration (Franchini et al. 2022; Wang et al. 2023b; Dittmann & Ryan 2024).

We have found that the growth of this instability and subsequent precession of the cavity are clearly represented using the density-weighted eccentricity vector \vec{e} defined previously (Equation (22)). The magnitude of this vector over time captures the growth and saturation of the cavity eccentricity, while the phase of this vector nicely tracks the precession of the cavity.

We plot each of these quantities as a function of time in Figure 10. All codes find similar growth and saturation of the instability, though some codes find growth beginning at different start times from others. 3D codes notably find a faster cavity precession rate, by $\sim 60\%$.

The growth rate Γ_c can be measured as $\Gamma_c = \dot{e}/e$:

$$\Gamma_c \equiv \frac{\dot{e}_x e_x + \dot{e}_y e_y}{e_x^2 + e_y^2}. \quad (26)$$

The precession rate $\Omega_c = \partial_t(\varpi)$, or

$$\Omega_c = \frac{\dot{e}_y e_x - \dot{e}_x e_y}{e_x^2 + e_y^2}. \quad (27)$$

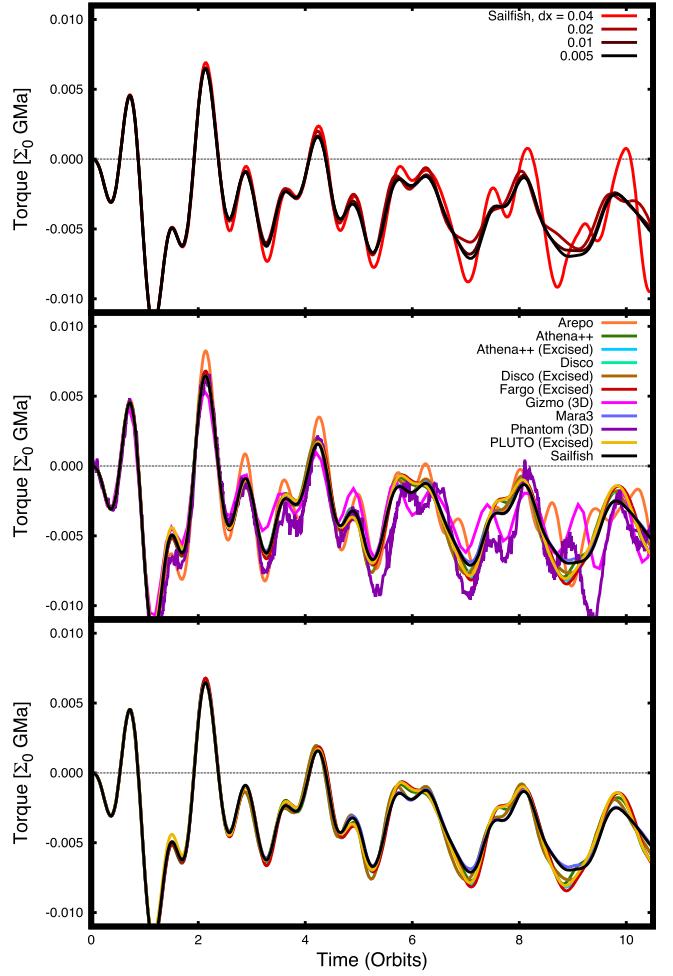


Figure 5. Torque comparison between all codes in the first 10 orbits. In the top panel, the excised torque ($r > a$) from the Sailfish code is compared at different resolutions, showing that the code approaches a consistent torque as a function of time at sufficiently high resolution. All codes are plotted at their highest resolution in the middle panel; the bottom panel highlights a subset of eight codes that agree the most precisely during the first 10 orbits.

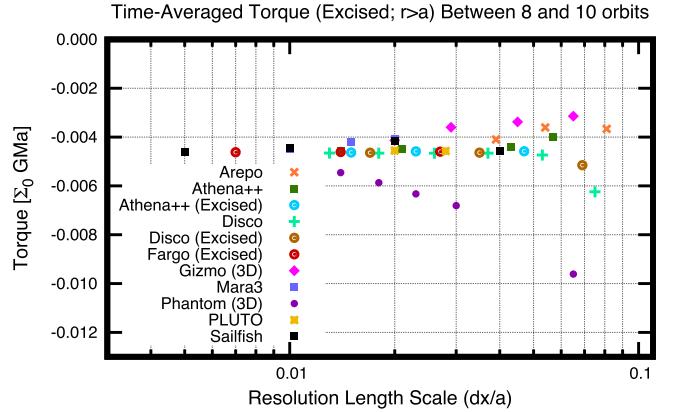


Figure 6. Convergence of the time-averaged torque from 8 to 10 orbits for all codes in this study.

Γ_c and Ω_c are computed for all codes, at all resolutions, and the result is plotted in Figure 11. The final saturated value of the eccentricity depended on whether the disk was finite or infinite, whether the binary was excised, and whether the disk was modeled in 2D or 3D. Thus, there was no consensus on a final

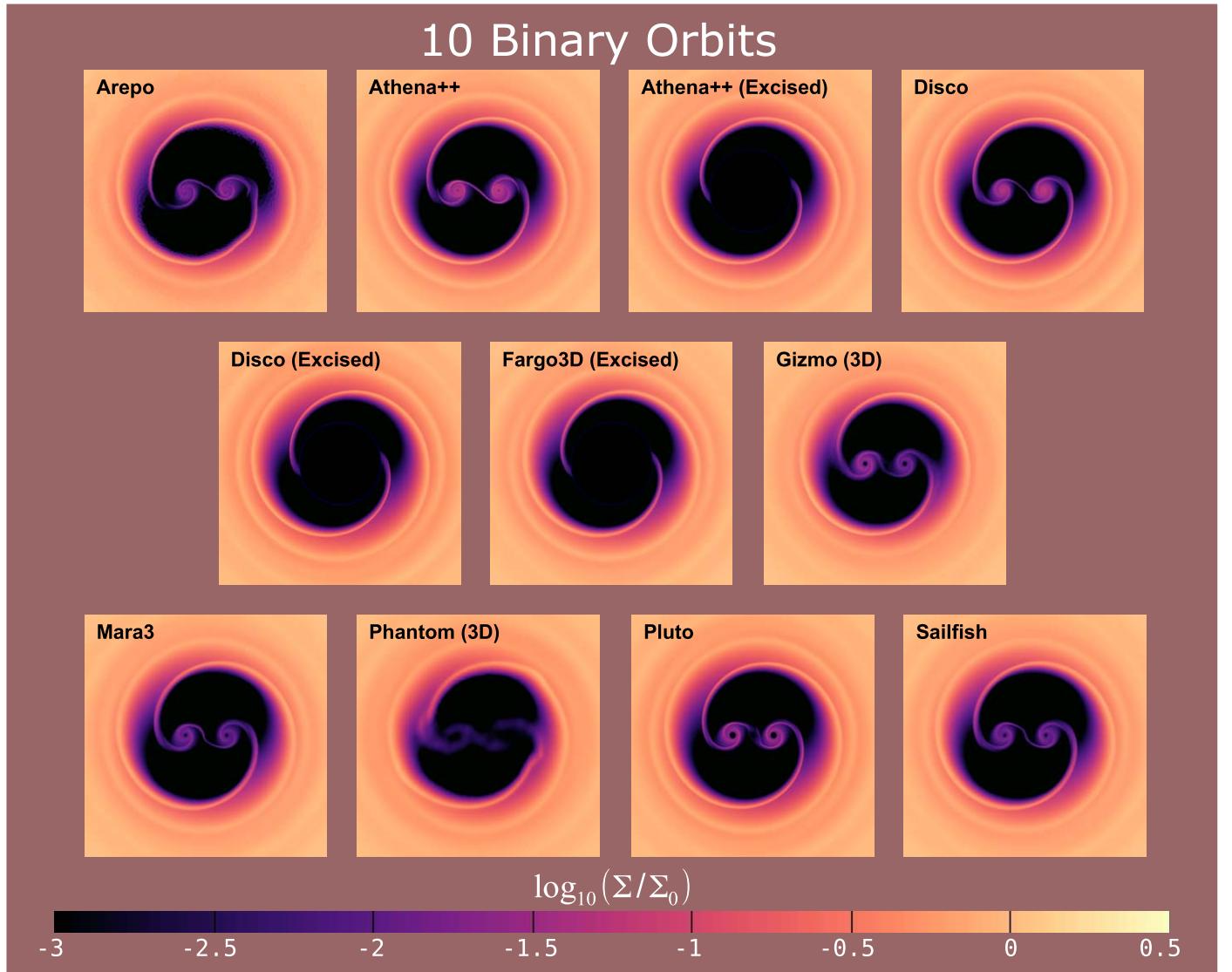


Figure 7. Surface density for all codes at 10 orbits. Codes are compared with similar zone spacing $\Delta x = 0.02a$ at $r = 3a$. Although discrepancies exist between codes at this resolution, it should be noted that many of the codes are capable of running at much higher resolution than in this figure; the idea is to attempt to compare all codes using roughly the same zone or particle spacing at $r = 3a$. Note that this means some codes will still have much better resolution near the origin.

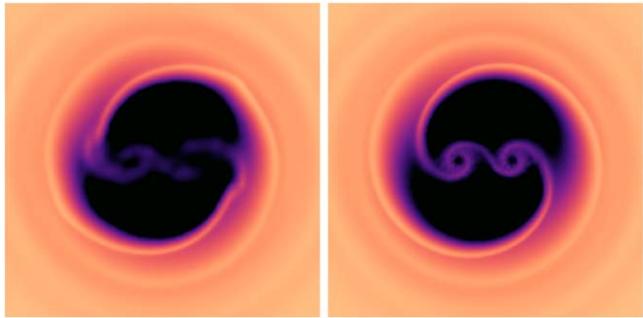


Figure 8. Phantom code at 10 orbits, comparing resolutions of 20 million particles (left) vs. 100 million particles (right). The color map is the same as in Figure 3.

saturated value of e , even though the converged growth rate was consistent between all codes.

All codes show an exponential behavior, with a growth rate $\Gamma_c/\Omega_B \approx 0.012$ (about a 13-orbit e -folding timescale) that appears roughly consistent across all codes (in both 3D and

2D). The cavity precession rate is $\Omega_c/\Omega_B \approx 0.004$ (240 orbit precession period) for all the 2D codes. In 3D, the cavity precesses 60% faster, giving a period of about 150 binary orbits. To disambiguate between effects of code choice and effects of 3D versus 2D, we have also run *Athena++* in 3D and *Gizmo* in 2D. We will discuss further the differences between 2D and 3D in Section 6.

Figure 12 shows a plot of surface density at 100 orbits for all numerical methods in our study. At this time, it can be difficult to compare these images directly, although this is a natural result of the nonlinear hydrodynamic processes that we have simulated. Because disk eccentricity was seeded by numerical perturbations, each disk began developing eccentricity at a different time and with a slightly different preferred orientation, leading to a variety of cavity orientations after and leading up to saturation.

5.4. Relaxed Disk—300 Orbits

The maximum time that most of the codes were run for was 300 orbits. As a representative example, *Sailfish* is shown at

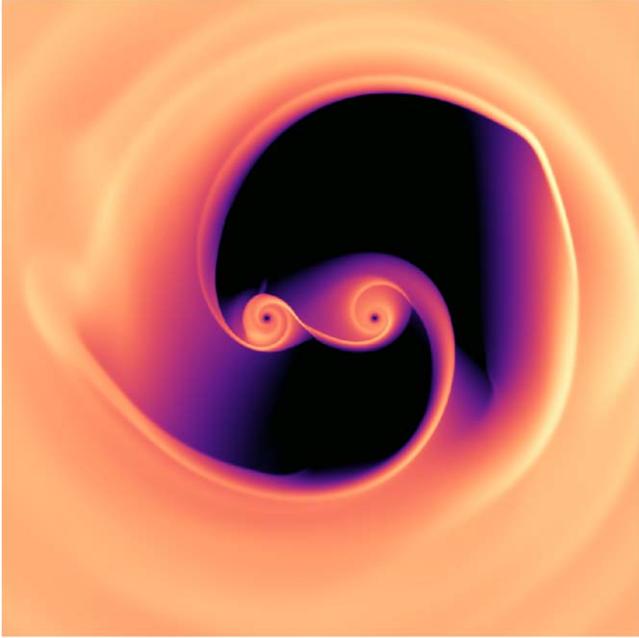


Figure 9. Logarithm of the surface density in the *Sailfish* code after 100 orbits. The color map is the same as in Figure 12.

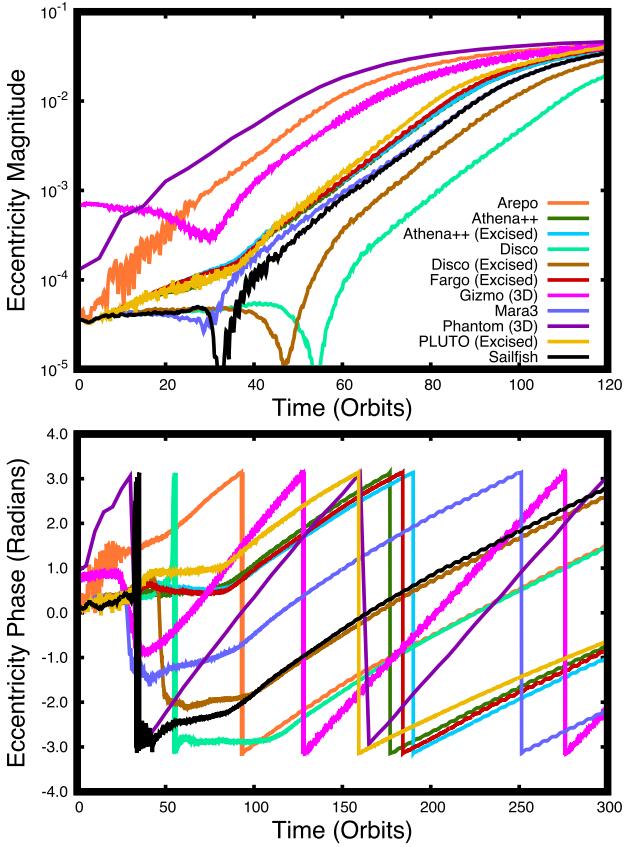


Figure 10. Eccentricity vector over time for all codes. The top panel shows the magnitude, and the bottom panel shows the phase. The growth of the instability and subsequent precession of the cavity begin at different times for different codes. Nevertheless, the growth rate and precession rate have reasonably consistent values when comparing between codes.

high resolution in Figure 13. A total of 300 orbits is sufficient for disk eccentricity to saturate and the disk to attain a quasi-steady state (this is confirmed in the next subsection, where

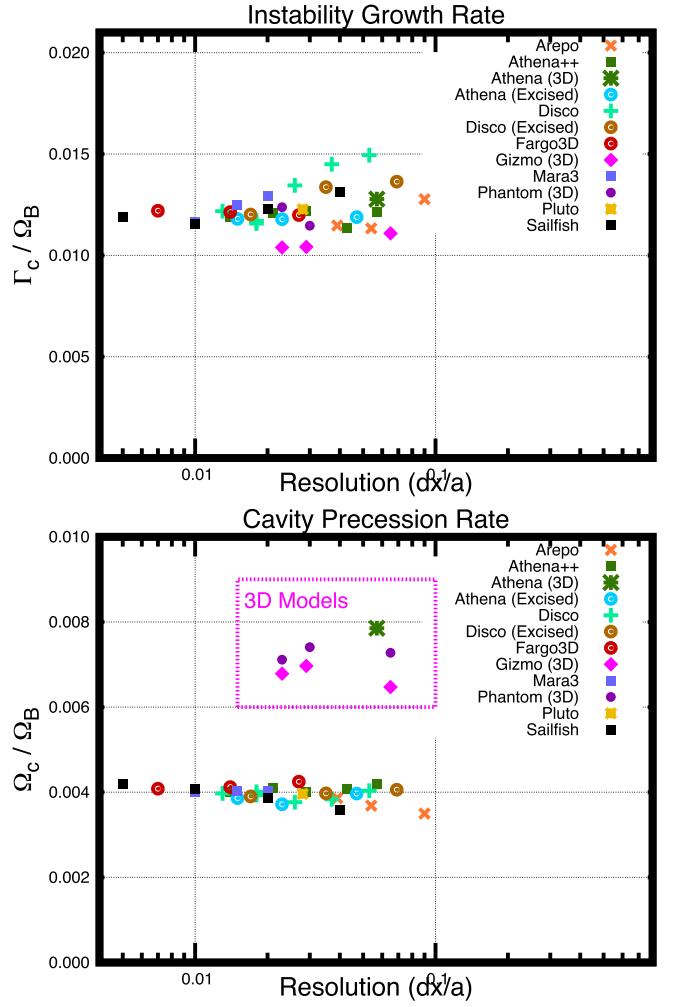


Figure 11. Growth rate and cavity precession rate for all codes at all resolutions. The precession rate is measured at 150 orbits. The growth rate is measured at a time when the eccentricity has grown to $e = 3 \times 10^{-3}$, when all codes find exponential growth (this occurs at different times for different codes; see Figure 10). The growth rate (Γ_c/Ω_B) converges to a rate of about 0.012, corresponding to an e -folding time of about 13 binary orbits. The cavity precession period ($2\pi/\Omega_c$) converges to a period of about 240 binary orbits. 3D codes find a shorter period, about 150 binary orbits.

some of the codes are run to 1000 orbits). After the disks become visibly eccentric, codes are unable to agree at specific moments in time. This is because the growth of the instability is sensitive to small details. Though we tried to mitigate this somewhat by including an initial seed perturbation (Equation (10)), these details nonetheless offset the phase of the cavity relative to the binary at late times, and therefore comparing codes at a single time becomes less meaningful. We therefore require time averaging in order to mitigate this effect and extract long-timescale trends that can be compared between codes. To this end, we perform a convolution integral with a Gaussian kernel (details in Section 4.4) to time-average the values of torque and accretion rate as a function of time, with the results reported in Figure 14.

Torques switch sign around 100 binary orbits—coincident with the growth and saturation of the cavity eccentricity. In Figure 15, we plot the gravitational torque (normalized to accretion rate) for all codes in this study, as a function of resolution. Here we find significant discrepancies between our codes, so it will now be necessary to diagnose these discrepancies

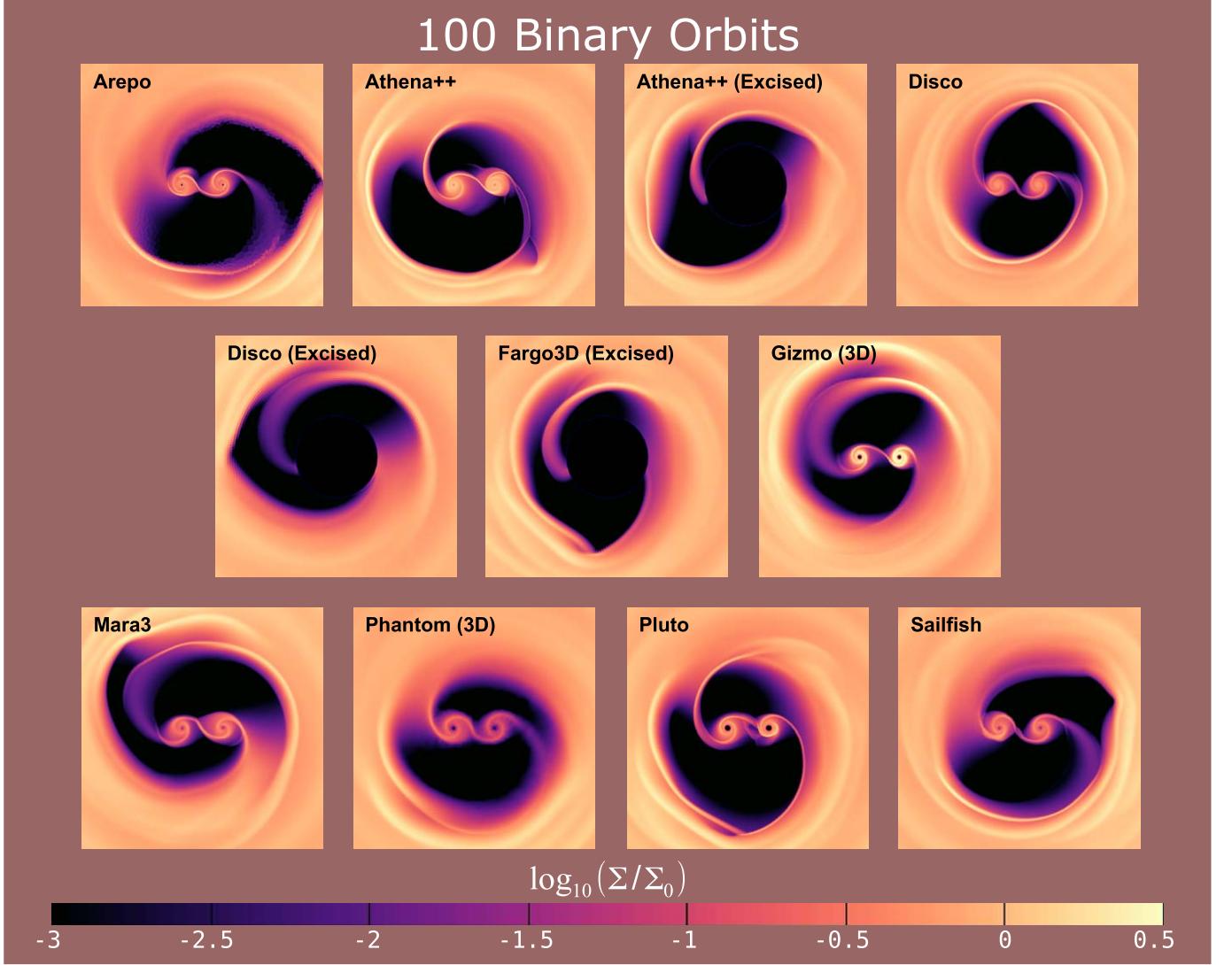


Figure 12. Surface density for all codes at 100 orbits. Codes are compared with similar zone spacing $\Delta x = 0.02a$ at $r = 3a$. The discrepancies in surface density between codes result from the different rates and times at which disk eccentricity grows in each simulation, as illustrated in Figure 10. Though we seeded this instability with a finite perturbation, the resulting growth is very sensitive to details and results in a different measured cavity phase between codes. In addition, the cavity precession rate is about 1.6 times as fast in 3D, so that there should be no expected agreement in the orbital phase of the asymmetry between 3D and 2D.

and determine whether they are code related or due to minor differences in our code setup. At first glance it appears that there is a cluster of codes that find torques of $T_{\text{grav}} \sim 0.4\text{--}0.5\dot{M}a^2\Omega_B$, while PLUTO, Gizmo, and Phantom all converge to much smaller gravitational torques. In fact, taken at face value, these three codes would find inward migration, while the rest of the codes would predict outward migration.

The largest difference between some of codes is in the dimensionality of the setup; Gizmo and Phantom performed calculations in 3D, whereas the rest of the codes performed the calculation in 2D. Actually, it is a little surprising that differences between 2D and 3D are not more severe. We will also show that the next major difference between codes is in the choice of sink prescription, which is responsible for the deviations of PLUTO from the other 2D codes. However, discrepancies in the sink prescription can be accounted for by properly accounting for the angular momentum eaten by the sink. Thus, while Figure 15 appears to show significant discrepancies between codes, we will find that these discrepancies can be understood and resolved (see Section 6.2). The

hydrodynamic state of all codes at 300 orbits is shown in Figure 16.

5.5. Accretion Periodograms

As one additional code comparison at late times, we compute accretion periodograms, by taking the Fourier transform of $\dot{M}(t)$ according to Equation (25) and computing the square of the amplitude $c^2(\omega)$. This is plotted in Figure 17. All codes in our study find a variability peak on timescales of both one orbit and four to five orbits, in addition to many other peaks in the periodogram. However, the amplitudes of the various peaks in the periodogram show a great deal of variation between codes. Much of this discrepancy is likely due to the different choices in sink prescription. We note that Athena++ and Disco, for example, exhibit less variability than some of the other codes on one orbit and shorter timescales. In contrast, codes such as Sailfish and Arepo find very large variability on orbital timescales and shorter. Particle-based codes Gizmo and Phantom exhibit even higher degrees of variability on all timescales.

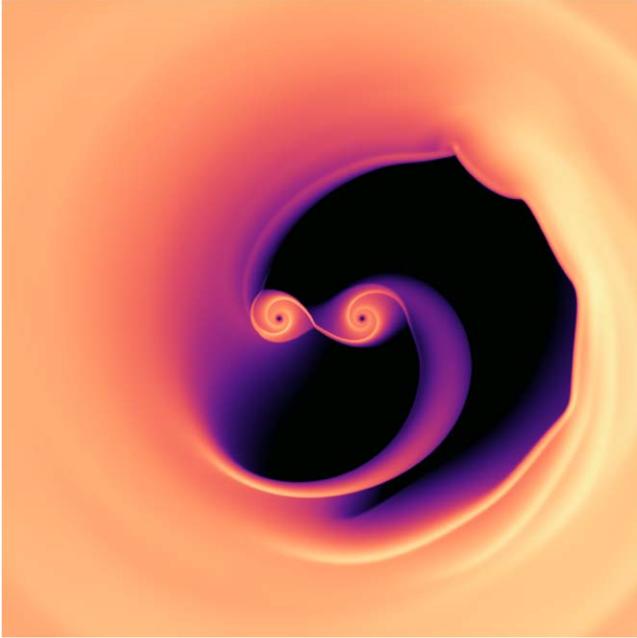


Figure 13. Logarithm of the surface density in the *Sailfish* code after 300 orbits. The color map is the same as in Figure 16.

The differences in sink parameters used by each code are responsible for the most major differences in the periodograms between codes, with faster, larger, and standard (vs. torque-free) leading to greater variability. Dittmann & Ryan (2021) explored this in a more controlled setting, demonstrating that at a constant sink rate “standard” sinks lead to higher-amplitude variability than torque-free sinks and that for a given sink method faster sinks lead to higher-amplitude variability (though to a much smaller extent when using torque-free sinks). The same behavior—the presence of minidisks reducing (quasi-)periodic accretion signatures—has also been observed on a physical rather than numerical basis in numerical relativity simulations (e.g., Bright & Paschalidis 2023).

As for the question of which sink prescription is more accurate, it depends on what the user intends to model on small scales. The torque-free sinks are designed to model disks that extend down to radii well below the grid scale. However, if the sink is meant to represent a resolved inner radius of the minidisk, such as the ISCO of a black hole, then a faster sink (and one that permitted accretion of spin) might mimic such an inner boundary condition. Ultimately, one must try to match their sink prescription to the physical problem at hand and keep in mind the effects of these numerical choices on the periodicities identified by each simulation.

5.6. 1000 Orbits—Checking the Steady State

By running our codes for 1000 orbits, it becomes possible to establish the time it takes to reach a (quasi-)steady state. The relaxed state of this system is typically achieved after about 200 binary orbits. This timescale is determined by both the growth rate of the instability and the number of e -foldings that the system must grow by to reach saturation. Accretion rate as a function of time up to 1000 orbits is plotted in Figure 18 for those codes that employed an infinite disk model.

Because our initial conditions assumed zero angular momentum current through the disk (see Lynden-Bell & Pringle 1974; Muñoz & Lithwick 2020, Equation (2)) but the

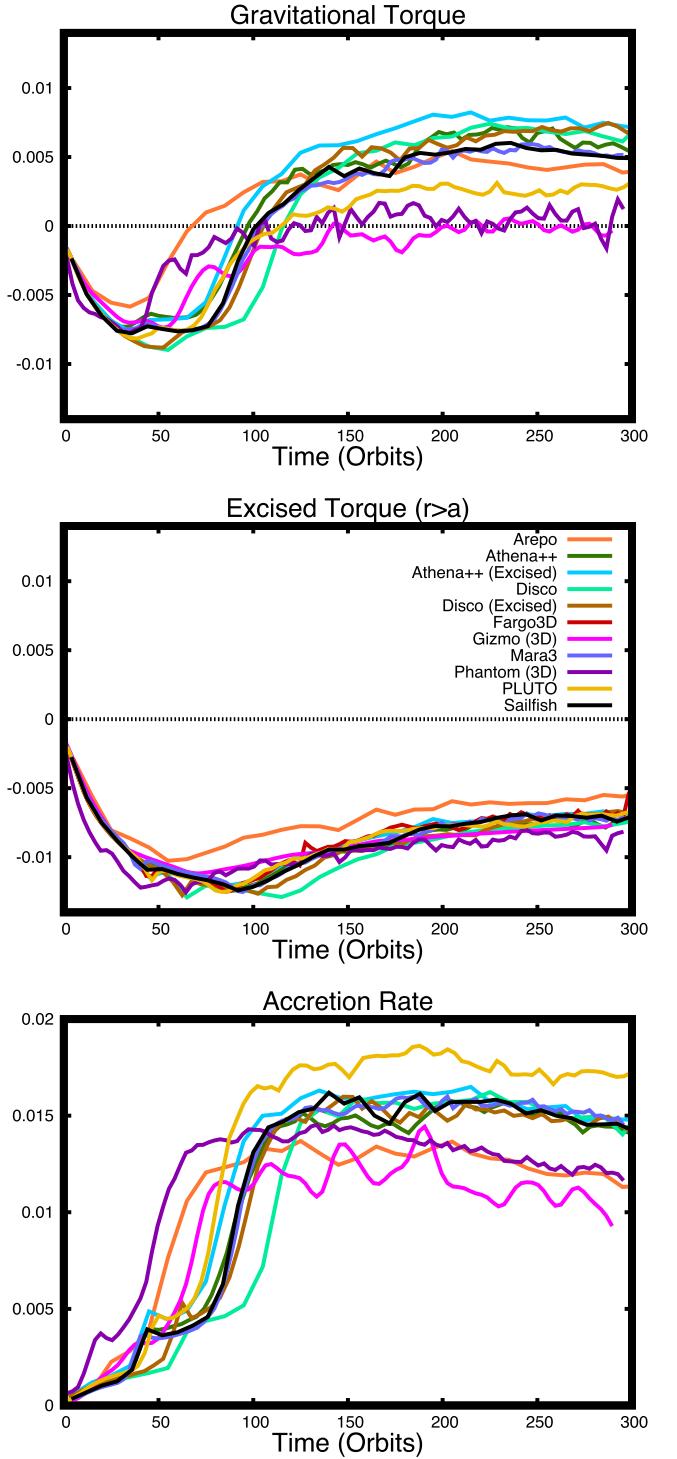


Figure 14. Smoothed torque and accretion rates as a function of time for all codes in this study. At around 100 orbits, this torque flips sign from negative to positive, demonstrating the importance of running codes for sufficiently long timescales to allow the instability to develop in the disk. The middle panel shows the excised torque, from fluid elements outside $r > a$. The bottom panel shows the smoothed accretion rate. The largest discrepancies in accretion rate can be traced back to discrepancies in the initial disk models (finite vs. infinite disk; see Section 6.3).

angular momentum current through the disk after settling into a quasi-steady state through interaction with the binary was positive ($\dot{J} \approx 0.7\dot{M}$), our simulations measured greater accretion rates than the steady-state value ($\dot{M}_0 = 3\pi\nu\Sigma_0$). This enhancement following underestimation of the angular

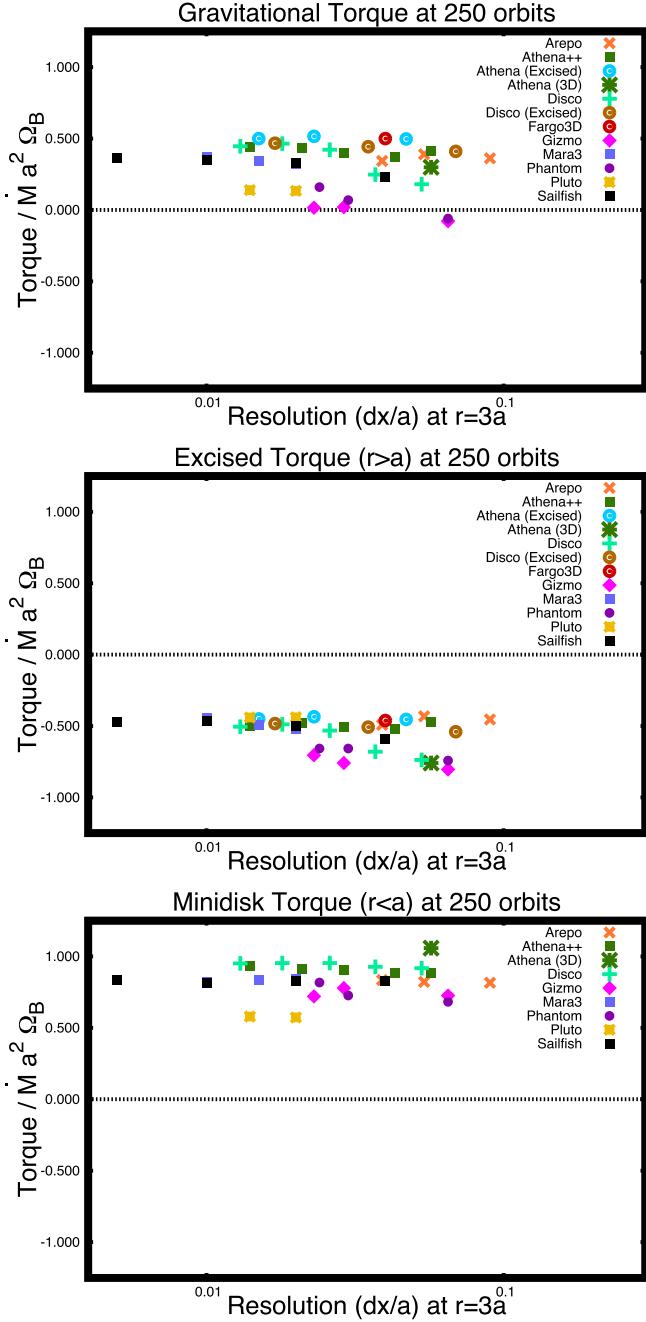


Figure 15. Time-averaged gravitational torque, normalized by $\dot{M}a^2\Omega_B$, for all codes at all resolutions. The top panel shows the total gravitational torque on the binary, whereas the middle and bottom panels separate this into excised torque (from $r > a$) and “minidisk torque” (from $r < a$). On the surface it appears that the different codes disagree, but these discrepancies can be resolved by considering the dimensionality of the problem and adjusting for the sink prescription (seen in Section 6.2).

momentum current was demonstrated in Miranda et al. (2017), and relative suppression in the case of overestimation of the angular momentum current was noted in Dittmann & Ryan (2022).

However, our simulations do gradually approach the asymptotic quasi-steady accretion rate, given sufficient time. Since the viscous time at a given radius R is $t_\nu \sim R^2/\nu$, one can define a viscous radius $R_\nu(t)$, the radius within which we have achieved a steady state and outside of which is out of

equilibrium (and determined by initial conditions),

$$R_\nu(t) \sim (\nu t)^{1/2}. \quad (28)$$

Inside of this radius, the steady state can be determined by the flow of mass and angular momentum

$$\dot{J} = \dot{M}j - 3\pi\nu\Sigma, \quad (29)$$

where \dot{J} and \dot{M} are both positive quantities, defined as the inward flow of angular momentum and mass, respectively. The right-hand side of the equation is given by the advective flux of angular momentum (first term) and the outward flow of angular momentum due to viscosity (second term).

This equation can be inverted to determine a solution for $\Sigma(R)$ within $R < R_\nu(t)$,

$$\Sigma(R) = \frac{\dot{M} - \dot{J}/j(R)}{3\pi\nu}, \quad (30)$$

where $j(R) = R^2\Omega(R)$ is the specific angular momentum at the radius R . Using $\dot{J} = l_0\dot{M}a^2\Omega_B$, this can be expressed as

$$\Sigma(R) = \dot{M} \frac{1 - l_0\sqrt{a/R}}{3\pi\nu}, \quad (31)$$

which is the solution within $R < R_\nu$. Outside, we can assume that the disk has not had time to change its surface density and $\Sigma = \Sigma_0$. This effectively sets a boundary condition of $\Sigma = \Sigma_0$ at $R = R_\nu(t)$, so that

$$\Sigma_0 = \dot{M} \frac{1 - l_0\sqrt{a/R_\nu(t)}}{3\pi\nu}. \quad (32)$$

Now we can invert this expression to find a solution for \dot{M} as a function of time,

$$\frac{\dot{M}(t)}{3\pi\nu\Sigma_0} = \frac{1}{1 - l_0\sqrt{a/R_\nu(t)}}. \quad (33)$$

Using expression (28) above for R_ν and including a dimensionless overall coefficient κ ,

$$R_\nu(t) = \kappa\sqrt{\nu t}, \quad (34)$$

we arrive at the following expression for \dot{M} :

$$\frac{\dot{M}(t)}{3\pi\nu\Sigma_0} = \frac{1}{1 - l_0(\kappa^2\nu t/a^2)^{-1/4}}. \quad (35)$$

Using the eigenvalue $l_0 = 0.7$, we find that the coefficient $\kappa = 3.2$ gives a good fit to the accretion rate (see Figure 18).

For an infinite disk, this accretion rate will eventually trend toward the asymptotic value $3\pi\nu\Sigma_0$ (the value one would find for a single point mass with no angular momentum input), but the $1/4$ power law in the angular momentum term ensures that the approach to a steady state is extremely slow. Running our codes for about 40,000 binary orbits would be necessary to achieve a value within 10% of the asymptotic value. In practical terms, the binary torques can change the resultant disk accretion rate over very long timescales.

To summarize, the time-averaged accretion rate onto the binary asymptotes toward a value identical to the value for a single point mass, but the convergence toward this rate is slow and there can be significant enhancement in the accretion rate (by up to a factor of 2) for many viscous times. Nevertheless, even though the accretion rate has not leveled out, the disk has reached a “relaxed” state; for example, the ratio of torque to

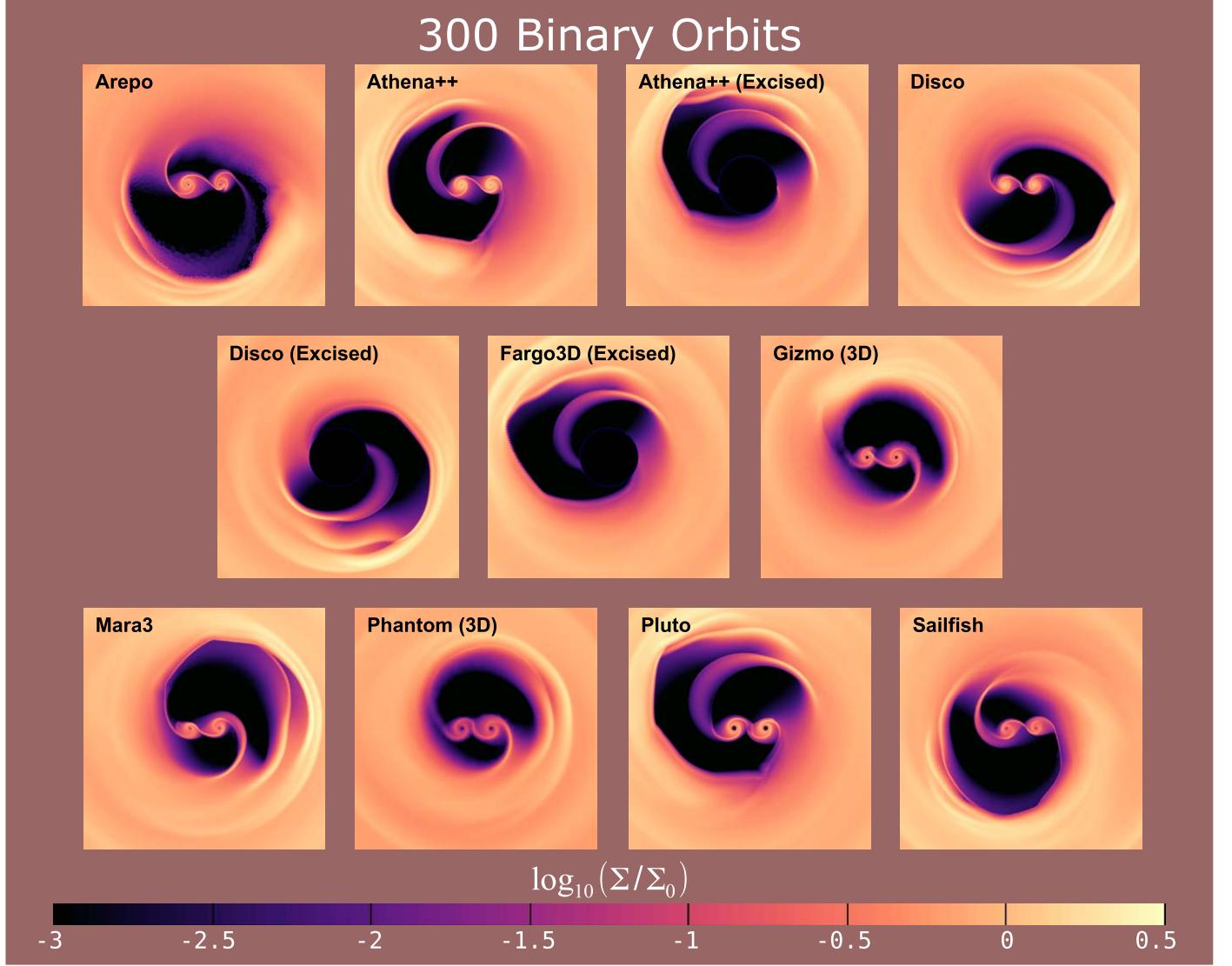


Figure 16. Surface density for all codes at 300 orbits. Codes are compared with similar zone spacing $\Delta x = 0.02a$ at $r = 3a$. At this stage, the asymmetry and size of the cavity have reached their peak, and we see the saturated state of the instability. At this point, the cavity precesses and the lump of gas responsible for accretion viability orbits the edge of the cavity, but otherwise the disk has reached a “relaxed” state—still time dependent and therefore not in a steady state, but arguably in a quasi-steady state.

accretion rate has reached a steady constant value, as we will show in Section 6.3.

6. Dependence on Numerical Choices

We have already discussed a few discrepancies in some of the measured diagnostics between codes. We now assess whether these discrepancies are due to numerical choices. If so, what controls them, and which (if any) method is more accurate? Additionally, we would like to test the effects of additional numerical choices, such as the gravitational softening radius, the live binary orbit, and other choices implicitly made by our codes.

6.1. 2D versus 3D

The dimensionality of our disk is arguably not a numerical choice but a choice of what problem to solve. Nevertheless, some codes are tailored to run in 3D and have therefore

performed a 3D version of this problem. Here we investigate what effect this has on our computed diagnostics.

Our measurements of the gravitational torque show significant discrepancies. In particular, the particle-based codes `Gizmo` and `Phantom` show a significantly lower torque than the other codes (though `PLUTO` also shows a smaller torque than the others). Interestingly, `Gizmo` and `Phantom` also find a much faster cavity precession rate. Notably, `Gizmo` and `Phantom` are solved 3D versions of the problem, whereas the other codes are being run in 2D. Hence, any discrepancy could easily be explained away by arguing that these codes are solving a different problem than the others. Unfortunately, that would not teach us much about how these codes compare with each other. Thus, we have performed an experiment where a grid-based code (`Athena++`) is run in 3D and a particle-based code (`Gizmo`) is run in 2D, to determine which of these discrepancies can be explained purely by the dimensionality of the system.

We first investigate the cavity precession rate. We find strong evidence that the precession rate is very different in 3D

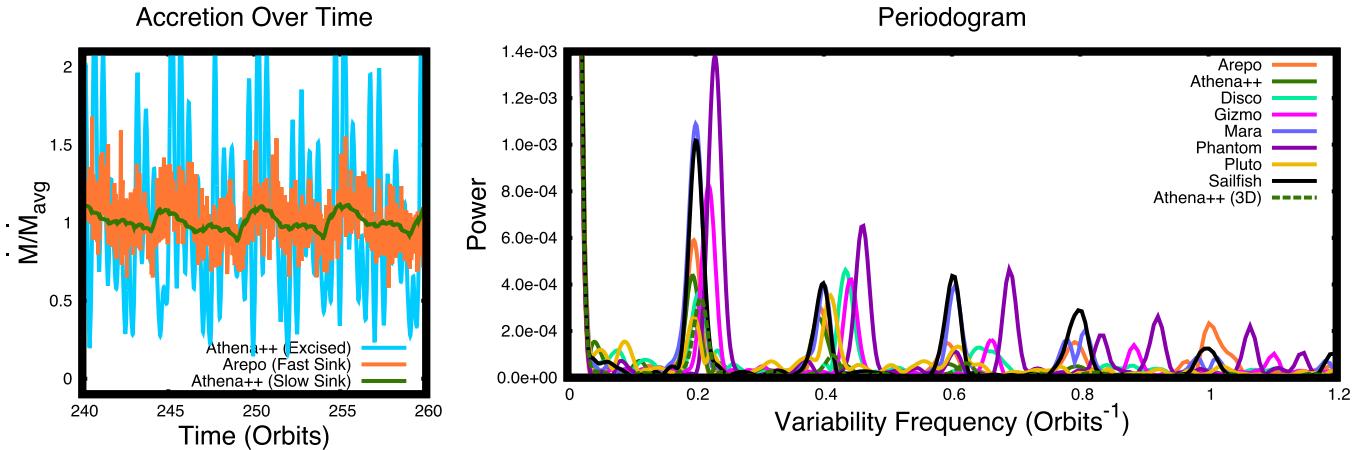


Figure 17. Accretion rate periodograms for a subset of the codes in our study. The choice of sink prescription or boundary condition can alter predictions for variability on different timescales. This is related to how much minidisk buffering is expected for a given inner truncation radius of the minidisk. Short-timescale variability will depend on this choice; it is possible that eccentric binaries might have variability less dependent on the sink or boundary choice. The left panel demonstrates what this looks like in the form of a time series; the excised Athena++ run (cyan) was included just to demonstrate the most extreme version of this effect. As the cavity was excised in that run, there can be no minidisk buffering at all.

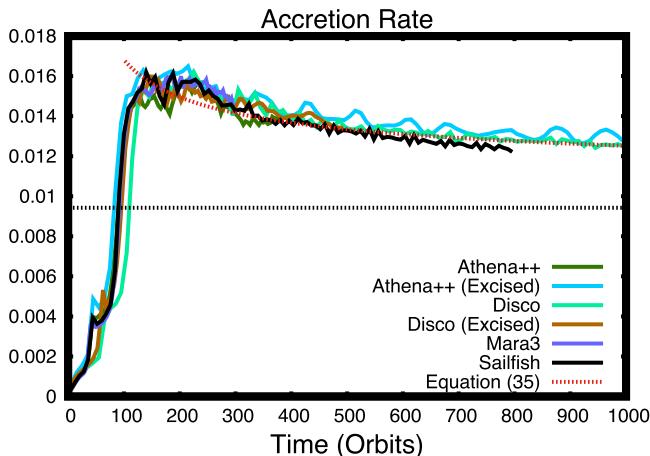


Figure 18. Comparison of the accretion rate reported by codes with infinite disks and the asymptotic value for accretion onto a single point mass (dashed gray line). \dot{M} matches the analytical expression (35), which very slowly asymptotes to the single-mass value after many thousands of orbits. In this transient period, the accretion rate is enhanced by the presence of the binary.

than in 2D, sufficient to explain entirely the discrepancy seen earlier. Athena++ finds a precession rate of $\Omega_c/\Omega_B \approx 0.008$ (125 orbit period) in 3D, whereas at the same resolution in 2D the precession rate was found to be $\Omega_c/\Omega_B \approx 0.004$, consistent with the converged value in 2D. So 3D gives a precession rate that is faster by about a factor of 2. This was similarly tested by running *Gizmo* in 2D. In 3D, the converged precession rate was $\Omega_c/\Omega_B \approx 0.007$ for *Gizmo*. In 2D, we found a slower precession rate of $\Omega_c/\Omega_B \approx 0.004$, in agreement with the other codes.

The simplest explanation for the difference in precession rate between 2D and 3D follows from the smaller cavity sizes observed in the 3D simulations, which cause the circumbinary disk to experience a more non-Keplerian potential and thus precess more rapidly (e.g., Muñoz & Lithwick 2020). However, in 3D there is an additional pressure term in the evolution equation for the disk eccentricity, which can potentially provide a comparable contribution to the precession rate (e.g., Goodchild & Ogilvie 2006; Teyssandier & Ogilvie 2016). This

discrepancy in the precession rate between 2D and 3D is likely some combination of the two effects.

A more complicated issue is the torque. *Gizmo* and *Phantom* exhibited a significantly lower gravitational torque than the other codes: low enough that one might expect the binary to migrate inward, rather than outward. Moody et al. (2019; using the grid-based Athena++ code) also found smaller torques in 3D, but the discrepancy was not as large; Moody et al. (2019) predict outward migration in both 2D and 3D.

We can shed some light on this puzzle by separating the torque into two components: an “excised torque” that adds up contributions with $r > a$ and a “minidisk torque,” i.e., the remaining component, from the minidisks themselves ($r < a$). When comparing 2D and 3D torques from Athena++ and *Gizmo*, we find that the *excised* torque ($r > a$) is significantly different between 2D and 3D, enough to explain a large portion of the discrepancy between codes. However, the *minidisk* torque ($r < a$) does not show a large discrepancy between 2D and 3D, and instead the spread between codes is much smaller (this remaining discrepancy is caused by the differences in sink prescription; see Section 6.2). Thus, we can reasonably confirm that all codes correctly compute a converged excised torque that agrees between codes, but there is a different answer in 2D than in 3D, a difference of about $0.16\dot{M}a^2\Omega_B$, which was noted in Moody et al. (2019).

We find a remaining significant discrepancy in the minidisk torque, but here there is no major dividing line between codes—as seen in the bottom panel of Figure 15, every code seems to converge to a different minidisk torque, with a general spread of about $0.5\dot{M}a^2\Omega_B$. Our next step is to investigate whether this remaining discrepancy could be caused by discrepancies in our sink prescriptions.

6.2. Sink Prescription

After accounting for the differences between 2D and 3D versions of the same problem, a (smaller but finite) discrepancy remains between different codes (see the bottom panel of Figure 15). For codes running the same number of dimensions, the biggest difference in our methods is the sink prescription, which varies significantly between codes. It would be a major practical challenge to implement the same sink prescription

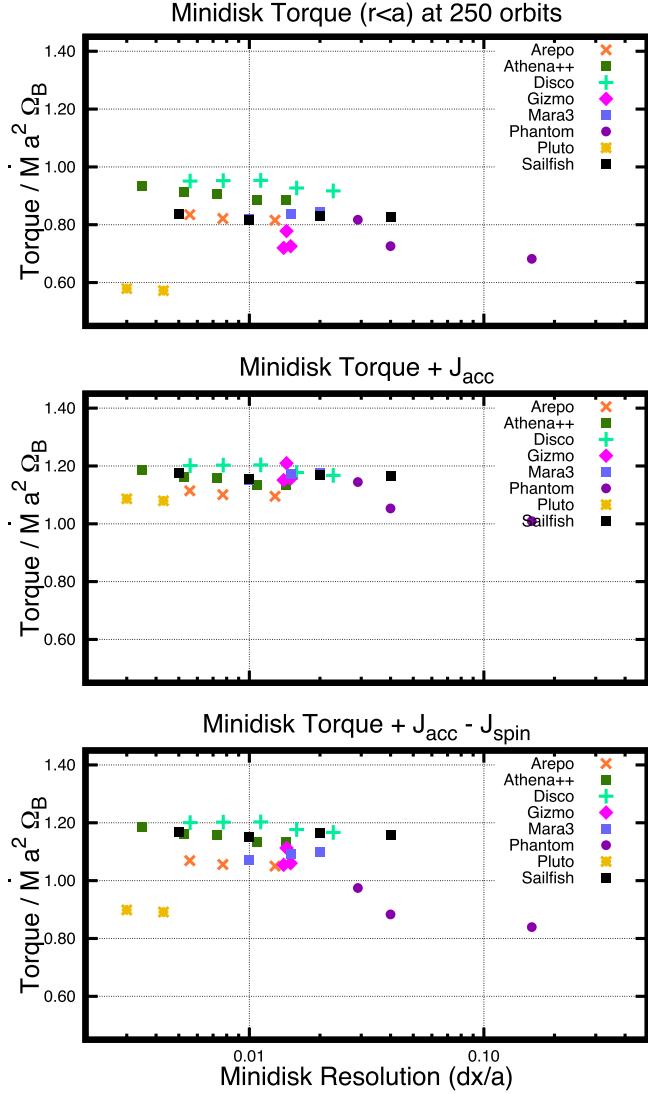


Figure 19. Torques after being corrected by the sink term \dot{J}_{acc} . While different codes disagree on the value of the minidisk torque (T_{grav} within $r < a$; top panel) and disagree on the angular momentum eaten by their sink prescription (\dot{J}_{acc}), we find the sum $T_{\text{grav}} + \dot{J}_{\text{acc}}$ to converge to the same value for all codes (shown in the middle panel). Note that if we only include the “orbital angular momentum” accreted ($\dot{J}_{\text{acc}} - \dot{J}_{\text{spin}}$; bottom panel), the discrepancy persists. Thus, to compute an orbital evolution of the binary that is consistent between all codes, one must simply take all angular momentum accreted by the sink (whether spin or orbital angular momentum) and give it to the orbital angular momentum of the binary.

across all codes; particle-based codes tend to remove gas instantaneously after a particle satisfies some conditions, whereas grid-based codes often employ a sink term to the equations, removing gas on a finite timescale. Ideally our solutions would not depend on these choices at all, so the fact that different codes use different sinks is a concern that we address and quantify here.

After some experimentation, we found that our measure for the torque (T_{grav} ; Equation (11)) i.e., the moment arm crossed with gravitational force integrated over all fluid elements, is sink dependent (if sink particles are not treated carefully; Dittmann & Ryan 2021). The sink dependence is because the total torque on the binary is partitioned between gravitational and accretion torques. When angular momentum is removed from the system by the sink, that angular momentum is no

longer available to be exchanged with the binary via gravitational torques, and therefore it reduces the measured gravitational torque.

In the limit of very small sinks, the spin angular momentum removed, \dot{J}_{spin} , should be zero, and the orbital angular momentum removed should be $\dot{J}_{\text{orb}} = \sum_i \dot{M}_i j_i$, or $0.25 \dot{M} a^2 \Omega_B$ for the equal-mass binaries studied here. However, for finite-sized sinks (as all codes require) this can only be the case when using a careful sink treatment (e.g., Dempsey et al. 2020; Dittmann & Ryan 2021).

We found that the different codes in our study with different sink prescriptions removed different amounts of both orbital and spin angular momentum from the system, leading to different predictions for the gravitational torque. It is also true in physical systems such as black hole accretion that there will be a real \dot{J}_{spin} , due to spin angular momentum removed at the ISCO, but for the purposes of this study we consider the ISCO to be at a much smaller radius than we are able to resolve. However, this shows that such effects in real physical systems may be capable of altering the sign of the torque (by stealing angular momentum from the system and giving it to the spin of the binary instead of the orbit).

That sinks could artificially change the measured gravitational torque (sufficiently to change its sign) was first noted by Tang et al. (2017). On the other hand, Muñoz et al. (2019) computed the total angular momentum given to the binary by measuring the flow rate of angular momentum through the disk in a steady state and argued that this measurement should be independent of the details of the sink prescription.

Many of the codes in our study are not set up to compute the angular momentum flux through the disk. Nevertheless, we can compute the total angular momentum given to the binary by adding the gravitational torque to the accreted angular momentum per unit time (both spin and orbital contributions). The hope is that while \dot{J}_{acc} and T_{grav} are sink dependent, their sum might not depend on the sink.

In order to compute these diagnostics, we must define \dot{J}_{acc} in terms of the sink term $\dot{\Sigma}_{\text{sink}}(r, t)$. Assume that the hydrodynamics equations can be expressed as

$$\partial_t \Sigma + \nabla_i (\Sigma v_i) = -\dot{\Sigma}_{\text{sink}} \quad (36)$$

$$\partial_t (\Sigma v_j) + \nabla_i (\Sigma v_i v_j + P \delta_{ij}) = -\Sigma \nabla_i \Phi_{\text{grav}} - \dot{\Sigma}_{\text{sink}} v_* \quad (37)$$

Then, the sink term $\dot{\Sigma}_{\text{sink}}$ contains information about the rate of removal of mass and angular momentum from the system,

$$\dot{M} = \int \dot{\Sigma}_{\text{sink}} dA, \quad (38)$$

$$\dot{J}_{\text{acc}} = \int \dot{\Sigma}_{\text{sink}} r \times v_* dA, \quad (39)$$

where v_* is the velocity associated with the removed mass and momentum given above. For “standard” sinks, gas is removed at the velocity of the fluid element, so $v_* = v_{\text{hydro}}$. However, for alternative sink methods, such as the “torque-free” sink in Athena++ and in Disco (Dempsey et al. 2020; Dittmann & Ryan 2021), the velocity v_* is chosen to avoid introducing spurious torques to the system. For Athena++, this is accomplished by boosting to the frame of the sink and subtracting off the spin component of v_* , i.e., only keeping a radial component in the frame of the sink. For Disco (in this study), this is accomplished in a simpler manner, by setting $v_* = v_{\text{sink}}$, so that $v_* = 0$ in the frame of the sink (although this also affects the radial component of the velocity near each

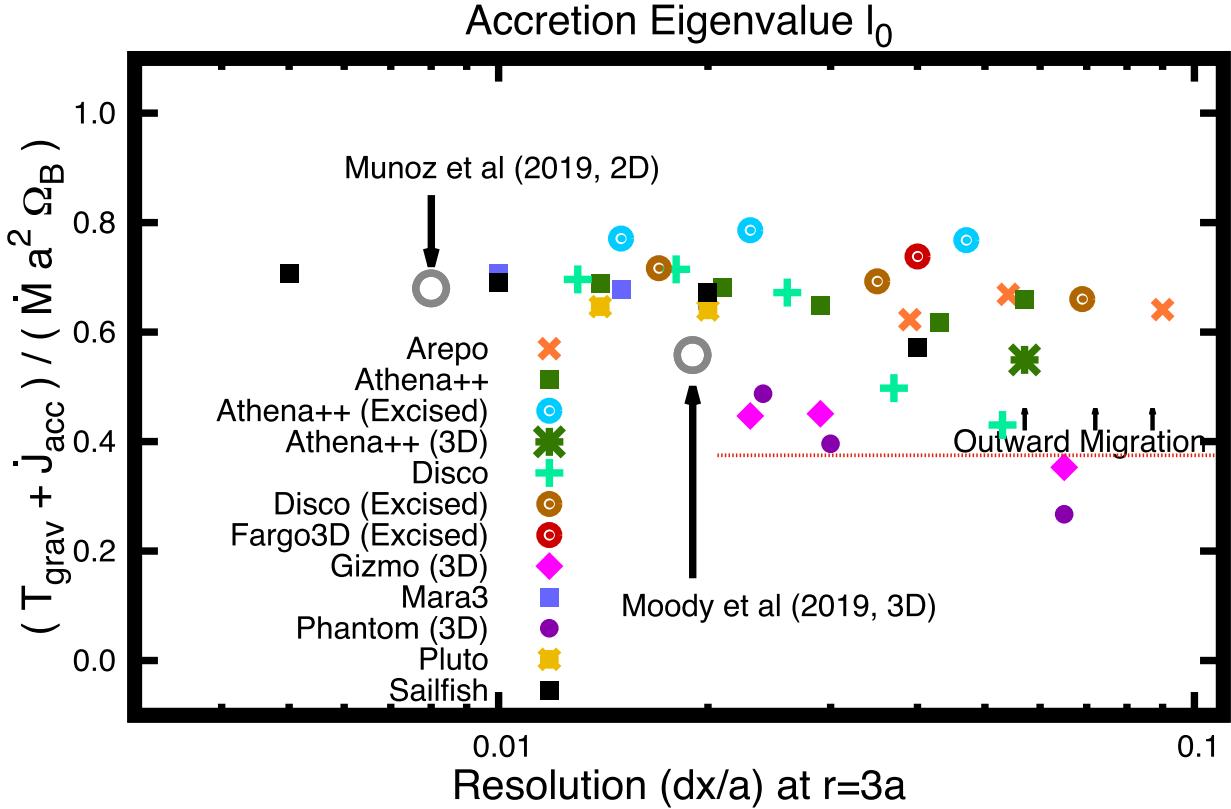


Figure 20. Final comparison between all codes on the total angular momentum delivered to the binary, after including the sink term j_{acc} . Vastly better agreement is found between codes after accounting for this contribution. 2D codes consistently converge toward an eigenvalue of $l_0 \approx 0.7$.

Table 2
Sink Prescriptions for Nonexcised Codes

Code	γ/Ω_B	R_S/a	Type	j_{acc}/\dot{M}	j_{spin}/\dot{M}
Arepo	∞^a	0.03	Standard	0.2794	0.04489
Athena++	1.33	0.05	Torque-Free	0.24996	10^{-20}
Disco	1	0.05	Torque-Free	0.2503	0.00075
Gizmo	∞^b	0.05	Standard	0.4311	0.09650
Mara3	10	0.05	Standard	0.3332	0.08096
Phantom	∞^b	0.05	Standard	0.3273	0.17258
PLUTO	10^3	0.075	Standard	0.5073	0.18770
Sailfish	10	0.05	Standard	0.3395	0.00602

Notes. *Fargo3D* was not included in this table, as this method excised the binary for all runs.

^a *Arepo* removed a fraction (up to 50% nearest the sink particle) of the mass within each cell within the sink region every time step, resulting in a comparatively large but time-dependent sink rate.

^b *Gizmo* and *Phantom* removed particles within the sink region instantaneously.

sink). For the standard sinks employed in most codes, v_* is the local fluid velocity. Note that particle-based codes accrete gas particles discretely, and in this study v_* is simply the velocity of the removed particle, i.e., the local fluid velocity.

Additionally, we have computed a separate quantity, the accreted spin,

$$j_{\text{spin}} \equiv \int \dot{\Sigma}_{\text{sink}} (r - r_s) \times (v_* - v_s) dA, \quad (40)$$

where r_s and v_s are the position and the velocity of the sink particle. One can then define the accreted *orbital* angular

momentum:

$$j_{\text{orb}} \equiv j_{\text{acc}} - j_{\text{spin}}. \quad (41)$$

Figure 19 shows the minidisk torque in all our (nonexcised) codes, after correcting for the sink term j_{acc} . The top panel shows the raw minidisk torque (gravitational torque measured within $r < a$) in units of $\dot{M}a^2\Omega_B$. The middle panel shows the minidisk torque plus the total angular momentum accreted, $T_{\text{mini}} + j_{\text{acc}}$, again in units of $\dot{M}a^2\Omega_B$, showing much better agreement between codes. Note that this agreement disappears if one only includes $j_{\text{orb}} = j_{\text{acc}} - j_{\text{spin}}$.

How should we interpret this result? Because each code handled accretion differently, the spin, accretion, and gravitational torques measured in each code depend on these numerical parameters (see Table 2). However, the angular momentum current through the disk is largely set by viscosity and the gravitational influence of the binary on the disk. Then, thanks to Newton's third law and the conservation of angular momentum, this angular momentum current through the disk directly determines the total torque on the binary. Thus, while various codes find discrepant values of the gravitational, accretion, and spin torques, the measured values of the accretion eigenvalue l_0 tend to be in good agreement, regardless of accretion prescriptions or whether the binary was on the grid or excised.

Figure 20 plots the accretion eigenvalue l_0 (total angular momentum given to the binary in units of $\dot{M}a^2\Omega_B$) for all codes at all resolutions in our study. We find that all 2D codes converge toward $l_0 \approx 0.7$ and that 3D codes converge toward a smaller value, $l_0 \approx 0.5$. All of this is consistent with previous studies, which have found $l_0 = 0.68$ in 2D (Muñoz et al. 2019)

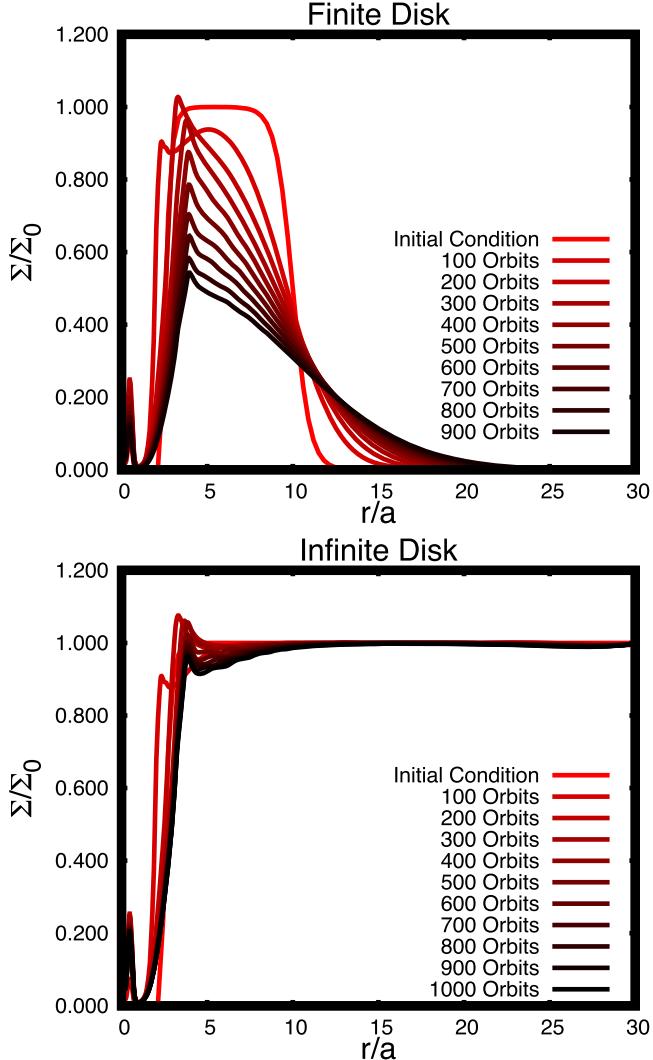


Figure 21. Evolution of our two disk models over time in the *Disco* code. During the first 100 orbits or so, the finite and infinite disks exhibit similar surface density. However, as time evolves, the finite disk accretes mass and spreads, reducing the surface density to half its original value over 1000 orbits. The infinite disk reaches a reasonably steady configuration at $\Sigma \sim \Sigma_0$. At late times, the disk structure is completely different, but the torque (when normalized to \dot{M}) is nearly identical, as shown in Figure 22.

and $l_0 = 0.56$ in 3D (Moody et al. 2019). All 11 numerical techniques find outward migration for the binary in the relaxed state of the disk, so long as the code’s resolution is sufficiently high. In practice, this requires a resolution of around $\Delta x \lesssim 0.01a$ in the vicinity of the minidisks and a resolution of $\Delta x \lesssim 0.03a$ at $r = 3a$ for grid codes, while particle-based codes require higher resolution (by this metric) to achieve convergence.

6.3. Finite versus Infinite Disk

The effect of a finite disk has already been investigated by Muñoz et al. (2020). Nevertheless, we can empirically demonstrate any difference between a finite and infinite disk, by comparing the two choices for $f(r)$ in Equation (3). Surface density evolution of the two different models is plotted in Figure 21. Torque as a function of time for the two models is plotted in Figure 22. When one waits for a sufficiently long time, the torque and accretion rates are reduced in the finite-

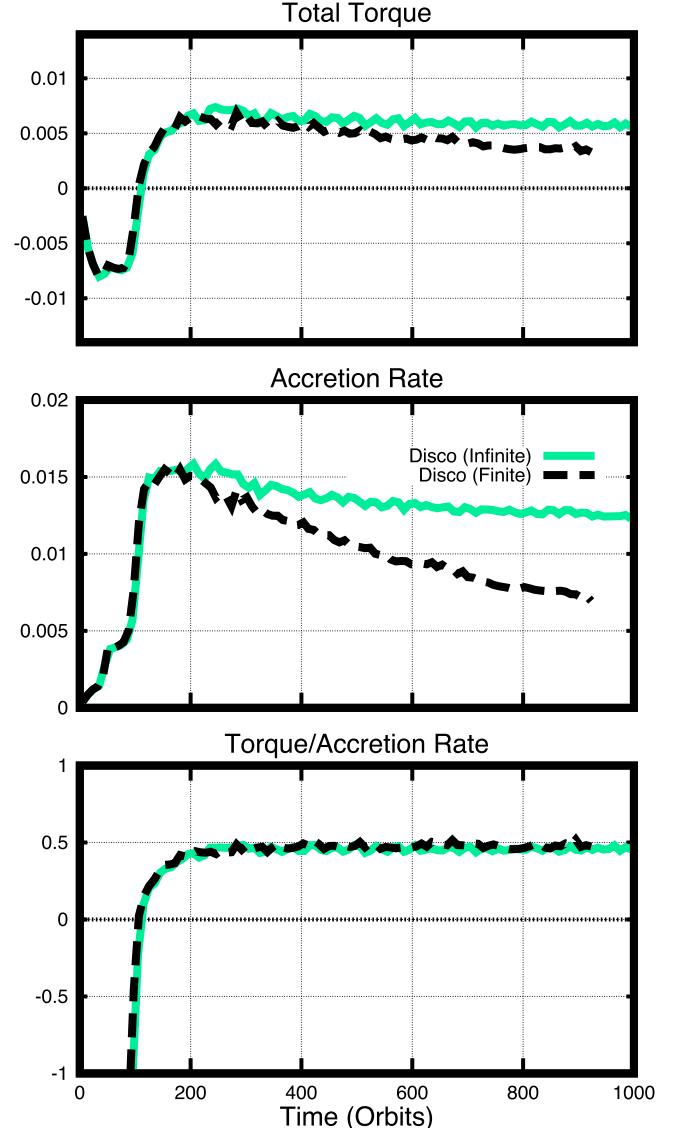


Figure 22. Finite and Infinite disk models compared in the *Disco* code over 1000 orbits. As shown in Figure 21, the disk structure at 1000 orbits is completely different, such that the torque and accretion rate are about half as large in the finite disk as they are in the infinite disk. However, their ratio T/\dot{M} is identical for the finite and infinite disk; this ratio does not appear to depend on disk structure at all, though a more systematic investigation of different disk models might be able to demonstrate this explicitly.

disk case, due to the fact that the disk mass is being depleted and the disk is spreading. However, even in this case, if one plots the ratio of torque to accretion rate, one achieves an identical answer between finite and infinite disks. Therefore, both models provide an equivalent measurement of the torque.

6.4. Live versus Fixed Binary

Most of these calculations employ a binary on a fixed circular orbit, assuming the limiting case where the binary orbital evolution is slow compared to the relaxation time of the disk. Such a choice is only valid in the limit of a low disk mass, akin to a test particle in orbital dynamics. If one wishes to follow the interaction of a binary with a substantially massive disk, the binary orbit should react accordingly. We measure whether this effect is significant for low disk mass by analyzing orbital evolution in the *Phantom* code run, where the orbit is

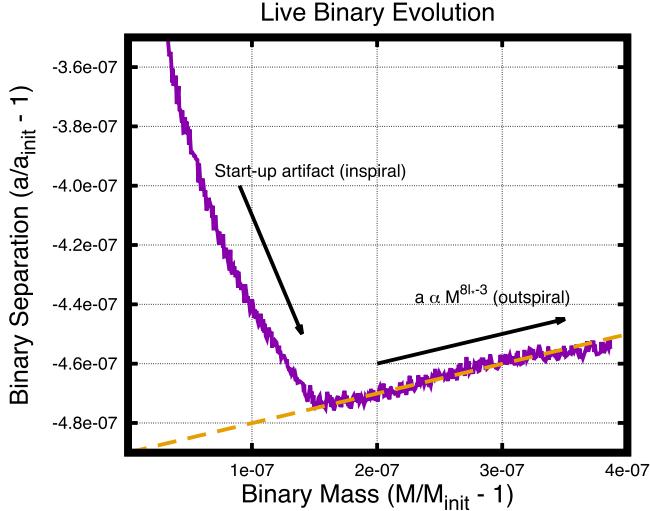


Figure 23. Employing a “live binary” in the Phantom code. For sufficiently low surface density Σ_0 , the impact of the live binary is negligible and the binary migrates according to Equation (46) (orange dashed line), with $l_* \approx 0.39$ consistent with the measured torques. Additionally, there is an artificial start-up transient consistent with the negative torques seen in the first 150 orbits, highlighting the need for long-duration computations.

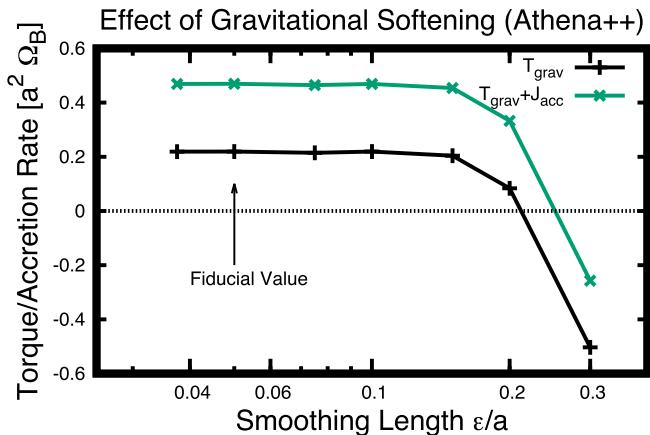


Figure 24. Gravitational softening was varied in the Athena++ code. The value of ϵ represents the length scale being smoothed over. This demonstrates that our fiducial value $\epsilon = 0.05a$ is much smaller than necessary to give converged torques. The criterion turns out to be $\epsilon \lesssim 0.15a$. This criterion would likely vary with the Mach number, mass ratio, or eccentricity of the binary.

not fixed a priori but the disk mass is assumed to be negligible, i.e., $M_d/M = 5 \times 10^{-6}$, in order to minimize the change in binary orbit due to the interaction with the disk.

For a circular equal-mass binary, the orbital angular momentum is

$$J = 0.25Ma^2\Omega_B = 0.25M\sqrt{GMa}. \quad (42)$$

As the orbit and mass of the binary change, one can compute $J = l_0\dot{M}a^2\Omega_B$ in terms of \dot{a} and \dot{M} ,

$$\frac{J}{\dot{M}} = \frac{3}{2} \frac{\dot{M}}{M} + \frac{1}{2} \frac{\dot{a}}{a} = 4l_0 \frac{\dot{M}}{M}. \quad (43)$$

This reduces to an expression for \dot{a} in terms of \dot{M} ,

$$\frac{\dot{a}}{a} = (8l_0 - 3) \frac{\dot{M}}{M}, \quad (44)$$

which can be solved to determine the separation as a function of mass $a(M)$,

$$a \propto M^{8l_0-3}. \quad (45)$$

In general, we expect our assumption to neglect variations of the binary orbit not to be valid when the disk mass is significant in comparison to the binary mass (see, e.g., Franchini et al. 2023; Tiede et al. 2024, for a discussion). In particular, at the resolution tested, with an eigenvalue $l_0 = 0.39$, we obtain the scaling $a(M) \propto M^{0.1}$, consistent with the numerical result by Phantom, as shown in Figure 23.

Note that an artificial period of inspiral occurs (consistent with the negative torque seen in all codes for the first 100 orbits). This highlights the need for long-duration computations to ensure that such artificial start-up transients do not dominate the solution. This is important because in the first 100 orbits the solution should depend on our choice of initial density profile, whereas the late-time behavior does not. For example, Figures 21 and 22 show that very different disk models give nearly identical accretion eigenvalues l_0 in a steady state; this was also shown previously by Muñoz et al. (2020).

It should also be noted that the live binary setup in Phantom only includes the orbital angular momentum accreted, \dot{J}_{orb} , not the total accreted angular momentum—i.e., it neglects \dot{J}_{spin} , assuming that this term does not contribute to the total angular momentum of the binary (e.g., assuming that the point mass is a black hole and the spin is accreted and spins up the hole). Thus, for the live binary setup of Phantom, the expansion of the binary will follow

$$a \propto M^{8l_0-3}, \quad (46)$$

where

$$l_* = \frac{T_{\text{grav}} + \dot{J}_{\text{orb}}}{\dot{M}a^2\Omega_B}, \quad (47)$$

which explains the somewhat slower expansion rate than one would expect by including both orbital and spin angular momentum accreted. Nevertheless, the migration of the binary orbit does not affect the measured torques or accretion rates, and the migration is consistent with the gravitational torque combined with the accreted orbital angular momentum term.

6.5. Gravitational Softening

One approximation that was made for the sake of uniformity across codes was the form of the gravitational potential,

$$\phi_i = \frac{GM_i}{\sqrt{(\vec{r} - \vec{r}_i)^2 + \epsilon^2}}, \quad (48)$$

with $\epsilon = 0.05$. As a simple test of how much this affects our solutions, we ran this same problem using the Athena++ code using a range of values of ϵ . The results of this series of calculations appear in Figure 24. We find that for sufficiently small $\epsilon \lesssim 0.15a$ our measured torque is independent of the choice of ϵ , suggesting that the gravitational softening does not affect our results. This also suggests that the precise form of the potential did not matter in the vicinity of the point masses (as some studies favor cuspier potential forms, e.g., using spline softening). A softening of $\epsilon \sim 0.15a$ is quite large, comparable to the size of the minidisks. We did not test the eccentric case, but it is likely that this nice “plateau” behavior would occur at

smaller scales if the binary were of unequal mass or on an eccentric orbit, or orbiting retrograde to the disk (see softening studies in Dittmann & Ryan 2024; Tiede & D’Orazio 2024), so we can only say what value and form of the potential are sufficient for this specific test problem. For example, the value of ϵ found in this study is sufficiently small that softening is negligible beyond the Roche lobe of each sink; naturally, this scale would become smaller near the pericenter of an eccentric orbit or around lower-mass objects. Regardless, for this test problem, our softening length is sufficiently small and, as shown in Dittmann & Ryan (2021), the precise form of the potential is immaterial.

7. Conclusions

We have developed a straightforward code test that exhibits the basic fundamental features of the binary–disk problem. We have tested 11 very different numerical schemes on this same problem, to assess code performance and to test whether very different codes can agree on the solution to the same underlying physics problem. Our code outputs have been made public on a static Zenodo repository at doi:[10.5281/zenodo.10557951](https://doi.org/10.5281/zenodo.10557951) (Duffell 2024) and online.²⁶ Although there are some expected differences between 2D and 3D setups, overall we find general agreement between codes, with small deviations that we believe can be accounted for or understood.

Over the first 1–10 orbits, all codes can achieve high-precision convergence (this includes both excised runs and calculations performed in 3D), and this work can therefore provide a convenient code test for checking one’s initial setup, diagnostics, and code convergence.

Over the first 100 orbits, the disk becomes eccentric, and agreement between codes becomes more difficult to determine. We find agreement in the growth rate of the instability $\Gamma_c/\Omega_B \approx 0.012$ and the precession rate of the cavity $\Omega_c/\Omega_B \approx 0.004$. Here is where we first see differences between 2D and 3D, as the cavity is smaller in 3D and the precession rate is faster (more like 0.045 per binary period). The instability growth rate, however, seems to be comparable between 2D and 3D.

For the problem studied here, torque saturates to a steady state in a few hundred orbits. At this time, \dot{J}/\dot{M} is independent of the disk size (as also shown previously by Muñoz et al. 2020), as demonstrated by Figures 21 and 22. This suggests that this code test can be extended to more complicated and disk models. The final average gravitational torque was $T_{\text{grav}} \approx 0.5\dot{M}a^2\Omega_B$, albeit with significant discrepancies between codes.

The biggest contribution to this discrepancy was the fact that some codes were run in 2D, while others simulated these models in 3D. By running Athena++ in 3D and Gizmo in 2D, we were able to determine how much of the discrepancy was due to the dimensionality of the problem. This accounts for a discrepancy of $\delta T \approx 0.16\dot{M}a^2\Omega_B$, with 3D calculations experiencing a lower (but still positive) torque. This difference between 2D and 3D torque is consistent with the difference found by Moody et al. (2019).

After accounting for this discrepancy, smaller discrepancies remained between codes, which were caused by the different sink prescriptions employed. However, the total angular momentum current through the disk should be largely independent of most sink choices, and thus the total torque on

the binary should not depend on the sink prescription. This also leads to good agreement between excised codes and those that simulate the binary itself. This also shows that, in a real physical system, if the point masses remove angular momentum from the disk and contribute it to the spin (e.g., if the black hole ISCO is sufficiently large), this could remove angular momentum from the circumbinary disk without giving it to the binary’s orbit, affecting the rate or direction of an object’s migration.

After accounting for both dimension and sink discrepancies, there are still (much smaller) deviations in the torque between codes, which could reflect how precisely these codes are able to converge. Nevertheless, all codes are sufficient to predict the direction of migration.

For this test problem, all codes found outward migration. Most codes fixed the binary orbit and mass, but some used a live binary, which obeyed the scaling

$$a \propto M^{8l_0-3}, \quad (49)$$

consistent with the angular momentum eigenvalue

$$l_0 = \frac{T_{\text{grav}} + \dot{J}_{\text{acc}}}{\dot{M}a^2\Omega_B}. \quad (50)$$

Thus, for sufficiently low disk mass, including the live reaction of the binary orbit to the gas forces is not vital for determining the orbital evolution of the system. This can be understood in the limit where the disk is modeled as a “test fluid” and hence the orbital evolution of the binary is sufficiently slow that it can be well approximated by a fixed orbit.

Overall, we found 2D codes converged to an eigenvalue of $l_0 \approx 0.7$ (with 3D codes converging to a lower value, $l_0 \approx 0.5$) and that this eigenvalue was very consistent across all 2D codes, so long as accreted angular momentum was accounted for. Codes that excised the binary were able to recover this eigenvalue so long as the angular momentum flow through the excision radius was accounted for.

Acknowledgments

This research was supported in part by grant NSF PHY-1748958 to the Kavli Institute for Theoretical Physics (KITP). Foundational meetings initiating this research took place at the KITP, during the program “Bridging the Gap: Accretion and Orbital Evolution in Stellar and Black Hole Binaries” (<https://www.kitp.ucsb.edu/activities/binary22>).

P.D. is supported by the National Science Foundation under grant No. AAG-2206299. A.J.D. acknowledges the University of Maryland supercomputing resources (<http://hpcc.umd.edu>) that were made available for conducting the research reported in this paper and the ASTRA computing cluster maintained by the Department of Astronomy at the University of Maryland. A.J.D. was supported in part by NASA ADAP grant 80NSSC21K0649. D.J.D. received funding from the European Union’s Horizon 2020 research and innovation program under Marie Skłodowska-Curie grant agreement No. 101029157 and from the Danish Independent Research Fund through Sapere Aude Starting grant No. 121587. The Tycho supercomputer hosted at the SCIENCE HPC center at the University of Copenhagen was used for supporting this work. A.F. acknowledges financial support provided under the European Union’s H2020 ERC Consolidator Grant “Binary Massive Black Hole Astrophysics” (B Massive, Grant Agreement: 818691). E.R. thanks Daniel Price, Guillaume Laibe, and Elliot Lynch for fruitful discussions and acknowledges financial support from

²⁶ <https://www.physics.purdue.edu/duffell/data.html>

the European Union’s Horizon Europe research and innovation program under the Marie Skłodowska-Curie grant agreement No. 101102964 (ORBIT-D) and from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation program (grant agreement No. 864965, PODCAST). J.Z. acknowledges support from the LISA Preparatory Science Program (LPS) through NASA award No. 80-NSSC-24K0440.

Numerical calculations were performed using many different computing facilities, including the Petunia cluster at Purdue University. Sailfish simulations were run on the Palmetto cluster at Clemson University. Phantom simulations were performed using the DiRAC Data Intensive service at Leicester, operated by the University of Leicester IT Services, which forms part of the STFC DiRAC HPC Facility (www.dirac.ac.uk). The equipment was funded by BEIS capital funding via STFC capital grants ST/K000373/1 and ST/R002363/1 and STFC DiRAC Operations grant ST/R001014/1. DiRAC is part of the National e-Infrastructure.

Research presented in this article was supported by the Laboratory Directed Research and Development program of Los Alamos National Laboratory under project No. 20220087DR. This research used resources provided by the Los Alamos National Laboratory Institutional Computing Program, which is supported by the US Department of Energy National Nuclear Security Administration under contract No. 89233218CNA000001. The LA-UR number is LA-UR-23-26505.

Research at Perimeter Institute is supported in part by the Government of Canada through the Department of Innovation, Science and Economic Development and by the Province of Ontario through the Ministry of Colleges and Universities.

PLUTO simulations were performed with the support of the High Performance and Cloud Computing Group at the Zentrum für Datenverarbeitung of the University of Tübingen, the state of Baden-Württemberg through bwHPC, and the German Research Foundation (DFG) through grant No. INST 37/935-1 FUGG. A.B.T.P. acknowledges support from the Royal Society in the form of a University Research Fellowship and Enhanced Expenses Award and grant 285676328 of the German Research Foundation (DFG).

Finally, we would like to thank the anonymous referee for the very thoughtful and helpful report.

Appendix Code Descriptions

In the following, we provide details for each code and the various algorithmic choices employed in this work.

A.1. Arepo

`Arepo` (Springel 2010) is a moving-mesh hydrodynamics code based on a dynamic Voronoi tessellation of the computational domain performed each time step. `Arepo` was originally designed for cosmological simulations but has been adapted for the solution of the Navier–Stokes equations (Muñoz et al. 2013) and the study of astrophysical disks (e.g., Muñoz & Lai 2016), especially protoplanetary disks and circumbinary disks. The mesh motion helps by performing hydrodynamic calculations in the frame of the otherwise supersonic (Mach 10 in this study) flow. Additionally, `Arepo`’s resolution is naturally adaptive in nature, resulting in much higher resolution in the minidisks surrounding each

point mass than farther out in the disk (the zone size near each binary component is nearly an order of magnitude smaller than it is at a radius of a few times the binary separation).

A.2. Athena++

`Athena++` is a finite-volume code, supporting Cartesian, cylindrical, and spherical geometries in addition to both static and adaptive mesh refinement (Stone et al. 2020). `Athena++` is a rewrite of the `Athena` code (Stone et al. 2008) using C++, which introduced new features such as block-based adaptive mesh refinement (e.g., Stout et al. 1997) and dynamic task scheduling. In this work, `Athena++` has been used to solve the equations of viscous hydrodynamics in both 2D and 3D, using both Cartesian coordinates with the binary on the grid and cylindrical coordinates with an excised binary.

The 2D Cartesian simulations used the Roe approximate Riemann solver (Roe 1981), the third-order piecewise parabolic spatial reconstruction (Felker & Stone 2018), and the second-order van Leer (VL2) predictor–corrector time integrator described in Stone & Gardiner (2009). We also carried out a limited number of 3D simulations in Cartesian geometry, utilizing piecewise linear spatial reconstruction (van Leer 1974), the VL2 time integrator, and the HLLC approximate Riemann solver (Toro et al. 1994).

A.3. Disco

`Disco` is a finite-volume code that employs a moving, shearing polar grid (Duffell 2016). The numerical method is essentially the “moving-mesh” technique of `Arepo`, but rather than using Voronoi cells, `Disco` employs moving volumes that are wedgelike annular segments (as in a fixed polar grid code) that can shear azimuthally. The moving mesh reduces advection errors as it does in `Arepo`, but the smooth shearing of the zones reduces numerical noise, which can be advantageous for many problems, especially where subtle or weak nonlinear effects must be evolved and measured alongside the large bulk Keplerian shear flow of the disk.

In this comparison, `Disco` was run in two configurations, one in which the grid extended to $r = 0$, and another with an excised inner region. In the former, the grid spacing was logarithmic at large radii and linear near the origin, and the azimuthal resolution was varied in each annulus to maintain a roughly constant aspect ratio. In the latter configuration, a diode boundary condition was applied at the excision radius ($r = a$, where a is the binary separation). Notably, both versions of `Disco` used in this study employed a simplified version of the viscous terms in the hydro equations (see Duffell 2016, Section 2.9 and the Appendix, for the precise equations being integrated). When compared with a version of the code that includes all viscosity terms in the equations, one finds only minor deviations for equal-mass binaries such as those studied in this comparison (see Appendix A of Dittmann & Ryan 2021), but larger deviations have been found for unequal-mass binaries (see Appendix B of Dittmann & Ryan 2024).

A.4. Fargo3D

`Fargo3D` (Benítez-Llambay & Masset 2016), the successor of the `FARGO` code, is a versatile multifluid HD/MHD code that runs on clusters of CPUs or GPUs, with special emphasis on protoplanetary disks. An important aspect of `Fargo3D`, which makes it ideal for the study of protoplanetary disks, is

the orbital advection scheme originally developed in the FARGO code (Masset 2000), for use with tracking highly supersonic orbital motions. The code’s name being *Fargo3D* may cause some confusion, as it has only been run in 2D in this study (though, as the name suggests, it is capable of 3D). Distinguishing features of *Fargo3D* include its use of a staggered grid with face-centered velocity components and an artificial von Neumann–Richtmyer artificial viscosity as in Stone & Norman (1992).

A.5. *Gizmo*

Gizmo is a multimethod and multiphysics code for hydrodynamics simulations that implements smoothed particle hydrodynamics (SPH), meshless finite volume (Gaburov & Nitadori 2011), and meshless finite mass schemes (Hopkins 2015). The last are very similar to moving-mesh methods, except that the Riemann problem is solved across smooth boundaries (“faces”) between particles instead of using the faces resulting from a Voronoi tessellation. In particular, the method employed here is the meshless finite mass one, which implicitly assumes a deformation of the face between particles such that there is no mass flux between them, effectively making the method Lagrangian.

In this study, *Gizmo* has been used coupled with adaptive particle splitting for numerical refinement of the gas dynamics inside the disk cavity (see Franchini et al. 2022, 2023, for details). The kernel size for the hydrodynamic interaction is defined by an effective number of neighbors equal to 58 (Franchini et al. 2022). The binary is modeled using two sink particles (Bate et al. 1995), each with sink radius $r_{\text{sink}} = 0.05a$ (particles within this radius are removed from the simulation).

A.6. *Mara3* and *Sailfish*

Mara3 and *Sailfish* both use a fixed-mesh second-order Godunov solver. They solve the locally isothermal and vertically averaged Navier–Stokes equations in Cartesian coordinates. *Mara3* is an older code originally configured to study relativistic MHD turbulence (Zrake & MacFadyen 2012). Features have been added specifically for simulations of the binary accretion problem, including the locally isothermal equation of state, viscous fluxes (constant- ν or constant- α), fixed-mesh refinement, test particles, and post-processing. *Mara3* was used to calculate gas-driven orbital evolution in Tiede et al. (2020) and Zrake et al. (2021) and to study kinematic aspects of binary accretion using test particles in Tiede et al. (2022). *Sailfish* is a GPU-accelerated version of *Mara3* that also includes post-processing modules to synthesize light curves of thermal disk emission. It is the code that was used in Westernacher-Schneider et al. (2022), Major Krauth et al. (2023), and Westernacher-Schneider et al. (2024). *Sailfish* is very fast ($\gtrsim 10^9$ zone updates per second for this problem on an A100 GPU), and it runs efficiently on multi-GPU nodes.

A.7. *Phantom*

Phantom is an SPH (Gingold & Monaghan 1977; Lucy 1977; Monaghan 1992) and smoothed particle magnetohydrodynamics (Price 2012) 3D code (Price et al. 2018), developed for applications in the context of stellar, galactic, planetary, and high-energy astrophysics. The code has been used widely for studies of accretion disks (e.g., Lodato & Price 2010) and turbulence (e.g., Price & Federrath 2010), from

the birth of planets (e.g., Dipierro et al. 2015) to how black holes accrete (e.g., Nealon et al. 2015; Ragusa et al. 2016), and more specifically for circumbinary disks (e.g., Facchini et al. 2013, 2019, 2021; Nixon et al. 2013; Ragusa et al. 2017; Smallwood et al. 2019; Hirsh et al. 2020).

In this work, the binary is modeled using two sink particles (Bate et al. 1995; Nixon et al. 2013; particles are removed from the simulation when they cross $r_{\text{sink}} = 0.05a$ and other criteria are met). Viscosity is calculated including the viscous stress term using two first derivatives (Price et al. 2018) and constant $\nu = 10^{-3}$, as prescribed for this code comparison setup. The artificial viscosity parameter for each particle is allowed to vary between $0.1 < \alpha_{\text{AV},i} < 1$ using the Cullen & Dehnen (2010) switch for shock capturing and $\beta_{\text{AV}} = 2$ to avoid particle interpenetration. Individual particle time stepping has been used to speed up the simulations. For this work, five simulations differing for their resolution have been performed (using 1M, 10M, 20M, 50M, and 100M particles). Among them, only those with 1M, 10M, and 20M particles reached 300 binary orbits. Results from the 50M- and 100M-particle runs will be used for diagnostics in the first 10–15 binary orbits. The resolution $\Delta x/a$ used to compare the various diagnostics has been calculated as the mode of the distribution of the particle smoothing lengths in the region of reference prescribed.

A.8. *PLUTO*

PLUTO is a finite-volume grid code developed by Mignone et al. (2007) using Cartesian, cylindrical, or spherical grids. *PLUTO* was created to solve astrophysical problems that require hydrodynamical, magnetohydrodynamical, and relativistic hydrodynamical prescriptions. The version used in this project is a GPU-capable modification by Thun et al. (2017). The numerical setup generally follows Thun et al. (2017). We use an HLL-solver and a VL2 for the simulations. The binary is modeled in a 2D polar configuration with a radially logarithmic spaced grid up to $0.3a$ and another 30 cells linearly spaced down to $0.03a$. The inner boundary reflects radial velocities and density, such that no mass is lost at the inner radius near the origin.

ORCID iDs

Paul C. Duffell  <https://orcid.org/0000-0001-7626-9629>
 Alexander J. Dittmann  <https://orcid.org/0000-0001-6157-6722>
 Daniel J. D’Orazio  <https://orcid.org/0000-0002-1271-6247>
 Alessia Franchini  <https://orcid.org/0000-0002-8400-0969>
 Kaitlin M. Kratter  <https://orcid.org/0000-0001-5253-1338>
 Anna B. T. Penzlin  <https://orcid.org/0000-0002-8873-6826>
 Enrico Ragusa  <https://orcid.org/0000-0001-5378-7749>
 Magdalena Siwek  <https://orcid.org/0000-0002-1530-9778>
 Christopher Tiede  <https://orcid.org/0000-0002-3820-2404>
 Haiyang Wang  <https://orcid.org/0000-0001-7167-6110>
 Jonathan Zrake  <https://orcid.org/0000-0002-1895-6516>
 Adam M. Dempsey  <https://orcid.org/0000-0001-8291-2625>
 Zoltan Haiman  <https://orcid.org/0000-0003-3633-5403>
 Alessandro Lupi  <https://orcid.org/0000-0001-6106-7821>
 Michal Pirog  <https://orcid.org/0000-0002-3164-5923>
 Geoffrey Ryan  <https://orcid.org/0000-0001-9068-7157>

References

Armitage, P. J., & Natarajan, P. 2002, *ApJL*, 567, L9
 Artynowicz, P., & Lubow, S. H. 1994, *ApJ*, 421, 651

Artymowicz, P., & Lubow, S. H. 1996, *ApJL*, **467**, L77

Avara, M. J., Krolik, J. H., Campanelli, M., et al. 2023, arXiv:2305.18538

Bankert, J., Krolik, J. H., & Shi, J. 2015, *ApJ*, **801**, 114

Bate, M. R., & Bonnell, I. A. 1997, *MNRAS*, **285**, 33

Bate, M. R., Bonnell, I. A., & Price, N. M. 1995, *MNRAS*, **277**, 362

Benítez-Llambay, P., & Masset, F. S. 2016, *ApJS*, **223**, 11

Bourne, M. A., Fiacconi, D., Sijacki, D., Piotrowska, J. M., & Koudmani, S. 2023, arXiv:2311.17144

Bowen, D. B., Mewes, V., Noble, S. C., et al. 2019, *ApJ*, **879**, 76

Bright, J. C., & Paschalidis, V. 2023, *MNRAS*, **520**, 392

Cimerman, N. P., & Rafikov, R. R. 2024, *MNRAS*, **528**, 2358

Combi, L., Lopez Armengol, F. G., Campanelli, M., et al. 2022, *ApJ*, **928**, 187

Cuadra, J., Armitage, P. J., Alexander, R. D., & Begelman, M. C. 2009, *MNRAS*, **393**, 1423

Cullen, L., & Dehnen, W. 2010, *MNRAS*, **408**, 669

D'Angelo, G., & Lubow, S. H. 2008, *ApJ*, **685**, 560

d'Ascoli, S., Noble, S. C., Bowen, D. B., et al. 2018, *ApJ*, **865**, 140

Dempsey, A. M., Muñoz, D., & Lithwick, Y. 2020, *ApJL*, **892**, L29

Dipierro, G., Price, D., Laibe, G., et al. 2015, *MNRAS*, **453**, L73

Dittmann, A. J., & Ryan, G. 2021, *ApJ*, **921**, 71

Dittmann, A. J., & Ryan, G. 2022, *MNRAS*, **513**, 6158

Dittmann, A. J., & Ryan, G. 2024, *ApJ*, **967**, 12

Dittmann, A. J., Ryan, G., & Miller, M. C. 2023, *ApJL*, **949**, L30

D'Orazio, D. J., & Duffell, P. C. 2021, *ApJL*, **914**, L21

D'Orazio, D. J., Haiman, Z., Duffell, P., MacFadyen, A., & Farris, B. 2016, *MNRAS*, **459**, 2379

D'Orazio, D. J., Haiman, Z., & MacFadyen, A. 2013, *MNRAS*, **436**, 2997

Duffell, P. 2024, The Santa Barbara Binary Code Comparison, v1, Zenodo doi:10.5281/zenodo.10557951

Duffell, P. C. 2016, *ApJS*, **226**, 2

Duffell, P. C., D'Orazio, D., Derdzinski, A., et al. 2020, *ApJ*, **901**, 25

Dunhill, A. C., Cuadra, J., & Dougados, C. 2015, *MNRAS*, **448**, 3545

Facchini, S., Lodato, G., & Price, D. J. 2013, *MNRAS*, **433**, 2142

Farris, B. D., Duffell, P., MacFadyen, A. I., & Haiman, Z. 2014, *ApJ*, **783**, 134

Farris, B. D., Duffell, P., MacFadyen, A. I., & Haiman, Z. 2015, *MNRAS*, **446**, L36

Farris, B. D., Gold, R., Paschalidis, V., Etienne, Z. B., & Shapiro, S. L. 2012, *PhRvL*, **109**, 221102

Felker, K. G., & Stone, J. M. 2018, *JCoPh*, **375**, 1365

Franchini, A., Lubow, S. H., & Martin, R. G. 2019, *ApJL*, **880**, L18

Franchini, A., Lupi, A., & Sesana, A. 2022, *ApJL*, **929**, L13

Franchini, A., Lupi, A., Sesana, A., & Haiman, Z. 2023, *MNRAS*, **522**, 1569

Franchini, A., Prato, A., Longarini, C., & Sesana, A. 2024, arXiv:2402.00938

Franchini, A., Sesana, A., & Dotti, M. 2021, *MNRAS*, **507**, 1458

Gaburov, E., & Nitadori, K. 2011, *MNRAS*, **414**, 129

Gingold, R. A., & Monaghan, J. J. 1977, *MNRAS*, **181**, 375

Gold, R., Paschalidis, V., Etienne, Z. B., Shapiro, S. L., & Pfeiffer, H. P. 2014, *PhRvD*, **89**, 064060

Goodchild, S., & Ogilvie, G. 2006, *MNRAS*, **368**, 1123

Hanawa, T., Ochi, Y., & Ando, K. 2010, *ApJ*, **708**, 485

Hayasaki, K., Mineshige, S., & Sudou, H. 2007, *PASJ*, **59**, 427

Heath, R. M., & Nixon, C. J. 2020, *A&A*, **641**, A64

Hirsh, K., Price, D. J., Gonzalez, J.-F., Ubeira-Gabellini, M. G., & Ragusa, E. 2020, *MNRAS*, **498**, 2936

Hopkins, P. F. 2015, *MNRAS*, **450**, 53

Kratter, K. M. 2011, PhD thesis, Univ. of Toronto, Canada

Lai, D., & Muñoz, D. J. 2023, *ARA&A*, **61**, 517

Lodato, G., & Price, D. J. 2010, *MNRAS*, **405**, 1212

Lucy, L. B. 1977, *AJ*, **82**, 1013

Lynden-Bell, D., & Pringle, J. E. 1974, *MNRAS*, **168**, 603

MacFadyen, A. I., & Milosavljević, M. 2008, *ApJ*, **672**, 83

Mahesh, S., McWilliams, S. T., & Pirog, M. 2023, arXiv:2305.01533

Major Krauth, L., Davelaar, J., Haiman, Z., et al. 2023, *MNRAS*, **526**, 5441

Martin, R. G., & Lubow, S. H. 2019, *MNRAS*, **490**, 1332

Masset, F. 2000, *A&AS*, **141**, 165

Mignon-Risse, R., Varniere, P., & Casse, F. 2023, *MNRAS*, **520**, 1285

Mignone, A., Bodo, G., Massaglia, S., et al. 2007, *ApJS*, **170**, 228

Miranda, R., Muñoz, D. J., & Lai, D. 2017, *MNRAS*, **466**, 1170

Monaghan, J. J. 1992, *ARA&A*, **30**, 543

Moody, M. S. L., Shi, J.-M., & Stone, J. M. 2019, *ApJ*, **875**, 66

Muñoz, D. J., & Lai, D. 2016, *ApJ*, **827**, 43

Muñoz, D. J., Lai, D., Kratter, K., & Miranda, R. 2020, *ApJ*, **889**, 114

Muñoz, D. J., & Lithwick, Y. 2020, *ApJ*, **905**, 106

Muñoz, D. J., Miranda, R., & Lai, D. 2019, *ApJ*, **871**, 84

Muñoz, D. J., Springel, V., Marcus, R., Vogelsberger, M., & Hernquist, L. 2013, *MNRAS*, **428**, 254

Nealon, R., Price, D. J., & Nixon, C. J. 2015, *MNRAS*, **448**, 1526

Nixon, C., King, A., & Price, D. 2013, *MNRAS*, **434**, 1946

Nixon, C. J., Cossins, P. J., King, A. R., & Pringle, J. E. 2011, *MNRAS*, **412**, 1591

Noble, S. C., Krolik, J. H., Campanelli, M., et al. 2021, *ApJ*, **922**, 175

Noble, S. C., Mundim, B. C., Nakano, H., et al. 2012, *ApJ*, **755**, 51

Ochi, Y., Sugimoto, K., & Hanawa, T. 2005, *ApJ*, **623**, 922

Park, J., Ricotti, M., & Sugimura, K. 2024, *MNRAS*, **528**, 6895

Penzlin, A. B. T., Kley, W., Audiffren, H., & Schäfer, C. M. 2022, *A&A*, **660**, A101

Price, D. J. 2012, *JCoPh*, **231**, 759

Price, D. J., & Federrath, C. 2010, *MNRAS*, **406**, 1659

Price, D. J., Wurster, J., Tricco, T. S., et al. 2018, *PASA*, **35**, e031

Ragusa, E., Alexander, R., Calcino, J., Hirsh, K., & Price, D. J. 2020, *MNRAS*, **499**, 3362

Ragusa, E., Dipierro, G., Lodato, G., Laibe, G., & Price, D. J. 2017, *MNRAS*, **464**, 1449

Ragusa, E., Lodato, G., & Price, D. J. 2016, *MNRAS*, **460**, 1243

Roe, P. L. 1981, *JCoPh*, **43**, 357

Roedig, C., Dotti, M., Sesana, A., Cuadra, J., & Colpi, M. 2011, *MNRAS*, **415**, 3033

Roedig, C., Sesana, A., Dotti, M., et al. 2012, *A&A*, **545**, A127

Shi, J.-M., & Krolik, J. H. 2015, *ApJ*, **807**, 131

Shi, J.-M., Krolik, J. H., Lubow, S. H., & Hawley, J. F. 2012, *ApJ*, **749**, 118

Siwek, M., Weinberger, R., & Hernquist, L. 2023a, *MNRAS*, **522**, 2707

Siwek, M., Weinberger, R., Muñoz, D. J., & Hernquist, L. 2023b, *MNRAS*, **518**, 5059

Smallwood, J. L., Lubow, S. H., Franchini, A., & Martin, R. G. 2019, *MNRAS*, **486**, 2919

Springel, V. 2010, *MNRAS*, **401**, 791

Stone, J. M., & Gardiner, T. 2009, *NewA*, **14**, 139

Stone, J. M., Gardiner, T. A., Teuben, P., Hawley, J. F., & Simon, J. B. 2008, *ApJS*, **178**, 137

Stone, J. M., & Norman, M. L. 1992, *ApJS*, **80**, 753

Stone, J. M., Tomida, K., White, C. J., & Felker, K. G. 2020, *ApJS*, **249**, 4

Stout, Q. F., De Zeeuw, D. L., Gombosi, T. I., et al. 1997, in Proc. of the 1997 ACM/IEEE Conf. on Supercomputing, SC '97 (New York: ACM), 1

Sudarshan, P., Penzlin, A. B. T., Ziampras, A., Kley, W., & Nelson, R. P. 2022, *A&A*, **664**, A157

Tang, Y., MacFadyen, A., & Haiman, Z. 2017, *MNRAS*, **469**, 4258

Terquem, C., Sørensen-Clark, P. M., & Bouvier, J. 2015, *MNRAS*, **454**, 3472

Teyssandier, J., & Ogilvie, G. I. 2016, *MNRAS*, **458**, 3221

Thun, D., Kley, W., & Picogna, G. 2017, *A&A*, **604**, A102

Tiede, C., & D'Orazio, D. J. 2024, *MNRAS*, **527**, 6021

Tiede, C., D'Orazio, D. J., Zwick, L., & Duffell, P. C. 2024, *ApJ*, **964**, 46

Tiede, C., Zrake, J., MacFadyen, A., & Haiman, Z. 2020, *ApJ*, **900**, 43

Tiede, C., Zrake, J., MacFadyen, A., & Haiman, Z. 2022, *ApJ*, **932**, 24

Toro, E. F., Spruce, M., & Speares, W. 1994, *ShWav*, **4**, 25

van Leer, B. 1974, *JCoPh*, **14**, 361

Wang, H.-Y., Bai, X.-N., & Lai, D. 2023b, *ApJ*, **943**, 175

Wang, H.-Y., Bai, X.-N., Lai, D., & Lin, D. N. C. 2023a, *MNRAS*, **526**, 3570

Westernacher-Schneider, J. R., Zrake, J., MacFadyen, A., & Haiman, Z. 2022, *PRD*, **106**, 103010

Westernacher-Schneider, J. R., Zrake, J., MacFadyen, A., & Haiman, Z. 2024, *ApJ*, **962**, 76

Young, M. D., Baird, J. T., & Clarke, C. J. 2015, *MNRAS*, **447**, 2907

Zrake, J., & MacFadyen, A. I. 2012, *ApJ*, **744**, 32

Zrake, J., Tiede, C., MacFadyen, A., & Haiman, Z. 2021, *ApJL*, **909**, L13