Tools for Studying Magnetospheric-Wind Interactions

by

Christopher Bard

A dissertation submitted in partial fulfillment of

the requirements for the degree of

Doctor of Philosophy

(Astronomy)

at the

University of Wisconsin–Madison

2016

Date of final oral examination: 23 June 2016

The dissertation is approved by the following members of the Final Oral Committee:

Dr. Richard Townsend, Associate Professor, Astronomy
Dr. Sebastian Heinz, Professor, Astronomy
Dr. Robert Mathieu, Professor, Astronomy
Dr. Asif ud-Doula, Associate Professor, Physics, Penn State Worthington-Scranton
Dr. Ellen Zweibel, Professor, Astronomy

Abstract

Magnetic fields have an enormous influence on the behavior of plasma, especially around stars and planets. When winds flowing from stars encounter strong magnetic structures, both field and flow change their very natures. In this thesis, I ask the question: How do stellar winds interact with planetary and stellar magnetic fields within magnetospheres? Here, I present tools to study each of these two cases. In both cases, high plasma wavespeeds (usually caused by strong magnetic fields) make traditional computer codes inefficient due to very small timesteps and require unique methods to ameliorate this problem. For a planetary magnetosphere, where the wind source is external to the magnetic field, I accelerate a Hall magnetohydrodynamics code with multiple graphics processing units. I discuss the process involved in converting a traditional code running on normal computer processors to run on graphics processors. For a massive-star magnetosphere with a line-driven wind coming from within the field, I simplify the three-dimensional magnetosphere as many one-dimensional flows along magnetic field lines. I present the resulting Arbitrary Rigid-Field Hydrodynamics equations and analyze them to quantify the effect of a stellar magnetic dipole field on the line-driven wind of a massive-star. I find that the divergence of the magnetic field boosts the surface mass-flux, but the non-radial tilt hinders it; the overall effect is a reduction in the total stellar mass loss rate, which has implications for massive star emission and evolution. The results of this analysis is then used to update the most current X-ray magnetospheric emission model. I find that using non-magnetic

values of wind properties leads to an underestimate of O star X-rays and an overestimate of later-type B star X-rays.

Acknowledgments

This thesis would not be possible without Mom and Dad, for obvious reasons. Thank you for letting me do whatever (despite Uncle Steve's insistence that everyone in the family become an engineer), and for always having a place I can come home to.

I certainly would not have become an astronomer without the support of my mentors at the University of Delaware. Mike Shay, thanks for trusting a dumb undergraduate with hundreds of thousands of computer hours and getting me started on this path of creating simulations. Stan Owocki, you started me on this path of studying massive stars. Thank you for being a wise sage and training me in the ways of the (line-driven) Force. Dermott Mullan, thanks for your support throughout my astronomy classes and for recommending me for that NASA internship way back when. It's led to many opportunities for me. At NASA, John Dorelli, thank you for letting me work with you on that internship (and fellowship) and having a really interesting idea to use GPUs for physics. I look forward to working with you again next year.

It's been quite a ride doing astronomy these last six years at Wisconsin. Rich, I still remember meeting you for the first time in Stan's office and that cartoon map of Madison you drew. Thanks for being patient with me as I learned how to do science and allowing me to go off on this unplanned tangent which became my thesis. You always challenged me to have evidence to back up my claims, and this has helped me become a stronger scientist. Bob and Ellen, thanks for always making Stars coffee interesting and giving me insights

into the world of professordom. I have greatly enjoyed playing Foose with the table Bob bought for the grad students. Sebastian, thanks for making my first semester in Madison a smooth (though not easy!) introduction into the world of astronomy. Steve, thanks for always having a door open, ready to answer my clueless questions.

Madison has been a great place to live, and certainly not because of the weather. John, Anna, Jenna, I can't believe we've shared an office for six years and not killed each other. Thank you for putting up with my foibles and being great friends, always there for support and procrastination. It'll be a sad day when the Atom is fully ionized. I hope we have plenty more pictures in the future without anyone photoshopped in. Arthur, it's been great fun through all the beer, baseball, and trivia. You better get back from the South Pole so we can do it all again. Josh from Old Rochelle, thanks for inviting some random friends of some guy you knew from college to play softball. I still like to think I made your one title possible by not playing that season. Claire and David, here's to all those late nights we spent at your parties. Ben, we need to get back to Rock Island someday (a la Jack from Lost). To all my friends here (and graduated), including the ones I am forgetting in this post-thesis haze, thank you for making the last six years an amazing journey.

Finally, I raise a pint to the best trivia team City Bar has ever seen (Nick, Greg, Arthur, Brian, Jenna, with newcomers Zach, DK, Ben R.) in remembrance of nearly 200 different team names. My favorite one? Probably Jort Malort and the Blorty Blorts.

Contents

List of Tables

List of Figures

List of Selected Abbreviations and Symbols

$\aleph$ Rotation parameter

χ Wind transparency parameter

χ_∞ XADM terminal speed cooling parameter

δ_i Ion inertial length

$\dot{M}$ Mass-loss rate

$\dot{M}_{\rm CAK}$ CAK spherical mass-loss rate

$\dot{M}_{\rm global}$ Global mass-loss rate

$\dot{m}_*$ Stellar surface mass-flux into field line

$\dot{m}_{\rm wgh}$ Weighing term for XADM mass-flux

$\dot{m}_{\rm dip}$ Dipole-corrected CAK surface mass-flux

$\dot{m}_{\theta=0}$ or $\dot{m}_{{\rm gen},\theta=0}$ Polar surface mass-flux

$\dot{m}_r$ Radial surface mass-flux

η_* Wind magnetic confinement parameter

μ_* Cosine of angle to edge of stellar disk

μ_B Surface tilt of magnetic field line

μ_m Cosine of surface colatitude

μ_s Cosine of angle to shock location

Σ_0 Optically-thin-correction parameter, zero rotation

$\Sigma_{\rm rot}$ Optically-thin-correction, with rotation

$\Sigma_{\rm surf}$ Oblate star surface area

Σ_f Optically-thin correction, finite-disk case

Σ_g Surface-weighed gravity

$\tau_{\rm sob}$ Sobolev optical depth

θ_m Surface colatitude of field line

A Cross-sectional area of field line

D Limb-darkening function

f Finite-disk correction factor

K_s Shock-dissipated energy relative to total wind kinetic energy

L Field line shell parameter

R_* Stellar radius

R_c Magnetospheric closure radius

R_K Kepler radius

R_p Stellar polar radius

s Arc distance along field line

v_∞ Terminal wind velocity

$v_{\rm esc}$ Escape velocity

$\mathbf{v}_H$ Hall velocity

$\Gamma_{\rm el}$ Eddington parameter

Chapter 1

Introduction

I tramp a perpetual journey, (come listen all!)

Walt Whitman

Magnetic fields permeate the universe, influencing the evolution of molecular clouds (Crutcher 2012), stars (McKee & Ostriker 2007) and the large-scale structure of the universe (Ryu et al. 2012). On smaller scales, winds can interact with large-scale stellar or planetary magnetic fields; these interaction regions are called "magnetospheres". We are primarily interested in understanding two kinds of magnetospheres. In the first kind, a planetary magnetosphere, the stellar wind flows from outside a planet's magnetic field and compresses it. In the second, a massive-star magnetosphere, the stellar wind flows from within the star's magnetic field and strains against it in an attempt to escape. Each type of magnetosphere has unique characteristics that severely reduce the maximum timestep size, making efficient numerical simulations difficult and requiring the development of special tools. In this thesis, we present two computational tools, one for each kind of magnetosphere, designed to solve their governing equations quickly and/or more efficiently. The first tool is a Hall magnetohydrodynamics (MHD) code accelerated by multiple graphics processing units (GPUs), which will allow us to recreate planetary magnetospheres and simulate them in less time. The second tool is a program to solve the Arbitrary Rigid-Field Hydrodynamics Equations, a simplification of the conditions within the strong, nearly-rigid magnetic fields of massive stars. These tools will further our understanding of how stellar winds affect large-scale magnetic fields, and vice versa.

In Section 1.1, we present the difficulties and goals of simulating planetary magnetospheres and explain why we need graphics processing units (GPUs) to speed up computer simulations. Massive-star magnetospheres are introduced in Section 1.2 and a brief history of the field is given. Finally, Section 1.3 serves as a primer on massive-star line-driven winds in preparation for the upcoming analysis on stellar winds in the context of a dipole magnetic field (Chapters 4 and 5).

1.1 Understanding Planetary Magnetospheres

Magnetic reconnection is considered to be one of the most important processes in space and astrophysical plasmas. On the Sun, it is believed to contribute to solar coronal mass ejections (Antiochos et al. 1999) and coronal heating (Cargill & Klimchuk 1997). In Earth's magnetosphere, reconnection allows the solar wind plasma and energy to enter within the planetary magnetic field (Sonnerup et al. 1981) and may also contribute to the formation of the auroral acceleration region (Haerendel 1987). In order to understand the physical processes within planetary magnetospheres, we need to understand magnetic reconnection.

1.1.1 Magnetic reconnection and the Hall term

While the nuances of reconnection are outside the scope of this thesis, I present here a basic understanding of why the Hall term is needed for planetary magnetosphere simulations despite its severe computational cost. For additional details of reconnection, the reader is referred to, e.g., Priest & Forbes (2000) and Yamada et al. (2010).

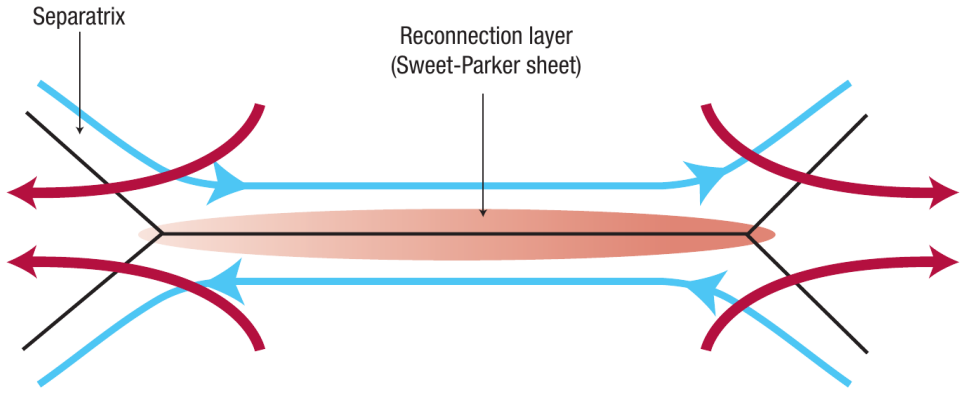


Figure 1.1 : Structure and flow (red) of field lines (blue) during two-dimensional magnetic reconnection. High reconnection rates in the layer around the X-line cause the reconnection region to flatten in to a Sweet-Parker current sheet, which places limits on the rates that can be described by "resistive" MHD. Figure adapted from Finn (2006).

In the simplest picture of magnetic reconnection (Figure 1.1, the Sweet-Parker model (Sweet 1958; Parker 1957), two regions separated by sheared magnetic fields are pushed together by plasma flows and form a current sheet in between. The magnetic fields break apart at the sheet and reform with oppositely directed field lines, creating a x-line configuration. The newly reconnected field lines immediately straighten due to magnetic tension, driving plasma away from the x-line. The process continues as more plasma flows into the x-line to replace the lost plasma.

The rate of reconnection is simply the rate at which magnetic flux goes into the x-line region. If we define x and z as the horizontal and vertical axes of Figure 1.1 with y into the page, we can say the reconnection region has a size δ in the z-direction and a length Δ along the x-direction. Sweet and Parker derived scalings for the reconnection rate in this region by considering a steady-state system, eventually deriving

$$\text{rate} \sim v_{\text{in}} \sim \frac{\delta}{\Delta} v_A, \tag{1.1}$$

with the Alfvén speed $v_A = B/\sqrt{4\pi\rho}$. δ is typically taken as the thinness of the current sheet, a microscopic quantity, while Δ is the macroscopic size of the system. This simple scaling results in reconnection rates far too slow to explain observations (e.g. in the subsolar magnetopause: Mozer et al. 2003).

In the 1990s, the Hall term was found to be important in allowing fast magnetic reconnection (e.g. Biskamp et al. 1995; Shay et al. 1999; Birn et al. 2001). In this picture (Figure 1.2), the length scale of the reconnection region no longer depends on the macroscopic size of the system. We can understand this by looking at the resistive Ohm's law, with the Hall term:

$$0 = \mathbf{E} + \frac{\mathbf{v} \times \mathbf{B}}{c} - \eta \mathbf{J} - \frac{1}{nec} \mathbf{J} \times \mathbf{B} \tag{1.2}$$

with $\mathbf{J}$ the current density, η the plasma resistivity, n the plasma number density, and m_e and e the electron mass and charge. In resistive MHD, as in the Sweet-Parker model, the Hall term $\mathbf{J} \times \mathbf{B}$ is neglected. The magnitude of the Hall term compared to the ideal MHD terms E and $\mathbf{v} \times \mathbf{B}$ is proportional to the ion inertial length $\delta_i = c\sqrt{m_i/(4\pi n e^2)}$, so we can safely neglect the Hall term for length scales longer than δ_i. Within length scales approaching δ_i, however, the ions decouple from the magnetic field while the electrons remain frozen-in to the magnetic field. As a result, reconnection occurs when the electrons finally decouple from the magnetic field (rather than the overall plasma as in Sweet-Parker). The reconnection rate is then set by the physics of electron whistler waves, with speeds inversely proportional to their wavelengths, rather than general Alfvén waves, which propagate at the Alfvén velocity independent of wavelength. This makes reconnection fast compared to the Sweet-Parker scaling.

We note that the actual effect of the Hall term is still an open issue (Zweibel & Yamada 2009). Fast collisionless reconnection can also occur without dispersive waves, like in electron-positron plasmas (Daughton & Karimabadi 2007) or in electron-proton plasmas

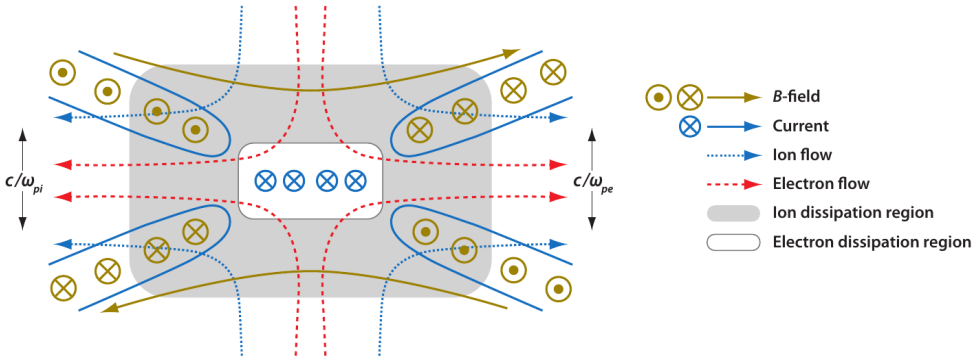


Figure 1.2 : Sketch of the magnetic field geometry in reconnection. The ions decouple from the electrons at the ion inertial length $\delta_i = c/\omega_{pi}$. The electrons continue flowing, and the field is reconnected within the much smaller electron layer. Figure taken from Zweibel & Yamada (2009).

with turbulence (Lazarian & Vishniac 1999). For MHD simulations with typical length scales $\ggg \delta_i$, the exact physics within the reconnection region is not so important; the fast rate of reconnection induced by the Hall term is. Thus, for simulations of planetary magnetospheres, adding the Hall term to the ideal MHD equations will be sufficient to induce fast reconnection in a physically-driven (rather than numerically-driven) manner.

1.1.2 Issues with Investigating Reconnection

The discussion in the previous section is limited since it only considers two-dimensional geometries. Reconnection is fundamentally a three-dimensional process. However, understanding the exact nature of reconnection in three dimensions is difficult; there have been many definitions proposed for this mechanism. One states that reconnection is the mechanism by which plasma flows across the boundary of topologically distinct magnetic field lines (Vasyliunas 1975). However, this definition is difficult to extend to three dimensions since the two-dimensional boundaries between distinct field lines do not survive three-dimensional perturbations of the field (Schindler et al. 1988). A more general definition proposed by Hesse & Schindler (1988) posits that reconnection occurs when the magnetic field evolves in such a way that two plasma parcels that were connected by the same field line are no longer connected. This may be too general, since it depends on an arbitrary definition of a plasma flow element.

Boozer (2002) proposed that magnetic reconnection is a spatially localized magnetic field evolution that does not conserve magnetic flux. Nevertheless, this definition may be too specific as it excludes some cases of reconnection in which magnetic flux is conserved, like steady-state reconnection in the magnetosphere.

A fourth, weaker definition states that reconnection occurs when the magnetic field can be divided into topologically distinct flux volumes defined by intersecting separatrix

surfaces and a parallel electric field exists at the intersections of the separatrix surfaces (Dorelli et al. 2007). Though this definition is a very straightforward generalization of two-dimensional reconnection, it does exclude cases that have been traditionally considered to be reconnection, like a finite-length plasmoid in the magnetotail (Schindler et al. 1988).

A more recent study (Glocer et al. 2014) focuses on magnetic separators, which are the three-dimensional analogue of a two-dimensional field line with a guide field and are defined by the line corresponding to the intersection of the separatrix surfaces associated with the so-called magnetic null points at which the magnetic field vanishes. A separator in the magnetosphere represents the intersection of different magnetic topologies; finding them gives us a way to measure the reconnection in three dimensions.

Magnetic null points may not be required for reconnection, however. Priest & Démoulin (1995) showed that reconnection may occur at so-called "quasi-separatrix layers", where the local field line velocity may exceed the local plasma velocity. This produces an electric field component parallel to the magnetic field and allows the field lines to rapidly slip through the plasma.

Ultimately, in order to find a clear, widely accepted definition of reconnection, we must further investigate its properties. The recently-launched Magnetosphere Multiscale (MMS) mission (Fuselier et al. 2016) promises to yield insights into the very nature of reconnection by directly measuring the behavior of post-reconnection plasma in Earth's magnetosphere. We will need simulations and models to help parse the data (Hesse et al. 2016).

1.1.3 Simulating Planetary Magnetospheres

Global computer simulations of magnetospheres demand intensive computational resources. In order to resolve the small length scales of thin current sheets (tens of

kilometers thick) within the context of the Earth's magnetosphere (hundreds of Earth radii long), we need millions of computational cells each separated by a small grid scale length. Additionally, to resolve the short, Alfvénic timescales associated with reconnection, we need to use short time steps. Adding the Hall term also introduces a new difficulty: whistler waves. Since whistler wave speeds are proportional to the grid scale (e.g. Tóth et al. 2008), reducing the grid resolution more than halves the largest stable timestep (Courant et al. 1928). This combination of many grid points, small grid scale lengths, and small time steps results in huge simulations taking several weeks, if not months, to run on traditional CPU clusters.

Though Earth's magnetosphere is too large for current Hall MHD simulation codes, smaller magnetospheres, like Ganymede, have successfully been simulated. MHD simulations of Ganymede's magnetosphere (Jia et al. 2009) are able to produce global pictures of the moon's magnetic field that agree well with *Galileo* magnetometer observations. More recent Hall MHD simulations (Dorelli et al. 2015) are able to demonstrate that currents associated with the Hall term have a significant impact on the global structure of Ganymede's magnetosphere by driving convective flows and contributing to the observed Ganymedean aurorae. Since these effects are driven by magnetic reconnection, it is expected that they be present in all magnetospheres. However, the large scale consequences and resulting observational signatures will depend on several factors which vary between planetary bodies: 1) the strength of the magnetic field, 2) the magnetospheric convection, and 3) the nature of the body's surface and atmosphere (which affects how field-aligned currents close).

In addition to Ganymede, there have been several initial MHD studies of exoplanet magnetospheres, such as hot Jupiters (Vidotto et al. 2015) and Venus-like planets around a M-dwarf host (Cohen et al. 2014). These are an important first step in understanding the environment as well as potential observational signatures (e.g. radio emission) of these

magnetospheres. However, these simulations do not incorporate the Hall term, instead using ideal MHD. As noted from simulations of Ganymede, the Hall term may have a significant impact on these exoplanetary magnetospheres and their resulting emission. Even though we have successfully simulated these magnetospheres, we will need better performing codes in order to understand how Hall effects influence larger magnetospheres, such as around both Earth and exoplanets.

1.2 Simulating Massive Star Magnetospheres

In cool stars, the vigorous motions of convection generate magnetic fields in their outer envelopes. While the exact details are not yet known, the basic principles of this dynamo mechanism are well understood (Donati & Landstreet 2009; Charbonneau 2010). This dynamo mechanism starts to disappear in F-type stars with masses $\gtrsim 1.5\ M_{\odot}$ (Hall 2008) and indeed does not exist for massive stars. Despite this, there have been many observations of magnetic fields from early-type stars (e.g. Babcock 1958; Borra & Landstreet 1980; Wade et al. 2016). These magnetic fields are thought to be *fossil fields* left over or accumulated from earlier stages of stellar formation/evolution (Donati et al. 2006; Donati & Landstreet 2009). Indeed, semi-analytic models and numerical simulations have shown that large-scale stable magnetic fields can exist in stellar radiative zones (Spruit 2002; Duez & Mathis 2010; Duez et al. 2010). While the creation of such magnetic fields are outside the scope of this thesis, these fields have a significant impact on stellar winds and their emission.

Over the last few decades, we have seen certain observational phenomena associated with these magnetic massive stars: Hα emission (Walborn 1974), photometric and Hα variability (Landstreet & Borra 1978), infrared emission and variability (Eikenberry et al. 2014), UV resonance line variations (Shore & Brown 1990), hard X-ray emission (Gagné et al. 2005), and radio emission (Linsky et al. 1992; Chandra et al. 2015). Models that seek

to explain these phenomena all have one thing in common: the magnetic field of the star is to blame.

1.2.1 Characterization of massive-star magnetospheres

In massive magnetic stars, there is a competition between the stellar wind and magnetic field. Under the frozen-flux condition of ideal MHD (e.g. Priest & Forbes 2000), there are two outcomes: either the magnetic topology can shift as a result of plasma motions or the plasma motion can be channeled along fixed magnetic field lines. The first case comes about when the wind kinetic energy dominates the magnetic energy; the second case occurs when magnetic energy dominates wind energy. This relationship between wind strength and magnetic confinement can quantified through the ratio of their respective energy densities:

$$\begin{aligned} \eta &= \frac{B^2/8\pi}{\rho v^2/2} \approx \frac{B^2 r^2}{\dot{M} v} \\ &= \frac{B_*^2(\theta) R_*^2}{\dot{M} v_\infty} \frac{(r/R_*)^{-4}}{(1 - R_*/r)^\beta}, \end{aligned} \tag{1.3}$$

where, in the second equality, we have substituted in the spherically-symmetric mass-loss rate $\dot{M} = 4\pi r^2 \rho v$. In the second line, we have approximated the stellar magnetic field strength at surface colatitude θ as $B = B_*(\theta)(R_*/r)^3$, appropriate for a dipole. We also assume that the stellar velocity follows a beta law $v = v_\infty(1 - R_*/r)^\beta$ with v_∞ the terminal velocity. We can evaluate Equation 1.3 at the magnetic equator ($\theta = 90°$), where the radial flow is most directly competing against the dipole field, and define the left fraction as the now canonical "wind magnetic confinement parameter", $\eta_\star$ (ud-Doula & Owocki 2002):

$$\eta_\star = \frac{B_{eq}^2 R_\star^2}{\dot{M}_{B=0} v_{\infty,B=0}} \ , \tag{1.4}$$

where B_{eq} is the magnetic field strength at the stellar equatorial radius $R_\star$, and $\dot{M}_{B=0}$ and $v_{\infty,B=0}$ are the mass-loss rate and terminal wind speed the star would have in the absence

of a magnetic field. For high values of $\eta_\star$, the wind is channeled by the field and retained. On the other end of the continuum, the wind escapes into the interstellar medium, though its flow is still slightly affected by the field.

Massive, magnetic stars typically have $\eta_\star > 1$. O stars, due to their enormous mass-loss rates, have values of $\eta_\star$ closer to 1 (e.g. θ^1 Ori C has $\eta_\star \approx 16$), while the He-strong B stars have $\eta_\star \gg 1$ (e.g. σ Ori E has $\eta_\star \approx 5 \times 10^5$). Having $\eta_\star \gg 1$ results in a fixed magnetic topology in which the plasma flow is unable to effect any large-scale changes in the field and is forced to corotate with the star in a magnetosphere. The behavior of plasma within the magnetosphere can be quantified by considering the length scales of the locally dominant mechanisms: magnetic field confinement and centrifugal support from rigid-body rotation. For centrifugal support, the relevant length scale is the Kepler corotation radius,

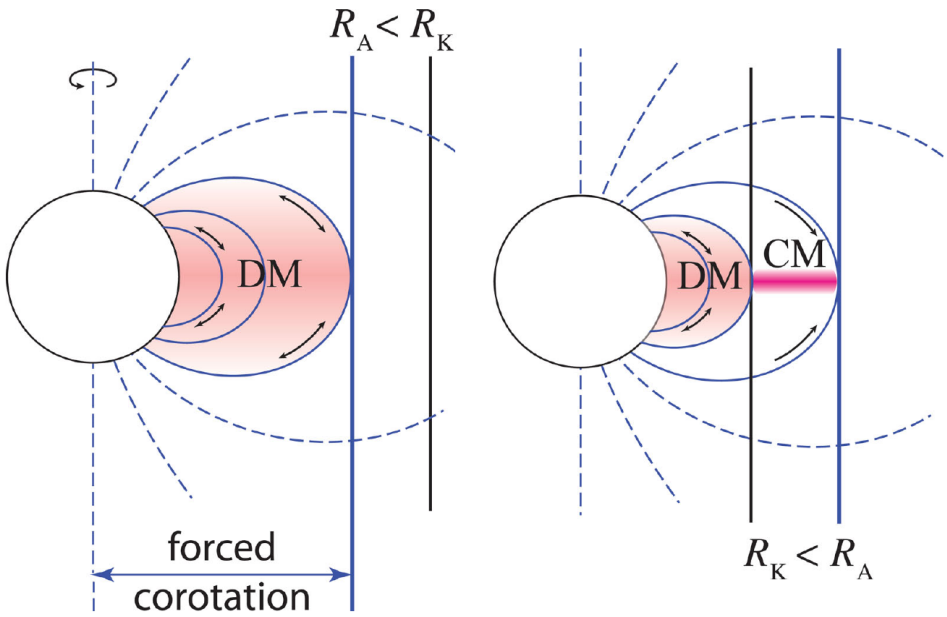


Figure 1.3 : Sketch of the regimes for dynamical versus centrifugal magnetospheres (DM vs. CM). The left panel illustrates a slowly rotating star with the Kepler radius beyond the Alfvén radius ($R_K > R_A$). The right panel illustrates a more rapidly rotating star with $R_A > R_K$, leading to a centrifugally-supported disk in between these radii. Color illustrates the rough time-averaged density of material; darker is denser. Figure adapted from Petit et al. (2013).

the point at which the outward centrifugal force exceeds the inward gravitational force:

$$R_K = \left(\frac{V_{rot}}{V_{orb}}\right)^{-2/3} R_\star = \omega^{-2/3} R_\star \ , \tag{1.5}$$

where V_{rot} is the stellar rotational velocity, $V_{orb} = \sqrt{GM_\star/R_\star}$ is the orbital speed near the equatorial surface, and $\omega \equiv V_{rot}/V_{orb}$ is the critical rotation fraction. For $R \lesssim R_K$, there is a net gravitocentrifugal force inwards, and for $R \gtrsim R_K$, the gravitocentrifugal force is outwards.

The magnetic confinment can be parameterized by the Alfvén radius (R_A), the point at which the plasma is just barely confined by the magnetic field. An approximation of R_A, as per ud-Doula et al. (2008), is given by:

$$\frac{R_A}{R_\star} \approx 0.29 + (\eta_\star + 0.25)^{1/4} \ . \tag{1.6}$$

MHD simulations suggest that the wind starts to open magnetic loops before this radius; the maximum extent of closed field lines in the magnetosphere can be estimated with the "closure radius"

$$R_c \approx R_* + 0.7(R_A - R_*). \tag{1.7}$$

There are then two general possibilities for the confined plasma: a dynamical magnetosphere (DM) or a centrifugal magnetosphere (CM). We can differentiate between these situations by comparing their Alfvén and Kepler radii (Figure 1.3). In a DM, since $R_K > R_A$, the plasma trapped within the magnetic field does not experience any centrifugal support and falls back to the star under gravity. In a CM, there are two regions of interest: $R < R_K$, where trapped plasma falls back to the star, and $R_K < R < R_A$, where centrifugally supported plasma nestles within local effective potential minima. In either case, plasma at $R > R_A$ does not stay confined. Within this basic framework, hot

magnetic stars can be classified by magnetosphere type using their rotation and magnetic confinement (Figure 1.4).

1.2.2 Magnetosphere Models Past to Present

In the previous section, we gave scaling relations for the extent of magnetospheres. These scaling relations came from models of massive-star magnetospheres, specifically MHD

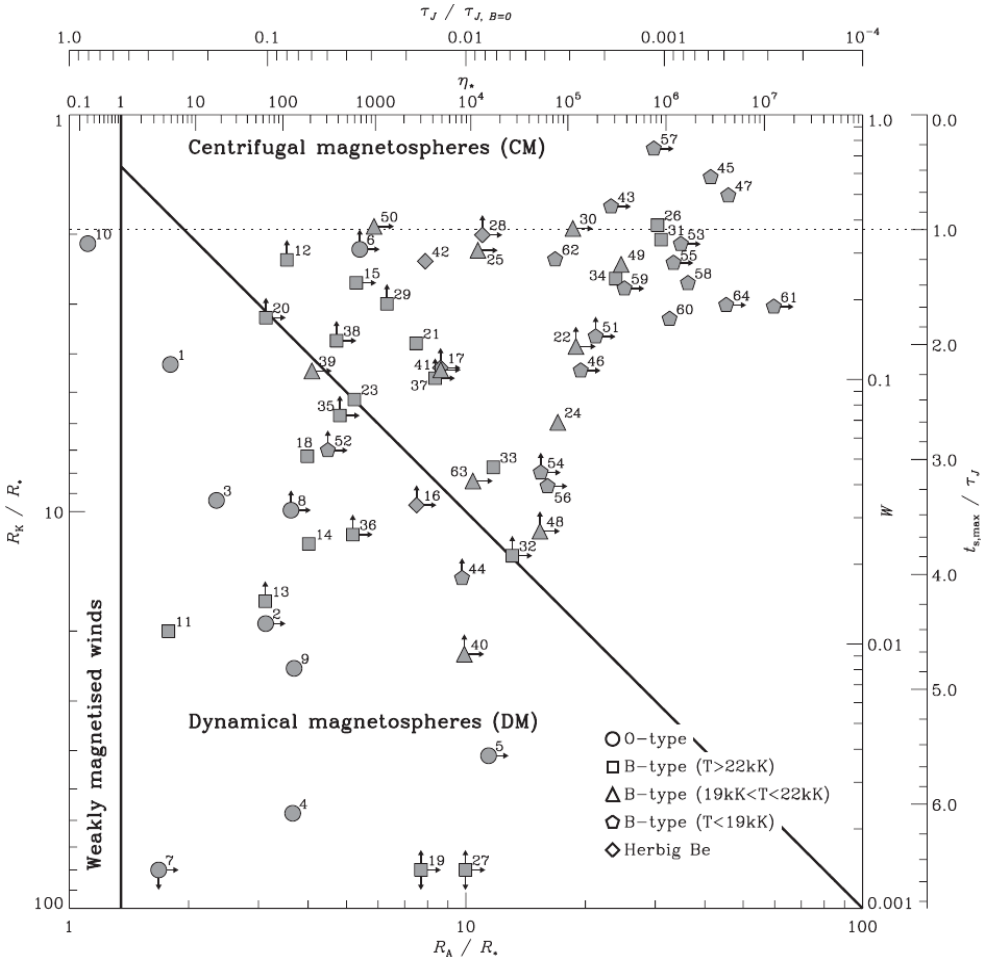


Figure 1.4 : Location of magnetic massive stars in the magnetic confinement-rotation diagram; a log-log plot with the Kepler radius R_K increasing downwards and the Alfvén radius R_A increasing to the right. The right and upper axes give the corresponding rotation fraction W (defined as ω in the text) and magnetic confinement parameter η_*. The solid lines separate the magnetosphere domains of weakly magnetized winds ($\eta_* < 1$), DM with $R_A < R_K$, and CM with $R_A > R_K$ as defined in the text. Symbol shapes denote spectral type. Figure taken from Petit et al. (2013); additional details can be found in that reference.

simulations (ud-Doula & Owocki 2002; ud-Doula et al. 2009). Here, we present the classical picture of a massive-star magnetosphere and several models designed to explain emission from these magnetospheres.

In the simplest picture of a magnetosphere, the star has a dipole magnetic field and is an "oblique rotator", having its magnetic dipole axis offset from its rotational axis (Stibbs 1950). Using this framework, Townsend & Owocki (2005), expanding on previous work by Michel & Sturrock (1974) and Nakajima (1985), studied the circumstellar gas trapped by

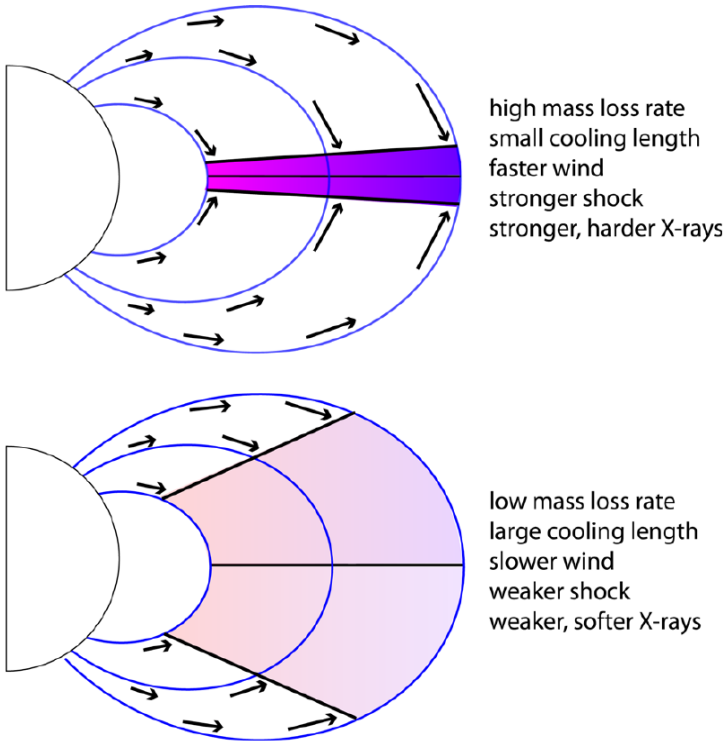


Figure 1.5 : Schematic illustration of the shock retreat from inefficient cooling associated with a lower mass-loss rate $\dot{M}$ showing a hemispheric, planar slice of a stellar dipole magnetic field. Wind outflow driven from opposite foot-points of closed magnetic loops is channelled into a collision near the loop top, forming wind shocks. For the high $\dot{M}$ case in the upper panel, the efficient cooling keeps the shock-heated gas within a narrow cooling layer, allowing the pre-shock wind to accelerate to a high speed and so produce strong shocks with strong, relatively hard X-ray emission. For the low $\dot{M}$ case in the lower panel, the inefficient cooling forces a shock retreat down to lower radii with slower pre-shock wind, leading to weaker shocks with weaker, softer X-ray emission. Figure taken from ud-Doula et al. (2014).

the star's magnetic field and corotating with the star. Within this so-called Rigidly Rotating Magnetosphere (RRM), the gas settles at local minima of the combined gravitational-centrifugal potential. As the star rotates, the changing Doppler shift of the various parts of the magnetosphere and the different column densities of the trapped gas in our line of sight combine to produce a periodic variation in $H\alpha$ emission. Additionally, the clouds in the magnetosphere also occult the star as it rotates, leading to photometric variability. This picture is supported by observations of the prototypical Bp star σ Ori E indicating that the circumstellar plasma emitting in $H\alpha$ and blocking photometric wavelengths is concentrated at the intersection between the magnetic and rotational equators (Groote & Hunger 1982; Short & Bolton 1994) as predicted by the RRM (Townsend et al. 2005). However, since the RRM only considers cooled plasma in hydrostatic equilibrium within a gravitocentrifugal potential, it can not be applied towards explaining X-ray emission and variability from hot, magnetic stars.

To explain the unusual hard X-rays emanating from these hot magnetic stars, Babel & Montmerle (1997) developed a Magnetically Confined Wind Shock (MCWS) paradigm. Essentially, since the strong magnetic field of the star is able to dictate the path of the stellar wind, it is possible for wind flowing from one footprint of a magnetic loop to collide with the wind flowing from the other footprint. The result from this collision of two supersonic flows is a strong shock capable of heating plasma to millions of Kelvin and producing hard X-rays. After the collision, the gas cools and can either fall back to the footprint or accumulate in a magnetosphere, depending on the local gravitocentrifugal force. The magnetosphere is thus comprised of two plasma states: a relatively cool, higher-density component near the rotational and magnetic equators and a hot, low-density coronal component at higher latitudes.

ud-Doula et al. (2014) expanded this model with their "XADM" framework by

introducing a "shock-retreat" paradigm in which the size of the X-ray emitting region within the magnetosphere depends on the post-shock cooling length from the shock to the apex of the field line (Figure 1.5). For higher mass-losses, the increased density in the wind shrinks the cooling length; this allows the wind to accelerate over longer lengths and reach higher velocities at the shock. For lower mass-losses, the cooling length increases and leads to lower shock velocities. The increase in shock velocity leads to higher X-ray luminosities at higher mass-loss rates, but the shrinking emission region eventually leads to a turnover in the X-ray luminosity at high mass-losses ($\dot{M} \sim 1 \times 10^{-5}\ M_{\odot}/\,\mathrm{yr}$). This trend is supported by an observational survey of X-ray emission from magnetospheres (Nazé et al. 2014).

One limitation of these "steady-state" models is that they do not attempt to solve for the dynamics of the wind or post-shock regions in the magnetosphere. To address this issue, ud-Doula & Owocki (2002) developed MHD simulations to investigate the stellar wind behavior and its interaction with the magnetic field (example in Figure 1.6). They found complex patterns of infall and outfall at the magnetic equator and stretched out

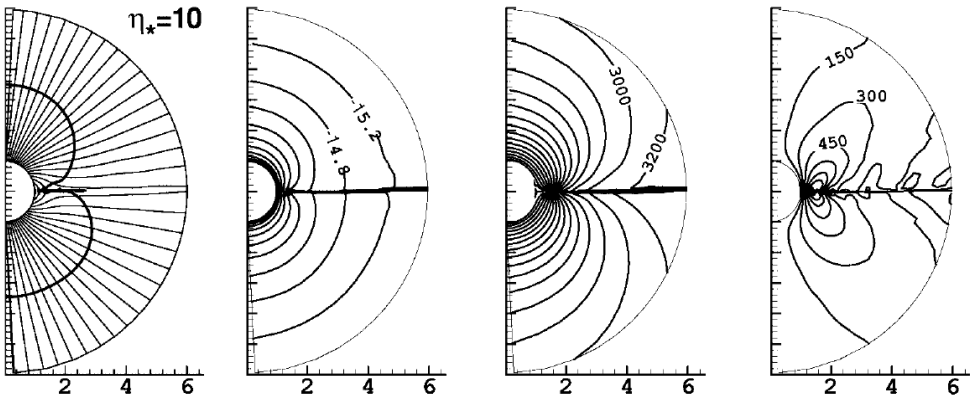


Figure 1.6 : Comparison of overall properties at the final simulation time ($t \approx 450$s) for a MHD model with $\eta_* = 10$. The leftmost panels show magnetic field lines, together with the location (bold contour) of the Alfvén radius, at which the radial flow speed equals the Alfvén speed. From left to right, the remaining columns show contours of log of density, radial velocity, and latitudinal velocity. Figure adapted from ud-Doula & Owocki (2002).

field lines nearer the magnetic poles. Additionally, these simulations found support for the MCWS paradigm, eventually leading to the development of a quantitative magnetic wind-shock model for the hard X-ray emission of magnetic massive stars (e.g. σ^1 Ori C: Gagné et al. (2005)). ud-Doula & Owocki (2002) were able to develop scaling relations to estimate the typical sizes of these magnetospheres (Equations 1.6, 1.7).

Unfortunately, large magnetic fields have very high Alfvén speeds, requiring very small timesteps to ensure numerical stability (Courant et al. 1928). Due to this limitation, ud-Doula & Owocki (2002) could only simulate magnetopsheres with $\eta_* \leq 10$. Later, ud-Doula et al. (2008) were able to simulate up to $\eta_* = 1000$. However, these confinement parameters are still much less than typical centrifugal magnetosphere values ($\eta_* \gtrsim 1 \times 10^4$); this raises the question of whether the wind behavior observed in the MHD simulations will be present within the very strong fields of centrifugal magnetospheres.

To get around this issue of large Alfvén speeds, Townsend et al. (2007) developed a Rigid-Field Hydrodynamics (RFHD) approach to modeling magnetospheres in the limit of very strong magnetic fields. In this ansatz, the magnetic fields are assumed to be completely rigid ($\eta_* \to \infty$), channeling the stellar wind along quasi-one-dimensional flux tubes subject to external conditions such as radiation pressure and gravitocentrifugal acceleration. These field lines can be simulated independently and their results stitched together to form a picture of the overall magnetosphere (Figure 1.7) The RFHD approach simulates the actual flow of the plasma along each field line, allowing for dynamical phenomena such as oscillations or the fallback of material to the footprint. Additionally, RFHD improves on the previous RRM model by allowing for a dynamic evolution of emission in both optical and X-ray, though it should be noted that RRM provides a good agreement with optical observations (Townsend et al. 2005) at a much lower computational cost.

Despite the successes of the RFHD approach, there is still room for improvement.

Using high-resolution spectropolarimetric observations of the centrifugal magnetosphere star σ Orionis E, Oksala et al. (2012) demonstrated that the periodic variations in its measured longitudinal magnetic field and circular polarization had salient features that were unexplained by RFHD. This result suggests that one of the key assumptions of RFHD, an oblique dipole, is not precise enough to reproduce observational variability. Indeed, magnetic field reconstruction from surface polarimetry (e.g. τ Sco: Donati et al. (2006), HD37776: Kochukhov et al. (2011)) shows significant non-dipole components in several massive star magnetic fields. Additionally, the Magnetism in Massive Stars (MiMeS) program (Petit et al. 2013; Wade et al. 2016) found several other stars with multipole components in their magnetic fields. Together, these results indicate that one of the next

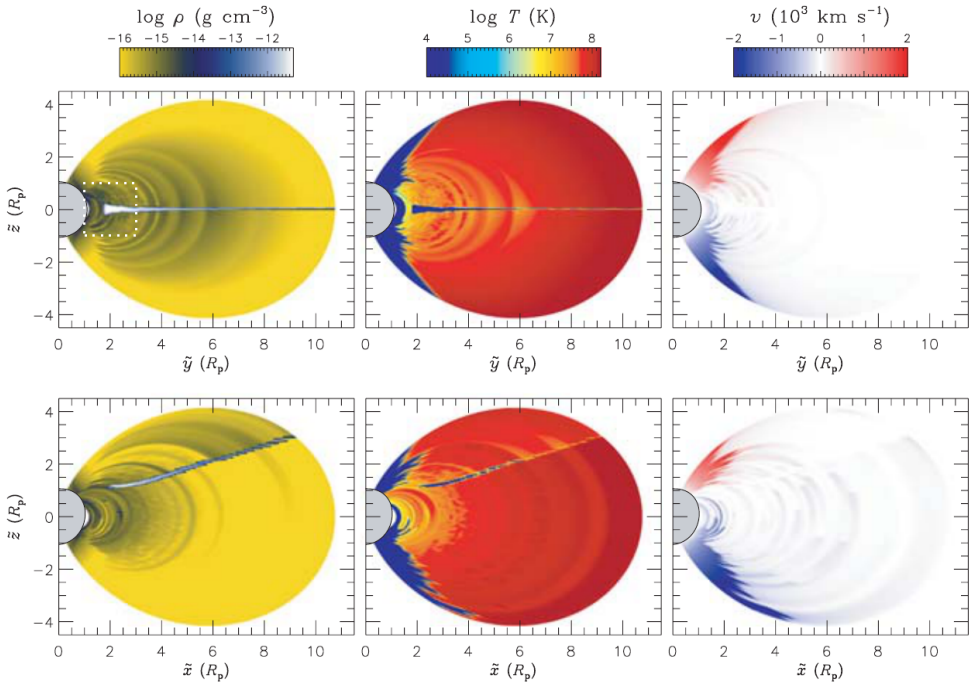


Figure 1.7 : The state of the flow at the end of the RFHD simulation of an oblique magnetic dipole, showing the density ρ, temperature T and velocity v in two different meridional planes (top and bottom rows). The oblate star is indicated in grey at the left-hand side of each image. Outside the regions threaded by simulation field lines (i.e. for dipole line apex radii $r_m < 1.2R_*$ and $r_m > 11.2R_*$), the images are left blank. Figure taken from Townsend et al. (2007).

steps in magnetosphere modeling is to introduce unusual magnetic field topologies.

1.3 Massive Star Winds (without magnetic fields)

In massive stars, winds are driven by the absorption of ultraviolet resonance lines, e.g. SiIV, CIV, NV, or SVI (Lucy & Solomon 1970). There are upwards of several hundred lines which contribute to the wind (Abbott 1982; Puls et al. 2000), all of which need to be summed to get the overall line-driven acceleration. In preparation for the discussion of such winds in magnetic massive stars later in this thesis, we present here one way of parameterizing the overall radiative line-driving: the $\bar{Q}$ formalism of Gayley (1995). We use this parameterization throughout the thesis.

1.3.1 Acceleration from a single line, point source

First, we begin with the general form of the radiative acceleration (e.g. Shu 1991)

$$\mathbf{g}_{\rm rad} = \frac{1}{c} \oint \int_0^\infty \kappa_\nu I_\nu d\nu \; \hat{\mathbf{n}} d\Omega \; , \tag{1.8}$$

where we calculate the acceleration of a material with opacity κ_ν from a source intensity I_ν at frequency ν along a ray $\hat{\mathbf{n}}$ from the stellar surface to the point in the wind. The acceleration is obtained by integrating over all possible frequencies and the solid angle Ω subtended by the source surface. In the simplest case, we can take a point-source star, such that its intensity is

$$I_* = \int_0^\infty I_\nu d\nu = \frac{L_*}{4\pi r^2} \delta(\hat{\mathbf{n}} - \hat{\mathbf{r}}), \tag{1.9}$$

with δ the Dirac delta function.

Since the effective temperatures of massive stars are greater than $1{\times}10^4$ K, their winds are effectively fully ionized. As a result, the continuum opacity is dominated by electron scattering. For acceleration via electron scattering, its opacity κ_e is both grey

(independent of frequency) and isotropic (independent of solid angle), yielding the result

$$\mathbf{g}_e = \frac{\kappa_e}{c}\,\mathbf{F}_{\rm rad}, \tag{1.10}$$

where we can calculate the bolometric flux from a point-source star using the stellar luminosity L_*:

$$\mathbf{F}_{\rm rad} = \oint \int_0^\infty I_\nu d\nu\ \hat{\mathbf{n}} d\Omega\ = \frac{L_*}{4\pi r^2 c}\hat{\mathbf{r}}\ . \tag{1.11}$$

We can quantify the relative strength of electron scattering versus gravity with the Eddington parameter

$$\Gamma_{\rm el} = \frac{g_e}{g_{\rm grav}} = \frac{\kappa_e L_*}{4\pi G M_* c} \tag{1.12}$$

with M_* the stellar mass. In massive stars, $\Gamma_{\rm el}$ typically ranges from $\lesssim 0.01$ for B stars to $0.1 - 0.5$ for O stars. Since gravity still dominates the electron scattering, another force must be driving these massive star winds!

This force is provided by the line-scattering of bound electrons in atoms. Even though there are very few bound electrons left in the hot wind, the resonant nature of bound-bound interactions produces a large line opacity relative to κ_e. The limited flux within the line center frequency ν_0 is quickly extinguished by the bound electrons; however, the wind acceleration Doppler shifts the observed source flux and provides new photons to interact with. These effects combine to make radiative line-driving hundreds, even thousands, of times stronger than the electron scattering force and allows the wind to defy gravity and flow out into space.

We can quantify the acceleration from a single line with

$$\mathbf{g}_{\rm line} = \frac{\kappa_L}{c} \oint \int_0^\infty \phi(\nu - \nu') I_\nu d\nu\ \hat{\mathbf{n}}\ d\Omega, \tag{1.13}$$

where κ_L is the direction- and frequency-independent opacity at line center and ϕ is a normalized function reflecting the shape of the line-profile. For a wind in motion, the frequency of radiation seen depends on the Doppler shift along the line of sight:

$$\nu' = \nu_0 \left(1 + \frac{\hat{\mathbf{n}} \cdot \mathbf{v}}{c}\right). \tag{1.14}$$

We make the standard change of variables to the dimensionless frequency displacement at line center

$$x \equiv \frac{\nu - \nu_0}{\Delta\nu_D} \tag{1.15}$$

with the Doppler width $\Delta\nu_D = \nu_0 v_{\rm th}/c$ and $v_{\rm th}$ the ion thermal velocity in the wind. We rewrite Equation 1.13 as

$$\mathbf{g}_{\rm line} = \frac{\kappa_L \Delta\nu_D}{c} \oint \int_{-\infty}^{\infty} \phi' \left(x - \frac{\hat{\mathbf{n}} \cdot \mathbf{v}}{v_{\rm th}}\right) I_\nu dx \; \hat{\mathbf{n}} \; d\Omega, \tag{1.16}$$

with ϕ' a normalized line profile function in the variable x. For a point star, the only relevant rays $\hat{\mathbf{n}}$ in Equation 1.16 are purely radial. In this "radial-streaming" limit, the line-of-sight velocity simplifies to $\hat{\mathbf{n}} \cdot \hat{\mathbf{v}} = v_r$.

Despite the appearance of the Doppler width here, the radiative driving does *not* depend on $v_{\rm th}$. The line center opacity can be expressed as

$$\kappa_L = \frac{\pi e^2}{m_e c} \frac{1}{\Delta\nu_D} \frac{n_l}{\rho} f_l, \tag{1.17}$$

where e and m_e are the charge and mass of the electron, n_l the number density of atoms which can be excited by the line transition, and f_l the quantum mechanical oscillator strength of the line (Puls et al. 1993). Since $\kappa_L \propto 1/\Delta\nu_D$, the $\Delta\nu_D$ term in Equation 1.16 has no effect on the line acceleration.

For optically-thin lines, the intensity is not significantly attenuated at line center, allowing us to approximate the point-star optically-thin line acceleration as

$$\mathbf{g}_{\rm thin} \approx \frac{\kappa_L \Delta\nu_D L_{*,\nu}}{4\pi r^2 c} \hat{\mathbf{r}}, \tag{1.18}$$

with $L_{*,\nu}$ the luminosity at that particular line frequency ν. We can relate this acceleration to the electron scattering acceleration by defining the weighting factor

$$w_\nu \equiv \nu_0 L_{*,\nu}/L_*, \tag{1.19}$$

and the line-strength ratio

$$q \equiv \frac{\kappa_L}{\kappa_e}\frac{v_{\rm th}}{c}. \tag{1.20}$$

Using the definition of the electron scattering opacity,

$$\kappa_e = \frac{\sigma_T n_e}{\rho}, \tag{1.21}$$

with $\sigma_T = \frac{8}{3}\pi r_e^2$, r_e the classical electron radius, and n_e the free electron number density, we can calculate

$$q = \frac{3}{8}\frac{\lambda_0}{r_e}\frac{n_l}{n_e}f_l \tag{1.22}$$

for a line with wavelength λ_0.

With these definitions, we easily see that

$$g_{\rm thin} = w_\nu q g_e. \tag{1.23}$$

$w_\nu q$ thus represents the maximum boost in acceleration that a given line provides relative to the continuum electron scattering acceleration. Typical values of q are around $10^2 - 10^3$ (Gayley 1995). Thus, for lines near the peak of the flux spectrum ($w_\nu \approx 1$), line acceleration can be hundreds of times stronger than electron scattering!

Since the wind may not be optically-thin, self-absorption within the spectral line will reduce the line-driving. The optical depth for a line, in a moving medium along a particular ray $\hat{\mathbf{n}}$ with path length s, is

$$\tau_\nu = \int_0^s \kappa_L \rho(s) \phi' \left(x - \frac{\hat{\mathbf{n}}\cdot\mathbf{v}}{v_{\rm th}} \right) ds, \tag{1.24}$$

where the velocity $\mathbf{v} = \mathbf{v}(\mathbf{r}')$ is evaluated at some instantaneous position along the ray, $\mathbf{r}' = \mathbf{r}_0 + s\hat{\mathbf{n}}$. $\mathbf{r}_0$ is defined as the position vector at the location on the stellar surface from which the tracing ray begins ($s = 0$).

Fortunately, we do not have to solve this complex integral over the entire ray length; Sobolev (1960) demonstrated that this integral can be locally calculated within the "Sobolev" length

$$L_{\rm sob} \approx \frac{v_{\rm th}}{dv_n/dn}, \tag{1.25}$$

with dv_n/dn the velocity gradient along the line-of-sight of a ray $\hat{\mathbf{n}}$. Sobolev showed that a photon with frequency $\nu \gtrsim \nu_0$ can only interact with a line over a resonance zone of length $L_{\rm sob}$. Thus, for winds with hydrodynamical scale heights $H \approx v/(dv_n/dn) \gg L_{\rm sob}$ (or $v \gg v_{\rm th}$), κ_L and ρ can be calculated from *local* conditions. We can take them out of the optical depth integral:

$$\tau_\nu = \kappa_L \rho(s) \int_0^s \phi' \left(x - \frac{\hat{\mathbf{n}} \cdot \mathbf{v}}{v_{\rm th}} \right) ds. \tag{1.26}$$

Changing the variable s into frequency space via

$$x' = x - \frac{\hat{\mathbf{n}} \cdot \mathbf{v}}{v_{\rm th}}, \tag{1.27}$$

$$dx' = -\frac{1}{v_{\rm th}} d\left[\hat{\mathbf{n}} \cdot \mathbf{v}(\mathbf{r}')\right] = -\frac{1}{v_{\rm th}} \hat{\mathbf{n}} \cdot \nabla(\hat{\mathbf{n}} \cdot \mathbf{v})\, ds', \tag{1.28}$$

yields the integral

$$\tau_\nu = \frac{\kappa_L \rho v_{\rm th}}{\hat{\mathbf{n}} \cdot \nabla(\hat{\mathbf{n}} \cdot \mathbf{v})} \int_{x - \frac{\hat{\mathbf{n}} \cdot \mathbf{v}}{v_{\rm th}}}^{\infty} \phi'(x')\, dx', \tag{1.29}$$

with $\hat{\mathbf{n}} \cdot \nabla(\hat{\mathbf{n}} \cdot \mathbf{v}) = dv_n/dn$. From Equation 1.29, we can define the optical depth along $\hat{\mathbf{n}}$

$$\tau_n \equiv \frac{\kappa_L \rho v_{\rm th}}{dv_n/dn} = \frac{q \kappa_e \rho c}{dv_n/dn}, \tag{1.30}$$

and define the integral as

$$\Phi' \equiv \int_{x-\frac{\hat{\mathbf{n}}\cdot\mathbf{v}}{v_{\rm th}}}^{\infty} \phi'\left(x'\right) dx', \tag{1.31}$$

so that our stellar intensity, after correcting for self-absorption, is

$$I_\nu = I_{\nu,*} e^{-\tau_n \Phi'}. \tag{1.32}$$

We can apply this to Equation 1.16:

$$\mathbf{g}_{\rm line} = \frac{\kappa_L \Delta\nu_D}{c} \oint \int_{-\infty}^{\infty} \phi'\left(x - \frac{\hat{\mathbf{n}}\cdot\mathbf{v}}{v_{\rm th}}\right) I_{\nu,*} e^{-\tau_n \Phi'} dx\ \hat{\mathbf{n}}\ d\Omega. \tag{1.33}$$

Since most spectral lines have limited extents in frequency, we assume that $I_{\nu,*}$ is constant over the line. This allows us to take the intensity out of the integral over the line profile frequency variable x. The integral then becomes

$$\mathbf{g}_{\rm line} = \frac{\kappa_L \Delta\nu_D}{c} \oint I_{*,\nu}\ \hat{\mathbf{n}}\ d\Omega \int_{x=-\infty}^{x=+\infty} e^{-\tau_n \Phi'} d\Phi'\ , \tag{1.34}$$

which we can solve analytically. For a point source, we obtain

$$\begin{aligned} \mathbf{g}_{\rm line} &= \frac{\kappa_L \Delta\nu_D L_{*,\nu}}{4\pi r^2 c} \frac{1 - e^{\tau_r}}{\tau_r}\ \hat{\mathbf{r}} \\ &= \mathbf{g}_{\rm thin} \frac{1 - e^{\tau_r}}{\tau_r}, \end{aligned} \tag{1.35}$$

with τ_r equivalent to τ_n evaluated for a radial line-of-sight velocity $dv_n/dn = dv_r/dr$.

There will also be acceleration resulting from scattering of photons, but Castor (1974) found that the ratio of the scattering acceleration to the absorption acceleration is on the order of $v_{\rm th}/v_{\rm wind} \approx 1 \times 10^{-2}$. We will ignore scattering in this thesis, though we note that Sundqvist & Owocki (2015) have begun to look at scattering effects on radiative line-driving.

1.3.2 Acceleration from many lines

As Abbott (1982) demonstrated, a summation over all of the possible resonance lines is critical in obtaining an accurate radiative line acceleration. Following Gayley (1995), we define a flux-weighted number distribution over line strength q, with an exponential cutoff:

$$\frac{dN}{dq} = \frac{\bar{Q}}{\Gamma(\alpha) Q_o^2} \left(\frac{q}{Q_0} \right)^{\alpha-2} e^{-q/Q_0}, \tag{1.36}$$

with Γ the gamma function. $\bar{Q}$ can be thought of as a flux and population weighted q, Q_0 is the "cutoff" line strength, and α is a power law parameter set by the line population at a certain wind temperature (Abbott 1982; Puls et al. 2000). The total line acceleration is then given by

$$\begin{aligned} \mathbf{g}_{\rm rad} &= \int_0^\infty \mathbf{g}_{\rm line} \frac{dN}{dq}\, dq \\ &= \frac{\mathbf{g}_e}{t} \int_0^\infty \frac{dN}{dq} (1 - e^{-qt})\, dq, \end{aligned} \tag{1.37}$$

where we have accounted for the effect of w_ν in our flux-weighted number distribution and define t as the τ_n when $q = 1$. After solving the integral, we obtain the total line acceleration

$$\mathbf{g}_{\rm rad} = \frac{\mathbf{g}_e \bar{Q}}{(1-\alpha) Q_0 t} \left[(1 + Q_0 t)^{1-\alpha} - 1 \right]. \tag{1.38}$$

From Gayley (1995), we take $\bar{Q} \approx Q_0$ for early-type massive stars; this approximation leads to our final form of the radiative driving (Owocki et al. 1988) for a point source star ($\tau_n = \tau_r$):

$$\mathbf{g}_{\rm rad} = \frac{\kappa_e \bar{Q} L_*}{4\pi r^2 c} \frac{(1 + \tau_{\rm sob})^{1-\alpha} - 1}{(1-\alpha)\tau_{\rm sob}}\, \hat{\mathbf{r}}\,, \tag{1.39}$$

with our Sobolev optical depth

$$\tau_{\rm sob} \equiv \frac{c \rho \kappa_e \bar{Q}}{dv_r/dr}. \tag{1.40}$$

We note that our assumption of $\bar{Q} \approx Q_0$ does not hold too well for B stars (Puls et al. 2000), but we will save this complication for future research.

The classical Castor et al. (1975) result did not use an exponential cutoff in the number distribution, instead using a distribution of the form

$$\frac{dN}{dq} = \frac{1}{\Gamma(\alpha)\bar{Q}} \left(\frac{q}{\bar{Q}} \right)^{\alpha-2}, \tag{1.41}$$

which leads to a total acceleration

$$\mathbf{g}_{\rm rad} = \frac{\kappa_e \bar{Q} L_*}{4\pi r^2 c(1-\alpha)} \tau_{\rm sob}^{-\alpha} \, \hat{\mathbf{r}} \, . \tag{1.42}$$

This same result can be obtained by assuming an optically-thick wind ($\tau_{\rm sob} \gg 1$) in Equation 1.39.

1.3.3 Non-magnetic mass-loss rate

Equation 1.42 is the classical form of the radiative line-acceleration as derived by Castor et al. (1975). We can include this in the equation of motion for a steady-state, spherically symmetric line-driven wind:

$$v_r \frac{dv_r}{dr} = g_{\rm rad} - \frac{GM_*(1-\Gamma_{\rm el})}{r^2}, \tag{1.43}$$

where we take $g_{\rm rad}$ as the magntiude of $\mathbf{g}_{\rm rad}$. In defining this equation, we have made the assumption that the sound speed is much lower than the flow velocity; all of its associated terms are neglected. Since the interaction between flow acceleration and line-driving (which itself depends on the flow acceleration) is tempered by gravity, we introduce a gravitationally scaled inertial acceleration (e.g. Owocki 2004)

$$w' \equiv \frac{r^2 v v'}{GM_*(1-\Gamma_{\rm el})} \tag{1.44}$$

and a inverse-radius coordinate $x \equiv 1 - R_*/r$. With these definitions, we note that $w' = dw/dx$ for a ratio $w \equiv v^2/v_{\rm esc}^2$ representing the ratio of wind kinetic energy to the

wind escape velocity $v_{\rm esc}^2 = 2GM_*(1-\Gamma_{\rm el})/R_*$. With this transformation, Equation 1.43 becomes

$$w' = Cw'^{\alpha} - 1, \tag{1.45}$$

with an eigenvalue

$$C = \frac{1}{1-\alpha}\left(\frac{\bar{Q}\Gamma_{\rm el}}{1-\Gamma_{\rm el}}\right)^{1-\alpha}\left(\frac{L_*}{\dot{M}c^2}\right)^{\alpha} \tag{1.46}$$

controlling the resulting behavior of the wind. We have substituted in the spherically-symmetric mass loss rate $\dot{M} = 4\pi r^2 \rho v$ in place of the density.

Figure 1.8 illustrates the behavior of Equation 1.45 for various values of C. Depending on the mass-loss rate chosen, this scaled equation of motion can have 0, 1, or 2 solutions.

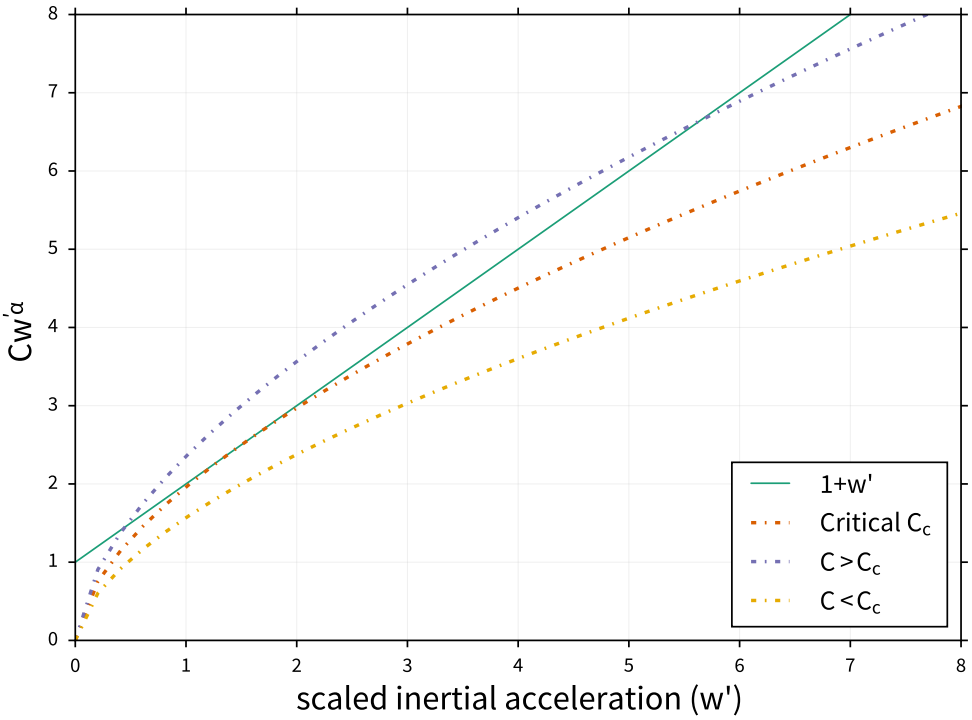


Figure 1.8 : Behavior of the scaled steady-state, spherically-symmetric line-driven wind equation of motion (Equation 1.45). Solutions to the equation of motion exist at intersections of the solid line with the dot-dashed lines. There is a critical value C_c for which only one possible solution exists; this sets the maximum allowed mass-loss rate.

Since $C \propto \dot{M}^{-\alpha}$, the minimum possible value of C where there is a solution corresponds to a maximum, "critical" mass-loss rate from the star. We can easily find this critical mass-loss by noting that, for the critical C, the line $1 + w'$ must intersect Cw'^{α} at a tangent. The critical C is thus given by

$$C_c = \frac{(1-\alpha)^{\alpha-1}}{\alpha^{\alpha}} \tag{1.47}$$

and we can easily solve for the so-called "CAK mass-loss rate" (Castor et al. 1975):

$$\dot{M}_{\rm CAK} \equiv \frac{\alpha}{1-\alpha}\left(\frac{\bar{Q}\Gamma_{\rm el}}{1-\Gamma_{\rm el}}\right)^{\frac{1-\alpha}{\alpha}} \frac{L_*}{c^2}. \tag{1.48}$$

The velocity structure of this simple stellar wind is easily found by integrating w' from the surface to infinity, yielding the CAK velocity law

$$v(r) = v_{\infty}\left(1 - R_*/r\right)^{1/2} \tag{1.49}$$

with the terminal velocity $v_{\infty} = v_{\rm esc}\sqrt{\alpha/(1-\alpha)}$. In general, winds from non-magnetic stars can be represented by a beta-velocity law $v = v_{\infty}(1 - R_*/r)^{\beta}$, with $\beta \approx 0.8 - 1$ (Friend & Abbott 1986; Pauldrach et al. 1986).

1.3.4 Finite-disk correction

In the above discussion, we have assumed that the stellar intensity comes from a point source (Equation 1.9). This allows us to simplify the calcuation of the line-of-sight velocity gradient to be purely radial. In reality, however, the star does not appear as a point from the perspective of the wind. Instead, it appears as an extended, finite disk over some solid angle $d\Omega' = \sin\theta' d\theta' d\phi'$, where θ' and ϕ' are measured relative to the wind position.

In this case, the stellar intensity can be expressed in terms of the stellar flux as

$$I_* = \frac{L_*}{4\pi^2 R_*^2}, \tag{1.50}$$

assuming an uniform distribution over the stellar surface (i.e. no limb or gravity darkening).

The line-of-sight velocity gradient dv_n/dn can be evaluated, and has been given for various coordinate systems by Batchelor (2000) (also see Equation A.12). This gradient is quite complicated to evaluate in the general case, but, for a purely radial outflow, it greatly simplifies to

$$dv_n/dn = \mu'^2 \frac{dv_r}{dr} + (1-\mu'^2)\frac{v_r}{r}, \tag{1.51}$$

with $\mu' = \cos\theta'$.

With these definitions, the radiative acceleration (Equation 1.37) becomes

$$\mathbf{g}_{\rm rad} = \oint \frac{\kappa_e I_*}{ct}\hat{\mathbf{n}}\; d\mu' d\phi' \int_0^\infty \frac{dN}{dq}(1-e^{-qt})\; dq, \tag{1.52}$$

with t now defined with our particular line-of-sight velocity gradient

$$t = \frac{\kappa_e \rho c}{\mu'^2 \frac{dv_r}{dr} + (1-\mu'^2)\frac{v_r}{r}}. \tag{1.53}$$

Using the pure power-law number distribution dN/dq (Equation 1.41), and simplifying over ϕ', we get the integral

$$g_{\rm rad,r} = \frac{2\kappa_e \bar{Q} L_*}{4\pi R_*^2 c(1-\alpha)} \frac{1}{(c\rho\kappa_e\bar{Q})^\alpha} \int_{\mu_*}^1 \left[\mu'^2 \frac{dv_r}{dr} + (1-\mu'^2)\frac{v_r}{r}\right]^\alpha \mu'\; d\mu', \tag{1.54}$$

with $\mu_* = \sqrt{1-R_*/r}$ marking the edge of the stellar limb as seen from the wind. For a radial wind, the integral is symmetric in the θ and ϕ components of $\hat{\mathbf{n}}$ and the angular components of $\mathbf{g}_{\rm rad}$ evaluate to zero. The radial component of $\hat{\mathbf{n}}$ is simply μ'.

After defining $\sigma = d\ln v_r / d\ln r - 1$, we obtain

$$\mathbf{g}_{\rm rad} = \frac{\kappa_e \bar{Q} L_*}{4\pi R_*^2 c(1-\alpha)} \tau_{\rm sob}^{-\alpha} \frac{(1+\sigma)^{1+\alpha} - (1+\sigma\mu_*^2)^{1+\alpha}}{(1+\sigma)^\alpha \sigma(1+\alpha)}. \tag{1.55}$$

We can then define a "finite-disk-correction factor" by dividing Equation 1.55 by Equation 1.42:

$$f \equiv \frac{\mathbf{g}_{\rm rad,disk}}{\mathbf{g}_{\rm rad,point}} = \frac{(1+\sigma)^{1+\alpha} - (1+\sigma\mu_*^2)^{1+\alpha}}{(1+\sigma)^{\alpha}\sigma(1+\alpha)(1-\mu_*^2)}, \tag{1.56}$$

which is the classical (Castor et al. 1975) result. We can finally update our point-source radiative acceleration with this factor:

$$\mathbf{g}_{\rm rad} = f \ \frac{\kappa_e \bar{Q} L_*}{4\pi r^2 c(1-\alpha)} \tau_{\rm sob}^{-\alpha} \ \hat{\mathbf{r}} \ . \tag{1.57}$$

1.4 Scientific advancement

In the field of planetary magnetospheres, our goal is to understand how magnetic reconnection affects the state of the global magnetosphere. Current computational codes cannot efficiently simulate the global magnetosphere and resolve the small length scales required to investigate reconnection events. In this thesis, I develop a three-dimensional, large-scale Hall MHD code able to run on many graphics processors which will be able to study reconnection in planetary magnetospheres. In Chapter 2, I discuss my experience in creating a simple, two-dimensional ideal MHD code which runs on one GPU. I expand this code in Chapter 3 by including the Hall term, adding a third dimension, and allowing it to run on muliple GPUs. Benchmarking and scaling results are presented and demonstrate that the code will work on many GPUs. This bodes well for future planetary magnetosphere simulations.

In the area of massive-star magnetospheres, I make advances in modeling the complex interactions between field and flow by considering how magnetic fields change the state of the wind. The scalings presented in Section 1.2 are based on η_* (Equation 1.4), which

is itself based on non-magnetic wind values for the mass-loss rate and terminal velocity. I ask the question: How does the magnetic field change $\dot{M}$ and v_∞?, and make the first strides towards finding an answer in this thesis. In Chapter 4, I improve on the Rigid-Field Hydrodynamic model presented in Section 1.2.2 by allowing for non-dipolar, even *arbitrary*, magnetic topologies provided that their field lines are continuous and do not intersect. Later in the same chapter, I then analyze the updated hydrodynamic equations and derive scaling relations that describe how a dipolar magnetic field changes the mass-flux across the stellar surface. My analysis also includes an updated velocity structure for magnetospheric lines. Chapter 5 extends the analysis by correcting for the simplification of a point source intensity with a finite-disk correction factor. The new mass-flux scaling relation and velocity structure are compared to the previous results to characterize the influence of the correction factor. In Chapter 6, I update the XADM model for magnetospheric X-ray emission with my results from Chapter 5. I discuss how the XADM results change as well as implications for future models. Combined, these three chapters represent a significant leap in our understanding of magnetic massive star line-driven winds and magnetospheres.

Beyond this thesis, the tools developed here will allow us to study how stellar winds interact with large-scale magnetic fields. This will provide insight into the interplay between winds and fields and allow us to better understand the size, structure, and emission of magnetospheres, around both planets and massive-stars.

References

Abbott, D. C. 1982, ApJ, 259, 282

Antiochos, S. K., DeVore, C. R., & Klimchuk, J. A. 1999, ApJ, 510, 485

Babcock, H. W. 1958, ApJ, 128, 228

Babel, J., & Montmerle, T. 1997, A&A, 323, 121

Batchelor, G. K. 2000, An Introduction to Fluid Dynamics, 635

Birn, J., Drake, J. F., Shay, M. A., et al. 2001, J. Geophys. Res., 106, 3715

Biskamp, D., Schwarz, E., & Drake, J. F. 1995, Physical Review Letters, 75, 3850

Boozer, A. H. 2002, Physical Review Letters, 88, 215005

Borra, E. F., & Landstreet, J. 1980, ApJS, 42, 421

Cargill, P. J., & Klimchuk, J. A. 1997, ApJ, 478, 799

Castor, J. I., Abbott, D. C., & Klein, R. I. 1975, ApJ, 195, 157

Castor, J. L. 1974, MNRAS, 169, 279

Chandra, P., Wade, G. A., Sundqvist, J. O., et al. 2015, MNRAS, 452, 1245

Charbonneau, P. 2010, Living Reviews in Solar Physics, 7, doi:10.12942/lrsp-2010-3

Cohen, O., Drake, J. J., Glocer, A., et al. 2014, ApJ, 790, 57

Courant, R., Friedrichs, K., & Lewy, H. 1928, Mathematische Annalen, 100, 32

Crutcher, R. M. 2012, ARA&A, 50, 29

Daughton, W., & Karimabadi, H. 2007, Physics of Plasmas, 14, 072303

Donati, J.-F., & Landstreet, J. D. 2009, ARA&A, 47, 333

Donati, J.-F., Howarth, I. D., Jardine, M. M., et al. 2006, MNRAS, 370, 629

Dorelli, J. C., Bhattacharjee, A., & Raeder, J. 2007, Journal of Geophysical Research (Space Physics), 112, A02202

Dorelli, J. C., Glocer, A., Collinson, G., & Tóth, G. 2015, Journal of Geophysical Research (Space Physics), 120, 5377

Duez, V., Braithwaite, J., & Mathis, S. 2010, ApJ, 724, L34

Duez, V., & Mathis, S. 2010, A&A, 517, A58

Eikenberry, S. S., Chojnowski, S. D., Wisniewski, J., et al. 2014, ApJ, 784, L30

Finn, J. M. 2006, Nature Physics, 2, 445

Friend, D. B., & Abbott, D. C. 1986, ApJ, 311, 701

Fuselier, S. A., Lewis, W. S., Schiff, C., et al. 2016, Space Sci. Rev., 199, 77

Gagné, M., Oksala, M., Cohen, D., et al. 2005, ApJ, 628, 986

Gayley, K. G. 1995, ApJ, 454, 410

Glocer, A., Dorelli, J., Toth, G., Komar, C. M., & Cassak, P. 2014, AGU Fall Meeting Abstracts

Groote, D., & Hunger, K. 1982, A&A, 116, 64

Haerendel, G. 1987, in ESA Special Publication, Vol. 275, Small Scale Plasma Processes in the Solar Chromosphere/Corona, Interplanetary Medium and Planetary Magnetospheres, ed. B. Battrick & E. J. Rolfe

Hall, J. C. 2008, Living Reviews in Solar Physics, 5, doi:10.12942/lrsp-2008-2

Hesse, M., & Schindler, K. 1988, J. Geophys. Res., 93, 5559

Hesse, M., Aunai, N., Birn, J., et al. 2016, Space Sci. Rev., 199, 577

Jia, X., Walker, R. J., Kivelson, M. G., Khurana, K. K., & Linker, J. A. 2009, Journal of Geophysical Research (Space Physics), 114, A09209

Kochukhov, O., Lundin, A., Romanyuk, I., & Kudryavtsev, D. 2011, ApJ, 726, 24

Landstreet, J., & Borra, E. 1978, ApJ, 224, L5

Lazarian, A., & Vishniac, E. T. 1999, ApJ, 517, 700

Linsky, J., Drake, S., & Bastian, T. 1992, ApJ, 393, 341

Lucy, L. B., & Solomon, P. M. 1970, ApJ, 159, 879

McKee, C. F., & Ostriker, E. C. 2007, ARA&A, 45, 565

Michel, F., & Sturrock, P. 1974, Planet. Space Sci., 22, 1501

Mozer, F. S., Phan, T. D., & Bale, S. D. 2003, Physics of Plasmas, 10, 2480

Nakajima, R. 1985, Ap&SS, 116, 285

Nazé, Y., Petit, V., Rinbrand, M., et al. 2014, ApJS, 215, 10

Oksala, M., Wade, G., Townsend, R., et al. 2012, MNRAS, 419, 959

Owocki, S. 2004, in EAS Publications Series, Vol. 13, EAS Publications Series, ed. M. Heydari-Malayeri, P. Stee, & J.-P. Zahn, 163–250

Owocki, S. P., Castor, J. I., & Rybicki, G. B. 1988, ApJ, 335, 914

Parker, E. N. 1957, J. Geophys. Res., 62, 509

Pauldrach, A., Puls, J., & Kudritzki, R. P. 1986, A&A, 164, 86

Petit, V., Owocki, S. P., Wade, G. A., et al. 2013, MNRAS, 429, 398

Priest, E., & Forbes, T. 2000, Magnetic Reconnection, 612

Priest, E. R., & Démoulin, P. 1995, J. Geophys. Res., 100, 23443

Puls, J., Owocki, S. P., & Fullerton, A. W. 1993, A&A, 279, 457

Puls, J., Springmann, U., & Lennon, M. 2000, A&AS, 141, 23

Ryu, D., Schleicher, D. R. G., Treumann, R. A., Tsagas, C. G., & Widrow, L. M. 2012, Space Sci. Rev., 166, 1

Schindler, K., Hesse, M., & Birn, J. 1988, J. Geophys. Res., 93, 5547

Shay, M. A., Drake, J. F., Rogers, B. N., & Denton, R. E. 1999, Geophys. Res. Lett., 26, 2163

Shore, S., & Brown, D. 1990, ApJ, 365, 665

Short, C., & Bolton, C. 1994, in IAU Symposium, Vol. 162, Pulsation; Rotation; and Mass Loss in Early-Type Stars, ed. L. A. Balona, H. F. Henrichs, & J. M. Le Contel, 171

Shu, F. H. 1991, Physics of Astrophysics, Vol. I (University Science Books)

Sobolev, V. V. 1960, Moving envelopes of stars

Sonnerup, B. U. O., Paschmann, G., Papamastorakis, I., et al. 1981, J. Geophys. Res., 86, 10049

Spruit, H. C. 2002, A&A, 381, 923

Stibbs, D. 1950, MNRAS, 110, 395

Sundqvist, J. O., & Owocki, S. P. 2015, MNRAS, 453, 3428

Sweet, P. A. 1958, in IAU Symposium, Vol. 6, Electromagnetic Phenomena in Cosmical Physics, ed. B. Lehnert, 123

Tóth, G., Ma, Y., & Gombosi, T. I. 2008, Journal of Computational Physics, 227, 6967

Townsend, R. H. D., & Owocki, S. P. 2005, MNRAS, 357, 251

Townsend, R. H. D., Owocki, S. P., & Groote, D. 2005, ApJ, 630, L81

Townsend, R. H. D., Owocki, S. P., & Ud-Doula, A. 2007, MNRAS, 382, 139

ud-Doula, A., Owocki, S., Townsend, R., Petit, V., & Cohen, D. 2014, MNRAS, 441, 3600

ud-Doula, A., & Owocki, S. P. 2002, ApJ, 576, 413

ud-Doula, A., Owocki, S. P., & Townsend, R. H. D. 2008, MNRAS, 385, 97

—. 2009, MNRAS, 392, 1022

Vasyliunas, V. M. 1975, Reviews of Geophysics and Space Physics, 13, 303

Vidotto, A. A., Fares, R., Jardine, M., Moutou, C., & Donati, J.-F. 2015, MNRAS, 449, 4117

Wade, G. A., Neiner, C., Alecian, E., et al. 2016, MNRAS, 456, 2

Walborn, N. R. 1974, ApJ, 191, L95

Yamada, M., Kulsrud, R., & Ji, H. 2010, Reviews of Modern Physics, 82, 603

Zweibel, E. G., & Yamada, M. 2009, ARA&A, 47, 291

Chapter 2

A simple GPU-accelerated two-dimensional MUSCL-Hancock solver for ideal magnetohydrodynamics

A version of this chapter has previously appeared in the Journal of Computational Physics

Bard & Dorelli 2014 J. Comp. Phys., 259, 444

Faster! Faster would be better!

Malcolm Reynolds

Abstract

We describe our experience using NVIDIA's CUDA (Compute Unified Device Architecture) C programming environment to implement a two-dimensional second-order MUSCL-Hancock ideal magnetohydrodynamics (MHD) solver on a GTX 480 Graphics Processing Unit (GPU). Taking a simple approach in which the MHD variables are stored exclusively in the global memory of the GTX 480 and accessed in a cache-friendly manner (without further optimizing memory access by, for example, staging data in the GPU's faster shared memory), we achieved a maximum speed-up of ≈ 126 for a 1024^2 grid relative to the sequential C code running on a single Intel i7 930 (2.8 GHz) core. This speedup is consistent with simple estimates based on the known floating point performance, memory throughput and parallel processing capacity of the GTX 480.

2.1 Introduction

The last several years have witnessed a dramatic increase in the use of Graphics Processing Units (GPUs) to accelerate scientific computing applications. The primary catalyst for this surge in GPU computing was NVIDIA's public release of the CUDA (Compute Unified Device Architecture) programming environment (Kirk & Hwu 2010) in 2007. The introduction of CUDA provided the scientific programmer with easy access to the SIMT (Single Instruction Multiple Thread) architecture underlying modern NVIDIA GPUs (previously accessible only through cumbersome graphics APIs like OpenGL) via an intuitive extension of ANSI C. The result has been a proliferation of GPU-accelerated applications in such diverse areas as N-body simulation (Belleman et al. 2008; Gaburov et al. 2012), signal processing (Harris et al. 2008), molecular dynamics simulation (Stone et al. 2007; Friedrichs et al. 2009) and computational fluid dynamics (CFD) (Brandvik & Pullan 2008; Schive et al. 2010). Speedups reported in the literature depend on the application, but one can expect, on average, an order of magnitude performance increase "out of the box" using CUDA.

Explicit finite volume schemes are particularly amenable to GPU acceleration due to the natural manner in which the computational mesh maps to the SIMT architecture. In a typical implementation, chunks of the mesh are handed out by the CPU to groups of threads executing on the GPU, with each thread responsible for updating a single computational cell using data in nearby memory locations. Each cell update involves many instructions (some combination of arithmetic operations and memory reads/writes), and the GPU achieves its performance advantage over the CPU through a combination of parallel execution (hundreds of threads may execute instructions simultaneously on multiple GPU "cores")

and latency hiding (if a thread stalls while executing a slow instruction, such as accessing off-chip memory, execution may be switched to one of thousands other "active threads" ready for execution).

While the speedups of GPU-accelerated finite volume fluid codes reported in the literature are impressive (e.g., Schive et al. (2010) report a speedup of ∼100x on a uniform mesh with several million cells), GPUs still have some significant limitations. The primary appeal of GPU computing is the promise of achieving the computational equivalent of 10-100 latest-generation CPU cores with a single graphics card for a fraction of the cost. Unfortunately, limited off-chip memory, absence of cache memory and poor double precision performance (or no double precision capability at all) on inexpensive consumer-grade graphics cards present barriers for the computational scientist interested in accelerating even a moderately sized (say, several million computational cells) CFD problem. While NVIDIA's Tesla cards – designed with high performance computing applications in mind – remove some of these constraints, they are significantly more expensive than the consumer-grade products (e.g., compare the ∼$2500 price tag of the Tesla M2090 to the several hundred dollar cost of the GTX 480[1]).

GPU programming effort presents another obstacle for the average computational scientist. While CUDA C has made programming graphics cards much less esoteric, taking full advantage of optimization opportunities is not a trivial exercise and requires more than a cursory knowledge of the underlying hardware. One must consider whether the investment of several programmer-months (or more) is worth the expected performance gain if one already has a code that scales well up to several thousand CPU cores and/or takes advantage of existing APIs such as OpenMP and OpenACC to gain moderate speedups with relatively small amounts of programming effort. For example, GPU-accelerated finite

[1]Prices accurate at time of publication.

volume solvers that use MPI to exchange ghost cell data residing on separate compute nodes may not scale gracefully up to hundreds of GPUs, since the slow rate of data transfer from GPU to CPU across the PCI bus essentially eliminates the benefit of a fast interconnect like InfiniBand. This drawback may be largely overcome, however, by NVIDIA's GPUDirect technology, which will support Direct Memory Access (DMA) between GPUs in separate compute nodes.

In this article, we describe our effort to port a MUSCL-Hancock ideal magnetohydrodynamics (MHD) solver to a single NVIDIA GTX 480 graphics card. We implement a relatively simple method for GPU speedup that does not require significant knowledge of the underlying GPU hardware. Our goal is to demonstrate that despite the difficulties inherent in GPU programming, one can achieve, with minimal programming effort, a significant performance gain (more than two orders of magnitude compared to the sequential algorithm running on a single Nehalem core) for a problem of moderate size (more than a million computational cells) on a consumer-grade graphics card. We hope that our article will serve as a roadmap for those seeking to port a standard finite volume MHD code to a GPU (or GPU cluster).

2.2 GTX 480 Architecture and the CUDA Programming Model

We begin with a short description of NVIDIA's Fermi architecture and the CUDA programming model (see NVIDIA White Paper (2009)) for further details). The GTX 480 was the natural choice for our experiment since it was the first widely available (for several hundred US dollars) consumer graphics card to implement NVIDIA's Fermi architecture, which is the same architecture implemented in the much more expensive high-end Tesla series marketed to the high performance computing community. The larger number of streaming processors (SP) and active threads per SP on Fermi GPUs allows for

greater parallelism and latency hiding. Additionally, Fermi GPUs have improved memory performance, including L1 and L2 caches, reducing the need to explicitly stage data in the faster shared memory.

The GTX 480 consists of 15 streaming multiprocessors (SM), 6 GB of GDDR5 RAM (memory bandwidth of 177.4 GB/s), a 768 KB L2 cache, a host interface (through which the GPU communicates with the host CPU via PCI-express), and a scheduler that distributes blocks of threads to the SMs. Figure 2.1 shows a schematic of a single SM on the GTX 480. Each SM consists of 32 stream processors (SP) (what NVIDIA refers to as "CUDA cores"), 16 load/store units and 4 special function units. The SM schedules threads for

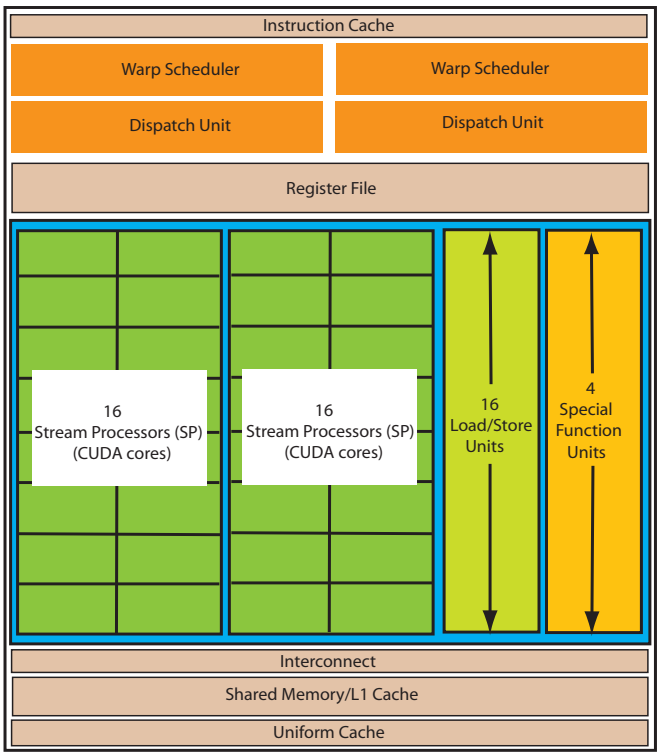


Figure 2.1 : A GTX 480 streaming multiprocessor consists of two warp schedulers, each of which can issue a single-precision instruction to 16 cores every shader clock cycle; thus, the single-precision instruction throughput is one warp (32 threads) instruction every shader cycle. The double precision instruction throughput is a factor of eight slower: 1 warp instruction every 8 shader clock cycles. Figure taken from NVIDIA White Paper (2009).

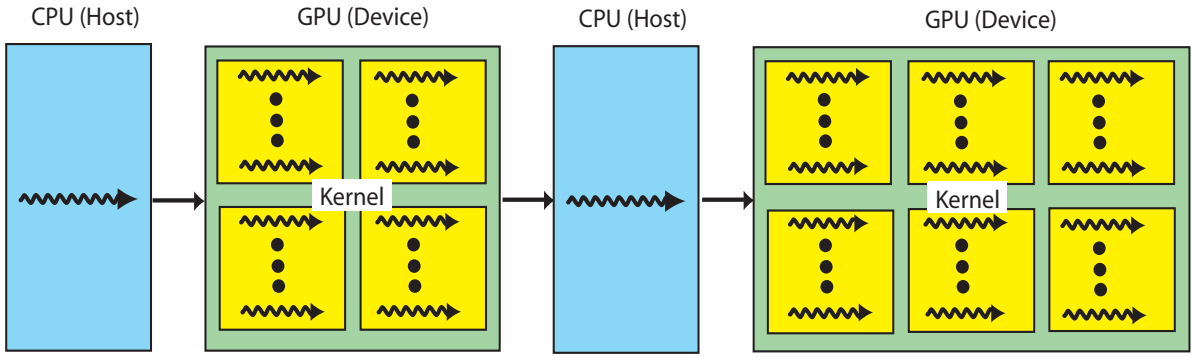


Figure 2.2 : In the CUDA programming model, the CPU ("Host") launches a sequence of "kernels" on the GPU ("Device"). Each kernel invocation launches a user-specified number of thread blocks (yellow boxes), each block containing the same number of threads. The number of thread blocks and the number of threads/block varies from kernel to kernel, but one should in general strive to maximize the number of total active threads per SM. Figure taken from NVIDIA White Paper (2009).

execution in groups of 32 called "warps", with each thread in a warp executing the same instruction. In the Fermi architecture, there are two such "warp schedulers" per SM, each capable of issuing a single-precision instruction (e.g., a multiply or add) to 16 of the 32 cores in one shader clock cycle, thus providing a single-precision throughput of 1 warp instruction per cycle. For double-precision instructions, a single warp scheduler issues the instruction to 16 of the 32 cores in one shader clock cycle. While the hardware is capable of providing a double-precision throughput of 1 warp instruction every two cycles, NVIDIA has intentionally reduced the double precision throughput of its consumer grade cards (such as the GTX480) by a further factor of 4. With 15 SMs and a shader clock speed of 1401 MHz, the overall floating point speed of the GTX 480 is 672 GFLOPS for single-precision multiplies or adds and 84 GFLOPS for double precision multiplies or adds (double these throughputs for fused multiply-adds).

We turn now to a summary of the CUDA programming model, illustrated schematically in Figure 2.2 (see the NVIDIA CUDA C Programming Guide (NVIDIA CUDA Doc-

umentation 2012) for further detail). The main serial code, running on the CPU (host), allocates GPU memory and launches a kernel that executes on the GPU (device). The programmer specifies, using a simple syntactical extension of the familiar C function call, the number of threads to execute on the GPU by specifying an MxN grid of thread blocks (yellow squares in Figure 2.2). Note that thread blocks are a programmer-defined construct to organize the calculation on the GPU. When a SM recieves a thread block, it partitions the block into warps which are then given to the scheduler for execution. Threads within a block can access a small pool (configurable to either 16 KB or 48 KB per block in Fermi) of shared memory and can be barrier-synchronized. Thread blocks are executed asynchronously on the SMs; that is, the programmer does not have the ability to synchronize the execution of the threads in different blocks (for example, by calling a "barrier" function). This asynchronous block execution allows the same CUDA C source code to work on later-generation GPUs that can execute more blocks concurrently. However, the ability to synchronize threads within a block resulted in the design constraint (to avoid long wait times) that all threads in a block are assigned to the same SM.

NVIDIA categorizes its GPUs according to "compute capability." The GTX 480 has a compute capability of 2.0, which implies the following resource constraints:

1. The maximum number of threads per block is 1024.
2. The maximum number of blocks that can be assigned to a single SM is 8.
3. The maximum number of warps that can be managed on a single SM is 48 (implying that the maximum number of threads on an SM is 1536).

When calling a kernel function, the programmer must decide carefully how many threads to assign to a block. A general rule of thumb is that if one's application is memory

bandwidth-bound, then one should strive for maximum thread occupancy (i.e., maximize the number of active warps on an SM) to take advantage of memory latency hiding. One should also set the number of threads per block to be an integer number of warps to avoid underpopulated warps (which result in idle execution resources). While NVIDIA provides an Occupancy Calculator spreadsheet to facilitate experimentation with different thread block configurations, using the Occupancy Calculator does not, as we will see below, guarantee optimal speedup.

2.3 Implementation of the MUSCL-Hancock scheme for ideal MHD

Implementing a finite volume fluid solver in CUDA is straightforward in principle, since the basic components of a standard Godunov scheme (reconstruction of edge states, evaluation of edge numerical fluxes using a Riemann solver, and conservative evolution of cell-centered quantities) map naturally to the CUDA SIMT architecture. That is, the components of the algorithm can be implemented as CUDA kernels in which each execution thread performs the same sequence of operations on a single computational cell. In the case of the MUSCL-Hancock scheme (van Leer 1984), we define a kernel for each of the following steps: 1) **slopes**: compute limited slopes at each computational cell, 2) **reconstruction**: reconstruct the conserved variables at the cell edges, 3) **evolve1**: advance reconstructed edge variables half a time step using the physical fluxes at the edges, 4) **riemann**: compute the numerical fluxes at the cell edges using a Riemann solver, and 5) **evolve2**: advance the conserved variables a full time step using the numerical fluxes computed in the previous step.

Following Dedner et al. (2002), we solve the "Generalized Lagrangian Multiplier" (GLM) formulation of the MHD equations to preserve the solenoidal constraint on the

magnetic field:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0, \tag{2.1}$$

$$\frac{\partial \rho \mathbf{v}}{\partial t} + \nabla \cdot \left[\rho \mathbf{v}\mathbf{v} + (p + \frac{B^2}{2})\mathbb{I} - \mathbf{B}\mathbf{B} \right] = 0, \tag{2.2}$$

$$\frac{\partial \mathcal{E}}{\partial t} + \nabla \cdot \left[(\frac{\rho v^2}{2} + \frac{\gamma}{\gamma - 1} p + B^2)\mathbf{v} - (\mathbf{v} \cdot \mathbf{B})\mathbf{B} \right] = 0, \tag{2.3}$$

$$\frac{\partial \mathbf{B}}{\partial t} + \nabla \cdot (\mathbf{v}\mathbf{B} - \mathbf{B}\mathbf{v}) + \nabla \psi = 0, \tag{2.4}$$

$$\frac{\partial \psi}{\partial t} + c_h^2 \nabla \cdot \mathbf{B} = -\frac{c_h^2}{c_p^2} \psi \tag{2.5}$$

where ρ is the mass density, $\mathbf{v}$ is the bulk velocity, p is the plasma pressure, $\mathbf{B}$ is the magnetic field, $\mathcal{E} = \rho v^2/2 + p/(\gamma - 1) + B^2/2$ is the total energy density, and γ is the ratio of specific heats (taken to be 5/3 in all of our simulations).

In Equation 2.4 and Equation 2.5, ψ is a scalar function whose evolution, is, by construction, equivalent to that of $\nabla \cdot \mathbf{B}$; thus, the parameters c_h and c_p represent the propagation and dissipation speeds of local magnetic field divergence errors. c_h is chosen to be the global maximum (over the grid cells) of $\max(|c_f + v_x|, |c_f + v_y|, |c_f + v_z|)$, where c_f is the fast magnetosonic wave speed. Dedner et al. (2002) found that setting $c_p^2/c_h = 0.18$ produced optimal results regardless of the grid resolution, and we use the same value in all of our simulations.

We discretize the system Equation 2.1 - Equation 2.5 using a second-order MUSCL-Hancock scheme (van Leer 1984) with a minmod slope limiter to suppress oscillations near discontinuities, Strang splitting (Strang 1968) to maintain second-order temporal accuracy and an HLL approximate Riemann solver (Toro 1999) to compute numerical fluxes. We emphasize that our use of time-splitting was not motivated by any GPU hardware or software constraints; one could easily implement an unsplit time stepping scheme using the same basic approach (indeed, we have developed an unsplit version of this MHD code and

intial tests show similar performance gains). Similarly, our choice of divergence cleaning is not driven by GPU hardware or software constraints. Other MHD algorithms which use staggered-mesh formulations to enforce the divergence constraint to machine precision (Evans & Hawley 1988; Balsara & Spicer 1999; Dai & Woodward 1998; Ryu et al. 1998; Londrillo & del Zanna 2004; Stone et al. 2008; Lee & Deane 2009) should also be easily ported to GPUs.

CUDA C source code for our programs is available online with the supplementary materials of the published paper.

Appendix A shows the implementation of the CUDA kernel `evolvex` (the z operator is implemented in a similar manner). `evolvex` takes as its first argument the pointer `p_cons` – allocated in `main.cu` using `cudaMalloc` (see Appendix A) – to the GPU global memory where the solution vector is stored. The thread block configuration is specified by declaring two variables of type `dim3`, `dimGrid` and `dimBlock`, that specify the dimensions of our two-dimensional block of threads. (Note that the absence of a third argument in these declarations defines the third dimension to be 1.) Recalling that the number of threads in a block should be an integer multiple of 32 (to avoid underpopulated warps), we specify a `TILE_WIDTH` $\times$ `TILE_WIDTH` array of threads, where `TILE_WIDTH` is typically set to either 8 or 16 (giving, respectively, 64 and 256 threads per block; see Section 2.5 for a comparison of different values of `TILE_WIDTH`). The declaration of `dimGrid` specifies the dimensions of our two-dimensional grid of thread blocks. Here, we set the number of blocks in each dimension so that the total number of threads is equal to the number of computational cells (`NX` $\times$ `NZ`); thus, each thread is responsible for updating a single computational cell at each time step. We define `dimGrid` and `dimBlock` the same way throughout the code (with the exception of the boundary condition kernels, which only involve those cells at the edges). The function calls in `evolvex` map straightforwardly to

the steps of the MUSCL-Hancock algorithm:

1. **slopes**: `compute_xslopes`
2. **reconstruction**: `d_from_slope`
3. **evolve1**: `d_from_fg_ker`
4. **riemann**: `wave_speeds_ker`, `physical_flux_f_ker`, `hll_flux_f`
5. **evolve2**: `cons_from_f_ker`, `damp_psic_ker`

Given a cell-averaged solution vector U_i^n, where i is the cell index and n is the time index, the first step in the MUSCL-Hancock scheme is to linearly extrapolate U_i^n to the left/right (or bottom/top) cell interfaces:

$$\begin{aligned} U_i^L &= U_i^n - \Delta U_i^n \\ U_i^R &= U_i^n + \Delta U_i^n \ . \end{aligned} \tag{2.6}$$

ΔU_i^n is the slope calculated by the minmod slope limiter,

$$\Delta U_i^n = \texttt{minmod}[(U_{i+1}^n - U_i^n), (U_i^n - U_{i-1}^n)] \ , \tag{2.7}$$

where

$$\texttt{minmod}(a,b) = \begin{cases} 0, & ab < 0 \\ a, & |a| < |b| \\ b, & |b| < |a| \end{cases} \ . \tag{2.8}$$

We implement **slopes** step (Equation 2.7) in the kernel `compute_xslope_ker` (the kernel `compute_zslope_ker` extrapolates to the bottom and top cell interfaces).

`compute_xslope` takes as input the GPU pointer to the solution vector (the vector of conserved cell averages, `p_cons`) and the GPU pointer to the slopes array (`p_xslope`). Within `compute_xslope_ker`, each thread defines its location in the grid (stored in the variables `a` and `b`) using both its location within the thread block and the block's location within the grid. Each thread calculates the slope at its cell by accessing the global memory pointed to by `p_cons` (where `index3` is a macro defined in `macros.h`). Slopes are then stored in the global memory pointed to by `p_xslope`. The remaining steps in the MUSCL-Hancock algorithm are implemented in a similar manner, with each thread responsible for its own computational cell and all executing threads performing the same sequence of operations at each time step.

We note here that our implementation of the minmod limiter involves a number of conditional statements (to check, for example, whether the left and right slopes differ in sign) which, while not optimal in CUDA's SIMT architecture, is nonetheless easier to code. Specifically, if some threads in a warp satisfy the `if` block and the rest satisfy the `else` block, then *all* threads in the warp must execute the `if` and `else` blocks in succession. We point out that other Riemann solvers, e.g. Roe-type (Roe 1981), HLLC (Toro et al. 1994; Batten et al. 1997) and HLLD (Miyoshi & Kusano 2005), incorporate several "if-then" conditions which will result in a performance hit. However, we do not anticipate this to be a large effect since most neighboring cells away from a discontinuity will have similar states. This means that each thread in a warp will more often than not fulfill the same if-then condition within the Riemann solver, reducing the number of instructions each warp has to execute. In general, though, one should strive to minimize the use of conditionals with divergent threads.

The **reconstruction** step (Equation 2.6) is implemented in the kernel `d_from_slope_ker`, which is invoked in both `evolvex` and `evolvez`.

`d_from_slope_ker` makes use of registers to limit the number of global memory accesses; the variables `cons_value` and `slope_value` store the value found in global memory, which is then used in the subsequent calculation. While reading this data from global memory and storing it in registers doubles the number of steps in the kernel, replacing the global memory reads with much faster register reads results in a speed-up of $\approx$ 15%.

After **reconstruction**, **evolve1** advances the extrapolated edge states by half a time step:

$$\begin{aligned} U_i^{R,n+1/2} &= U_i^R - \frac{\Delta t}{2\Delta x}[F(U_i^R) - F(U_i^L)] \\ U_i^{L,n+1/2} &= U_i^L - \frac{\Delta t}{2\Delta x}[F(U_i^R) - F(U_i^L)], \end{aligned} \tag{2.9}$$

where $F(U_i^R)$ and $F(U_i^L)$ are the ideal MHD physical fluxes evaluated at the left and right interfaces, respectively. These physical fluxes are computed in the kernel `physical_flux_f_ker` (likewise `physical_flux_g_ker`) which, like `d_from_slope`, makes use of thread registers to limit global memory accesses. **evolve1** (Equation 2.9) is performed in `d_from_fg_ker`, which is very similar in implementation to `d_from_slope_ker`.

Numerical fluxes at the cell interfaces are computed using the HLL Riemann solver:

$$F_{i+1/2}^n = \begin{cases} F(U_i^{R,n+1/2}) & \text{if } 0 \le c_i^L. \\ \frac{c_{i+1}^R F(U_i^{R,n+1/2}) - c_i^L F(U_{i+1}^{L,n+1/2}) + c_i^L c_{i+1}^R (U_{i+1}^{L,n+1/2} - U_i^{R,n+1/2})}{c_{i+1}^R - c_i^L}, & \text{if } c_i^L \le 0 \le c_{i+1}^R. \\ F(U_{i+1}^{L,n+1/2}), & \text{if } 0 \ge c_{i+1}^R. \end{cases} , \tag{2.10}$$

where $i+1/2$ denotes the interface between cells i and $i+1$, c_i^L is the minimum wave speed in cell i, and c_{i+1}^R is the maximum wave speed in cell $i+1$. The wave speeds are computed in kernels `wave_speeds_ker` and `hll_flux_f_ker`.

Finally, **evolve2**, implemented in kernel `cons_from_f_ker`, advances the cell-averaged solution vector by a full time step Δt:

$$U_i^{n+1} = U_i^n - \frac{\Delta t}{\Delta x}(F_{i+1/2}^n - F_{i-1/2}^n). \tag{2.11}$$

As a general rule, one should strive to minimize data transfers from CPU to GPU to avoid a significant PCI bandwidth bottleneck. For example, our CUDA implementation of MUSCL-Hancock involves defining a separate kernel to implement each basic component of the algorithm (**slopes**, **reconstruction**, **evolve1**, **riemann**, and **evolve2**). Thus, each kernel must receive input data from the previous kernel and send output data to the subsequent kernel. Rather than copy this data back and forth from CPU to GPU (using, for example, `cudaMemcpy`), it is much more efficient to allocate and deallocate GPU memory as needed (using `cudaMalloc`) and pass the associated pointers to the kernels. This approach requires that the entire problem live on the GPU for the duration of the calculation, thus limiting the maximum problem size. Increasing the problem size would require running on multiple GPUs (for example, by combining CUDA and MPI), taking care to minimize communication between GPUs on different compute nodes. The availablity of large GPU clusters (e.g. Keeneland, Blue Waters, Titan) thus holds the promise of similar efficiency gains with much larger problem sizes in the weak scaling limit.

A second generic optimization strategy is to favor register and shared memory over global memory to the extent possible, taking advantage of the much lower register and shared memory latency. One must, however, be aware of an important tradeoff between register and shared memory use and thread parallelism. Register and shared memory on a particular SM is divided up among all of the thread blocks on the SM. Exceeding the register or shared memory available on the SM may greatly reduce the number of active threads since threads can only be assigned to the SM at the block granularity. Experimentation

(on an application by application basis) is required to determine the proper balance between register and shared memory use, on the one hand, and thread parallelism on the other. For example, although the GTX 480 can accommodate 1024 threads per block, we found that the register footprint for our MUSCL-Hancock implementation limited us to a maximum of 256 threads per block. Interestingly, for problem sizes ranging from 64^2 to 1024^2 computational cells, 64 threads per block seemed to be optimal (for a fixed register footprint). Experimentation will be required for different architectures; the best combination of parameters for one particular type of GPU may not be ideal for other GPUs.

2.4 Results

On the CPU, the total time, T_{CPU}, required to apply a single instruction over the entire computational mesh is simply the execution latency of the instruction, $L_{E,CPU}$, multiplied by the number of mesh cells, N_{cells}: $T_{CPU} = N_{cells} L_{E,CPU}$. GPUs achieve their performance gains by executing thousands of threads in parallel, thereby hiding execution latency (which in memory bound applications is dominated by global memory latency). That is, a warp scheduler on an SM issues an instruction to an active warp (i.e., a warp that is ready to receive the instruction), which then proceeds to make use of whatever SM resources (arithmetic logic units, load/store units, special function units, etc.) it needs to execute the instruction. As the warp executes its instruction, the warp scheduler attempts to issue another instruction to another active warp; latency is completely hidden when there is always an active warp available to carry out the next instruction. Note, however, that it takes a finite time – the *issue latency*, $L_{I,GPU}$ – for the scheduler to issue one instruction to a warp. Here, $L_{I,GPU}$ is defined as the inverse of the instruction throughput (the number of instructions the SM can issue per clock cycle). For example, as we reviewed in Section 2.2, the single-precision instruction throughput for the GTX 480 is 1 instruction

per shader clock cycle so that $L_{I,GPU} = 1$ cycle (1/1401 μsec for the GTX 480). If $L_{E,GPU}$ is the execution latency of the instruction (i.e., the time it takes a warp to complete the instruction), then, assuming that there is always an active warp ready to receive the next instruction (memory latency is completely hidden), the total time, T_W, for the N_W active warps on an SM to complete the instruction is simply $T_W = (N_W - 1)L_{I,GPU} + L_{E,GPU}$. Since each active thread is responsible for a single computational cell, the N_{SM} SMs in our GPU can process $32N_W N_{SM}$ mesh cells in time T_W; thus, it takes the GPU a time $T_{GPU} = T_W[N_{cells}/(32N_W N_{SM})]$ to process all N_{cells} cells in our computational mesh with the same instruction. The expected speedup, S, for a sequence of add instructions (assuming the maximal number of active threads on the GPU) can thus be estimated as follows:

$$S = \frac{T_{CPU}}{T_{GPU}} = \frac{32N_W N_{SM} L_{E,CPU}}{L_{E,GPU} + (N_W - 1)L_{I,GPU}} \tag{2.12}$$

Problem Size	Unoptimized C	Optimized C	CUDA
64^2	13.37 s	6.45 s	0.57 s
128^2	73.39	41.80	1.81
256^2	484.33	277.73	5.24
512^2	2366.45	1476.98	18.27
1024^2	11488.6	8029.35	63.84

Table 2.1 : Average run times for various problem sizes for an unoptimized ideal MHD C code, an optimized ideal MHD C code, and a CUDA ideal MHD code. The CUDA code used here is identical to the `TILE_WIDTH` 8 code in Table 2.2.

As an example, let us estimate the expected speedup for a sequence of double precision adds (see the source code in Appendix A). Timing of the CPU code running on a single core (using gcc with the "-O3" option) resulted in an average execution latency, $L_{E,CPU}$, of approximately 10 CPU cycles for our 2.8 GHz Intel Nehalem (with array sizes of 1024x1024). Assuming $N_W = 48$ (full occupancy), $L_{I,GPU} = 8$ shader cycles (recall that

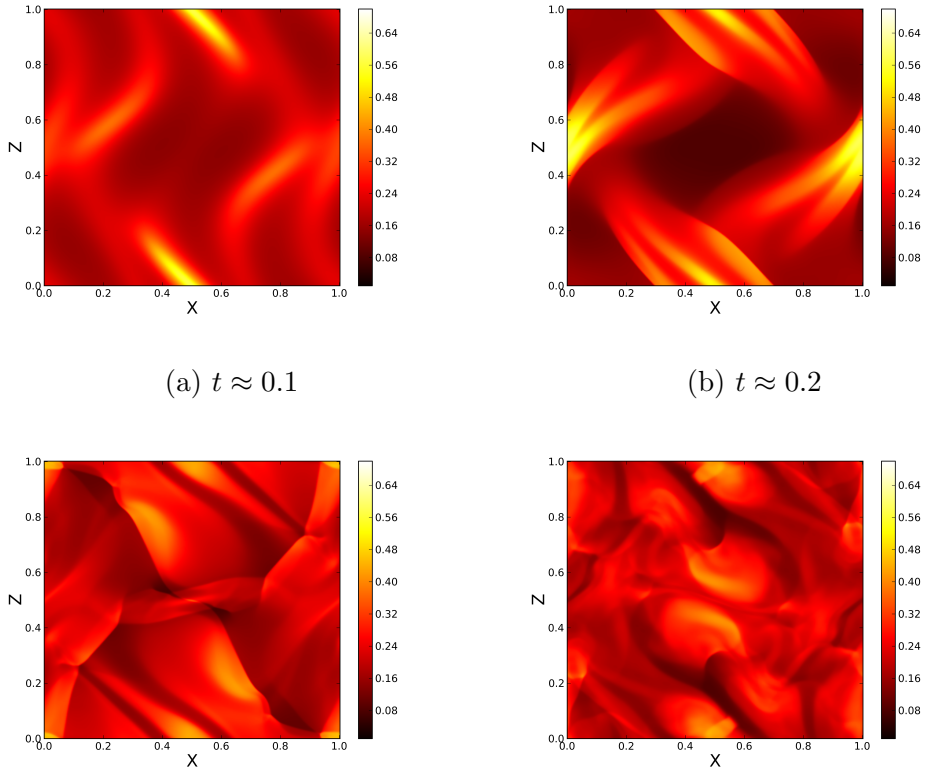
NVIDIA reduced the double precision throughput of the GTX 480 by a factor of 4 below the 1/2 double precision instruction per shader cycle of the HPC-grade Fermi cards), and $L_{E,GPU} \approx 600$ shader clock cycles (the GTX 480 global memory latency) in Equation 2.12, we obtain an overall speedup $S \approx 118$ compared to the sequential algorithm running on a single CPU core. This is close to our maximum speedup of 126 reported in Table 2.2. Timing of the corresponding CUDA kernel in Appendix A gives a speedup $S \approx 112$.

We emphasize that our simple estimate does not take into account other factors that influence GPU and CPU performance. Specifically, Equation 2.12 is independent of problem size; nevertheless, we expect the CPU memory latency to increase substantially for larger problems due to cache misses. Since memory latency is effectively hidden by the GPU, we expect more impressive GPU speedups for larger problem sizes. Additionally, we do not take into account other issues such as the relatively slow transfer of data between the CPU and GPU or the granularity of thread block sizes. Our results show that the speedups do depend on the number of threads per block, which indicates that the block size affects the number of active warps per SM. This would change the value of N_W in the above speedup equation. These are likely the biggest factors accounting for the difference between the estimated speedup (using Equation 2.12) and our measured speedups (Table 2.2). Finally, to be fair, we should compare our GPU performance on a parallel implementation of the CPU application that makes use of all four Nehalem cores; in this case, we expect an overall speedup – relative to a parallelized version of our MUSCL-Hancock code running on all four CPU cores – of roughly $118/4 \approx 30$. Additionally, many production machines have CPUs with 8 to 16 cores; this would result in a speedup closer to 10.

Ideally, one would like to achieve speedups consistent with those predicted by Equation 2.12 for a "real world" application (e.g., our MUSCL-Hancock algorithm) involving multiple kernels executing sequences of instructions consisting of many operations

(not limited to adds and multiplies). To measure the performance of our GPU-accelerated MUSCL-Hancock solver, we simulated the Orszag-Tang vortex with the following initial conditions (see, for example, Dai & Woodward (1998)): $\rho = 25/(36\pi)$, $p = 5/(12\pi)$, $v_x = -\sin(2\pi z)$, $v_y = 0.0$, $v_z = \sin(2\pi x)$, $b_x = \sin(2\pi z)/(4\pi)$, $b_y = 0.0$, and $b_z = \sin(2\pi x)/(4\pi)$. The simulation was run for 300 time steps in all test cases (Figure 2.3 shows the simulation results for a longer simulation that ran for approximately two Alfvén times). The CPU C source used in the test was compiled using gcc-4.4.3 for two cases: 1) with "-O3" option enabled (hereafter referred to as the "Optimized CPU", and 2) without the "-O3" option enabled (hereafter referred to as "Unoptimized CPU"). The simulation was run on a single Intel Core i7 930 (2.8 GHz) core. The CPU wall clock time was compared to the wall clock time of several versions of the CUDA GPU code running on the GTX 480 (see Tables 2.2 and 2.3): 1) "Register..." refers to versions with `TILE_WIDTH` $= \{2, 4, 8, 16\}$ that make use of register memory wherever possible to store intermediate values (the solution vector, however, remains in GPU global memory for the entire calculation); 2) "Global..." refers to the CUDA version that makes extensive use of GPU global memory with minimal usage of registers and `TILE_WIDTH` $= \{2, 4, 8, 16\}$. The timing results included disk I/O time; that is, we started our clock at the beginning of the main program and ended the clock after the final output had been written to disk.

On average, the CUDA code is roughly 59 times faster than Optimized C; however, it is clear from Table 2.1 that the speedup increases with problem size. For example, for a 64^2 double precision problem, the CUDA code executes only 11 times faster than Optimized C, while the 1024^2 double precision CUDA code is 126 times faster than Optimized C. The increase in speedup with problem size likely reflects the fact that the CPU code is memory bound for large problem sizes, so that memory bandwidth and latency determine the CPU execution latency, $L_{E,CPU}$ in Equation 2.12; in contrast, the global memory latency is



(a) $t \approx 0.1$ (b) $t \approx 0.2$

(c) $t \approx 0.4$ (d) $t \approx 0.8$

Figure 2.3 : Evolution of the Orszag-Tang vortex for a grid size of 512^2.

effectively hidden for large problems running on the GPU. Note that our CPU code does not implement any special optimizations to overcome the memory bottleneck, e.g. strip mining (Carr & Kennedy 1992), space tiling (Wolfe 1989), or linear loop transformations (Li & Pingali 1993).

It is clear from Tables 2.2 and 2.3 that `TILE_WIDTH` $= 8$ seems to be the "sweet spot" where the number of threads per block is large enough to effectively hide global memory latency but not so large that heavy use of thread registers reduces SM occupancy (due to block granularity of thread assignments). From our testing, the register-heavy code is always faster (by $\approx 15 - 25\%$) than the global memory-heavy codes, keeping `TILE_WIDTH` constant. Interestingly, the Global 8 code turns out to be faster than the Register 16 code.

Problem Size	Register 16	Register 8	Register 4	Register 2
64^2	0.8 s (8.1)	0.57 s (11)	0.67 s (9.7)	1.2 s (5.4)
128^2	2.25 (19)	1.81 (23)	2.04 (21)	4.34 (9.6)
256^2	6.52 (43)	5.24 (53)	6.71 (41)	16.13 (17)
512^2	22.58 (65)	18.72 (81)	25.19 (59)	68.12 (22)
1024^2	84.62 (95)	63.84 (126)	90.61 (89)	253.88 (32)

Table 2.2 : Average run times for permutations of our single GPU ideal MHD code. Register *number* refers to the codes using a register-heavy approach with the `TILE_WIDTH` set to *number*. Numbers in parentheses are the approximate speedups relative to the Optimized C timings in Table 2.1.

Problem Size	Global 16	Global 8	Global 4	Global 2
64^2	0.95 s (6.8)	0.63 s (10)	0.82 s (7.9)	1.59 s (4)
128^2	2.74 (15)	2.09 (20)	2.52 (17)	5.93 (7)
256^2	8.51 (33)	6.16 (45)	8.76 (32)	22.825 (12)
512^2	29.92 (49)	22.06 (67)	33.36 (44)	97.46 (15)
1024^2	113.88 (71)	77.34 (104)	120.31 (67)	357.69 (22)

Table 2.3 : Average run times for permutations of our single GPU ideal MHD code. Global *number* refers to the codes using a global memory-heavy approach with the `TILE_WIDTH` set to *number*. Numbers in parentheses are the approximate speedups relative to the Optimized C timings in Table 2.1.

Our finding that using 64 threads per block results in the fastest code is puzzling, given that the NVIDIA CUDA Occupancy Calculator suggests that our application can accommodate 256 threads/block (which should, in principle, result in a greater degree of parallelism and memory latency hiding). Ultimately, these results emphasize the importance of experimentation to find the right balance of register use and number of threads per block.

2.5 Conclusions and Discussion

In this paper, we have described in detail our experience porting a two-dimensional MUSCL-Hancock ideal MHD solver to a single NVIDIA GTX 480 (Fermi architecture)

using CUDA C. We compared a sequential CPU version of the algorithm – compiled with gcc-4.4.3 and executed on a single Intel Core i7 930 (2.8 GHz) CPU core – to a parallel version running on a single GTX 480. We made no special effort to stage data in the much faster shared memory of the GPU; rather, we experimented with different combinations of thread register use and thread block size to maximize the number of active threads per streaming multiprocessor (SM) (and thereby hide memory latency for our memory bound application). For a double precision problem with 1024^2 mesh cells, we achieved a maximum speedup of roughly 126 (with the CPU source code compiled using the "-O3" gcc optimization flag) by making use of thread registers to store kernel variables whenever possible and setting the number of threads per block to 64. Interestingly, keeping all of the data in global memory (with 64 threads per block) was more efficient than using large amounts of register memory in combination with a larger number (256) of threads per block.

Several other groups (see, for example, Schive et al. (2010), Wong et al. (2011) and references therein) have reported on GPU implementations of MHD codes. The results of Wong et al. (2011) (hereafter W11) are particularly relevant to our work, since their algorithm is an explicit finite volume method on a uniform mesh (in contrast to the AMR simulation tested by Schive et al. (2010)), and their timing tests included a two-dimensional case on a GTX 480. For problem sizes of 512^2 and 1024^2, W11 quote speedups (relative to a single 3.2 GHz Intel Nehalem core) of 320 and 600, respectively. For a 128^3 double precision problem (twice the size of their 1024^2 problem with speedup of 600), W11 report a speedup of only 155. They report a speedup of 260 for a 64^3 problem (still less than the reported 320 speedup for a 2D problem with the same number of computational cells). These speedups are higher than ours for similarly sized problems, and none are consistent with the simple estimate based on Equation 2.12.

We speculate that the discrepancy between our speedups and those reported by W11 is a result of inefficient memory access in the 1D and 2D versions of the Fortran code they used for their CPU runs. Specifically, it is clear from Table 4 of W11 that the $\mathbf{fluid}_y$ and $\mathbf{fluid}_z$ operations are much more expensive than $\mathbf{fluid}_x$ (due to less efficient memory access for the y and z fluid updates), whereas these operations take similar time on the GTX 480 (where memory latency is effectively hidden). Further, W11's Fortran CPU code calls the **Transposition** operation, whereas their CUDA code does not. Similarly, Table 5 of W11 shows that $\mathbf{fluid}_z$ is much more expensive than $\mathbf{fluid}_x$ or $\mathbf{fluid}_y$ (which are comparable since the **Transposition** operation allows efficient memory access in the x and y dimensions) in the Fortran CPU code, while all three operations take similar times in the CUDA code. In contrast, $\mathbf{fluid}_x$, $\mathbf{fluid}_y$ and $\mathbf{fluid}_z$ all take similar times in the 3D cases (for which the **Transposition** operation allows data in all three dimensions to be accessed efficiently).

Thus, it appears that W11's CPU implementation uses a fluid update algorithm which is significantly less efficient than the GPU algorithm for 1D and 2D problems. In the 64^3 case reported in their Table 6, if we ignore the **Transposition** operation (not included in W11's GPU implementation), their GTX 480 single precision speedup is 97 for a problem size of 64^3. Using W11's reported (in their Table 3) double vs. single precision run time of 1.3973 for their 64^3 problem, their double precision speedup is roughly 69, which is much closer to the double precision speedup (over the optimized C code, compiled with the "-O3" gcc option) of 67 we observe for our "Global 8" 512^2 problem (with the same number of computational cells) in Table 2.3. Thus, we are confident that the speedups we report in this paper are consistent with those reported by W11.

Though these results are encouraging for applications that fit on a single GPU, real world applications are much larger than 1024^2 and generally must be run on thousands of compute nodes. The newest supercomputers (e.g. Keeneland, Blue Waters, Titan) are

being created with a hybrid architecture in which each node consists of a multi-core CPU and one or more GPUs. The method presented here is chiefly based on minimizing the amount of data being transfered between the CPU and GPU, so the natural extension to many nodes is to minimize both the intra- and inter-node communication. In our approach, each thread on the GPU is responsible for one computational cell; this allows us to divide the overall simulation space among multiple GPUs since individual cell calculations are independent from other cell calculations. In this case, the only inter-GPU communication required is the boundary cell states prior to the time evolution and the timestep. Currently, passing data between GPUs in different compute nodes represents a significant communication bottleneck. We note that with CUDA 5, however, NVIDIA has implemented "GPUDirect", which enables direct communication between GPUs in a cluster rather than routing the data through their associated CPUs with MPI.

Though we have presented results indicating that a single-GPU code shows significant speedup over a single-CPU code, it is not yet clear whether a multi-GPU code will offer the same speedup relative to a multi-CPU code. It is possible, however, that an explicit finite volume MHD solver might achieve good weak scaling up to thousands of GPUs using a message passing approach in which only ghost cell data is communicated between GPUs (with the rest of the solution remaining on the GPUs for the entire calculation).

In summary, we have demonstrated that porting a non-trivial finite-volume algorithm (a second-order MUSCL-Hancock ideal MHD solver) to a modern (compute capability 2.0 or higher) GPU can yield a significant speedup compared to a single Nehalem core – comparable to that predicted by a simple estimate based on the arithmetic throughput, memory latency and parallel processing capacity of the GPU – with minimal programming effort. While some experimentation with different combinations of thread register use and thread block size is useful, one can in general expect speedups of ≈ 100 relative to a single

core for moderately large double precision problems ($> 1024^2$ computational cells). Future work will focus on comparing a multi-GPU MHD code with a similar multi-CPU MHD code in order to delineate the benefits and drawbacks of GPUs vs. CPUs in the high-performance supercomputing realm. For now, we are confident in recommending GPUs as a viable speedup tool for small- to moderate-sized simulations ($< 10^6$ cells).

2.6 Acknowledgment

The authors would like to thank X. Feng and H.C. Wong for helpful discussion regarding their timing results. CB was supported by the Delaware Space Grant program (NASA Grant NNG05G092H) and the NASA GSRP Fellowship (NASA Grant NNX11AK70H).

References

Balsara, D. S., & Spicer, D. S. 1999, Journal of Computational Physics, 149, 270

Batten, P., Clarke, N., Lambert, C., & Causon, D. M. 1997, SIAM J. Sci. Comput., 18, 1553

Belleman, R. G., Bédorf, J., & Zwart, S. F. P. 2008, New Atronomy, 13

Brandvik, T., & Pullan, G. 2008, 46th AIAA Aerospace Sciences Meeting and Exhibit

Carr, S., & Kennedy, K. 1992, in Proceedings of the 1992 ACM/IEEE Conference on Supercomputing, Supercomputing '92 (IEEE Computer Society Press), 114–124

Dai, W., & Woodward, P. R. 1998, Astrophys. J., 494, 317

Dedner, A., Kemm, F., Kröner, D., et al. 2002, J. Computational Phys., 175, 645

Evans, C. R., & Hawley, J. F. 1988, Astrophys. J., 332, 659

Friedrichs, M. S., Eastman, P., Vaidyanathan, V., et al. 2009, J. Comp. Chem., 30, 864

Gaburov, E., Bédorf, J., & Zwart, S. P. 2012, Procedia Computer Science, 1, 1119

Harris, C., Haines, K., & Stavely-Smith, L. 2008, Exp. Astron., 22, 129

Kirk, D. B., & Hwu, W.-m. 2010, Programming Massively Parallel Processors (Morgan Kaufmann)

Lee, D., & Deane, A. E. 2009, Journal of Computational Physics, 228, 952

Li, W., & Pingali, K. 1993, ACM Trans. Comput. Syst., 11, 353

Londrillo, P., & del Zanna, L. 2004, Journal of Computational Physics, 195, 17

Miyoshi, T., & Kusano, K. 2005, Journal of Computational Physics, 208, 315

NVIDIA CUDA Documentation. 2012, CUDA C Programming Guide Version 4.2, NVIDIA Corporation

NVIDIA White Paper. 2009, NVIDIA's Next Generation CUDA Compute Architecture: Fermi, Tech. rep., NVIDIA Corporation

Roe, P. L. 1981, Journal of Computational Physics, 43, 357

Ryu, D., Miniati, F., Jones, T. W., & Frank, A. 1998, Astrophys. J., 509, 244

Schive, H.-Y., Tsai, Y.-C., & Chiueh, T. 2010, Astrophys. J. Suppl., 186, 457

Stone, J. E., Phillips, J. C., Freddolino, P. L., et al. 2007, J. Comput. Chem., 28, 2618

Stone, J. M., Gardiner, T. A., Teuben, P., Hawley, J. F., & Simon, J. B. 2008, Astrophys. J., Suppl. Ser., 178, 137

Strang, G. 1968, SIAM J. Numerical Anal., 5, 506

Toro, E. F. 1999, Riemann solvers and numerical methods for fluid dynamics: A practical introduction (Springer-Verlag)

Toro, E. F., Spruce, M., & Speares, W. 1994, Shock Waves, 4, 25

van Leer, B. 1984, SIAM J. Sci. Statist. Comput., 5, 1

Wolfe, M. 1989, in Proceedings of the Third SIAM Conference on Parallel Processing for Scientific Computing (Society for Industrial and Applied Mathematics), 357–361

Wong, H.-C., Wong, U.-H., Feng, X., & Tang, Z. 2011, Comp. Phys. Comm., 182, 2132

A Appendix

All source code here has been removed to save space. It can be found with the published paper.

Chapter 3

A Hall Magnetohydrodynamics Code with CUDA and MPI

Based on work done with John Dorelli (NASA-GSFC) and Peng Wang (NVIDIA)

"Either this is madness or it is Hell."

"It is neither," calmly replied the voice of the Sphere, "it is Knowledge; it is Three Dimensions"

Edwin Abbott

Abstract

We describe our experience in updating our two-dimensional ideal-MHD code, written in NVIDIA's CUDA C language to run on a single graphics processing unit (GPU). Firstly, we update the ideal MHD algorithm to three dimensions and implement the Hall term in order to enable future realistic magnetosphere simulations with fast magnetic reconnection. We present several benchmarks for our new algorithm: the Orszag-Tang vortex, a whistler wave, and a three-dimensional blast wave. Secondly, we incorporate the Message Passing Interface standard to allow our code to run on multiple GPUs. We discuss several strategies we used to optimize the code's run time and present scaling results from the *Keeneland* and *KIDS* GPU supercomputers which demonstrate excellent strong and weak scaling with intermediate numbers of GPUs.

3.1 Introduction

Ideal magnetohydrodynamics (MHD) has proven to be a good approximation for simulating most space plasmas. However, in some situations (usually involving magnetic reconnection or turbulence), the ideal approximation no longer applies and a different scheme must be used. One option is to solve the Vlasov-Maxwell system of equations in a full-particle code (Dawson 1983; Bowers et al. 2008, 2009). Alternatively, one can employ a hybrid approach, treating the ions kinetically and approximating the electrons as a fluid (Byers et al. 1978; Hewett & Nielson 1978; Winske et al. 2003; Kunz et al. 2014). The MHD equations can also be extended with nonlinear terms (e.g. NIMROD, Sovinec et al. 2004). However, these approaches are still much more computationally expensive compared with the MHD fluid approach, especially in three-dimensional cases. An alternative option is to keep the fluid equations, but add the Hall term to enable the decoupling of the ion and electron fluids. Hall MHD appears to be the simplest, most computationally inexpensive scheme that allows for simulation of fast reconnection events (Birn et al. 2001).

Despite its relative simplicity compared to full-particle codes, Hall MHD is rather slow compared to ideal MHD; this difference is due to the introduction of the whistler wave. The whistler wavespeed is inversely proportional to its wavelength. As the grid resolution decreases, the maximum wavespeed of a resolved whistler wave increases. Since whistler waves are usually the fastest-propagating waves in a Hall MHD simulation, doubling resolution more than halves the explicit timestep as calculated by the Courant-Friedrichs-Lewy condition ($\Delta t < \Delta x/v$; Courant et al. 1928). As a result of these tiny timesteps, very large and high resolution Hall MHD problems, e.g. global magnetosphere simulations, take quite a long time to complete. One approach to remedy this issue is to use implicit

time stepping (Tóth et al. 2008), which allows for timesteps longer than CFL. Although the scheme is more computationally expensive than the explicit scheme, the longer implicit timestep allows for a shorter program runtime.

The strategy we implement in this paper is to use more powerful computational tools on the small explicit Hall MHD timesteps. Over the last few years, GPUs have risen as a viable source of computing acceleration in such areas as N-body simulations (Belleman et al. 2008; Nitadori & Aarseth 2012), molecular dynamics (Kylasa et al. 2014), and fluid dynamics (Schive et al. 2010). Additionally, several GPU MHD codes have been previously developed (Bard & Dorelli 2014; Wong et al. 2011; Schive et al. 2010), which reported speedups from 10x-100x over similar programs running on a CPU core. The most recent multi-GPU MHD code in the literature (Wong et al. 2014) divides the simulation grid across multiple GPUs, using GPUDirect to pass information between GPUs sharing a PCIe bus and MPI to pass information between GPU on different nodes.

Here, we report on our development of a multi-GPU Hall MHD code which uses MPI to communicate between GPUs. Unlike Wong et al. (2014), we use a different algorithim and solve the Hall MHD equations rather than ideal MHD. Additionally, we do not yet implement GPUDirect, as the latest version allowing for direct GPU-GPU memory transfer between nodes over Infiniband was not available to us. We document our experience developing this code and discuss several useful optimizations implemented in the process. Benchmarking results are presented, and we show that our code algorithm is valid and second-order accurate. Finally, we give strong and weak scaling results from the Keeneland Initial Delivery System (KIDS) cluster at Georgia Tech demonstrating that our code scales well to multiple GPUs, though better inter-node communication via GPUDirect will be needed.

3.2 Implementation of the MUSCL-Hancock scheme for Hall MHD

We solve the "Generalized Lagrangian Multiplier" (GLM) formulation of the MHD equations (Dedner et al. 2002) to preserve the solenoidal constraint on the magnetic field. For Hall MHD, we add the Hall term to the GLM-MHD equations:

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

$$\frac{\partial \rho \mathbf{v}}{\partial t} + \nabla \cdot \left[\rho \mathbf{v}\mathbf{v} + (p + \frac{B^2}{2})\mathbb{I} - \mathbf{B}\mathbf{B} \right] = 0, \tag{3.2}$$

$$\frac{\partial \mathcal{E}}{\partial t} + \nabla \cdot \left[(\frac{\rho v^2}{2} + \frac{\gamma}{\gamma - 1} p)\mathbf{v} + B^2(\mathbf{v} + \mathbf{v}_H) - [(\mathbf{v} + \mathbf{v}_H) \cdot \mathbf{B}]\, \mathbf{B} \right] = 0, \tag{3.3}$$

$$\frac{\partial \mathbf{B}}{\partial t} + \nabla \cdot [(\mathbf{v} + \mathbf{v}_H)\mathbf{B} - \mathbf{B}(\mathbf{v} + \mathbf{v}_H)] + \nabla \psi = 0, \tag{3.4}$$

$$\frac{\partial \psi}{\partial t} + c_h^2 \nabla \cdot \mathbf{B} = -\frac{c_h^2}{c_p^2} \psi \tag{3.5}$$

where ρ is the mass density, $\mathbf{v}$ is the bulk velocity, p is the plasma pressure, $\mathbf{B}$ is the magnetic field, $\mathcal{E} = \rho v^2/2 + p/(\gamma - 1) + B^2/2$ is the total energy density, and γ is the ratio of specific heats (taken to be 5/3 in all of our simulations). The Hall velocity is defined as $\mathbf{v_H} = -\delta_\mathbf{i} \mathbf{J}/\rho = -\delta_\mathbf{i} \nabla \times \mathbf{B}/\rho$, where δ_i is the ion inertial length and $\mathbf{J}$ is the current density. δ_i is a normalized parameter in our code; we typically set it so that one ion inertial length is covered by ten computational cells.

Following Tóth et al. (2008), we calculate the current density to second-order accuracy with central differencing and averaging. The equations for the x face are presented here:

$$\mathbf{J}^x_{i+1/2,j,k} = \frac{\mathbf{B}^z_{i,j+1,k} + \mathbf{B}^z_{i+1,j+1,k} - \mathbf{B}^z_{i,j-1,k} - \mathbf{B}^z_{i+1,j-1,k}}{4\delta y} - \frac{\mathbf{B}^y_{i,j,k+1} + \mathbf{B}^y_{i+1,j,k+1} - \mathbf{B}^y_{i,j,k-1} - \mathbf{B}^y_{i+1,j,k-1}}{4\delta z}, \tag{3.6}$$

$$\mathbf{J}^y_{i+1/2,j,k} = \frac{\mathbf{B}^x_{i,j,k+1} + \mathbf{B}^x_{i+1,j,k+1} - \mathbf{B}^x_{i,j,k-1} - \mathbf{B}^x_{i+1,j,k-1}}{4\delta z} - \frac{\mathbf{B}^z_{i+1,j,k} - \mathbf{B}^z_{i,j,k}}{\delta x}, \tag{3.7}$$

$$\mathbf{J}^z_{i+1/2,j,k} = \frac{\mathbf{B}^y_{i+1,j,k} - \mathbf{B}^y_{i,j,k}}{\delta x} - \frac{\mathbf{B}^x_{i,j+1,k} + \mathbf{B}^x_{i+1,j+1,k} - \mathbf{B}^x_{i,j-1,k} - \mathbf{B}^x_{i-1,j-1,k}}{4\delta y}, \tag{3.8}$$

with similar equations for the other faces. The normal derivatives only use the two closest cell centers, but the tangential derivatives use four cell centers and averaging.

The inclusion of the Hall term introduces whistler waves to the simulation, so we estimate the maximum wave speed for the CFL condition through

$$v_{\max} = \max[v_{w,x}, v_{w,y}, v_{w,z}], \tag{3.9}$$

with

$$v_{w,x} = |v_x| + v_{f,x} + \delta_i \frac{\pi |B|}{\rho \Delta x}, \tag{3.10}$$

$v_{f,x}$ being the fast magnetosonic speed in the x-direction. Similar equations apply for $v_{w,y}$ and $v_{w,z}$.

In Equation 3.4 and Equation 3.5, ψ is a scalar function whose evolution, is, by construction, equivalent to that of $\nabla \cdot \mathbf{B}$. The parameters c_h and c_p represent the propagation and dissipation speeds of local magnetic field divergence errors. c_h is chosen to be the global maximum (over the grid cells) of $\max(|c_f + v_{w,x}|, |c_f + v_{w,y}|, |c_f + v_{w,z}|)$, where c_f is the fast magnetosonic wave speed. Dedner et al. (2002) found that setting $c_p^2/c_h = 0.18$ produced optimal results regardless of the grid resolution, so we arbitrarily choose c_p to maintain that ratio in all of our simulations.

Our Hall MHD code implements a three-dimensional second-order MUSCL-Hancock

scheme (van Leer 1984) with explicit time integration:

$$U^{n+1}_{i,j,k} = U^n_{i,j,k} - \frac{\Delta t}{\Delta x}(F^n_{i+1/2,j,k} - F^n_{i-1/2,j,k}) \\ - \frac{\Delta t}{\Delta y}(G^n_{i,j+1/2,k} - G^n_{i,j-1/2,k}) - \frac{\Delta t}{\Delta z}(H^n_{i,j,k+1/2} - H^n_{i,j,k-1/2}), \tag{3.11}$$

where $U^n_{i,j,k}$ is the solution state $[\rho, \mathbf{v}, \mathcal{E}, \mathbf{B}, \psi]$ at time n in cell i, j, k. The fluxes F, G, H are calculated with a HLL approximate Riemann solver (Toro 1999) from half-time-step $(n + 1/2)$ interface states. The calculation for F at cell interface $i + 1/2$ (with j and k constant) is

$$F^n_{i+1/2} = \begin{cases} F(U_i^{R,n+1/2}) & 0 \le c_i^L \\ \frac{c^R_{i+1}F(U_i^{R,n+1/2}) - c_i^L F(U_{i+1}^{L,n+1/2}) + c_i^L c^R_{i+1}(U_{i+1}^{L,n+1/2} - U_i^{R,n+1/2})}{c^R_{i+1} - c_i^L} & c_i^L \le 0 \le c^R_{i+1} \\ F(U_{i+1}^{L,n+1/2}) & 0 \ge c^R_{i+1} \end{cases}, \tag{3.12}$$

where c_i^L is the minimum wave speed in cell i and c^R_{i+1} is the maximum wave speed in cell $i + 1$, both traveling in the x direction. Similar calculations apply for $G^n_{i\pm1/2}$ and $H^n_{i\pm1/2}$.

In calculating the fluxes, we use the half-time-step evolved edge states $U^{\text{edge},n+1/2}_{i,j,k}$. Each cell has six edge states: left (L) and right (R) in the x-direction, down (D) and up (U) in the y-direction, and back (B) and front (F) in the z-direction. The edge states are linearly extrapolated from the cell center state:

$$U^{L,n}_{i,j,k} = U^n_i - \Delta U^{x,n}_{i,j,k} \\ U^{R,n}_{i,j,k} = U^n_i + \Delta U^{x,n}_{i,j,k} \tag{3.13}$$

(with similar equations for U^D, U^U with $\Delta U^{y,n}_{i,j,k}$, etc.). Since the whistler wave speed in Hall MHD is proportional to the grid resolution, we need to calculate the slopes $\Delta U_{i,j,k}$ with a monotonized central (MC) limiter in order to get second-order numerical dissipation

and a second-order accurate MUSCL-Hancock scheme (Tóth et al. 2008):

$$\Delta U^{x,n}_{i,j,k} = \text{minmod}\left[\beta(U_{i+1} - U_i), \beta(U_i - U_{i-1}), \frac{U_{i+1} - U_{i-1}}{2}\right], \tag{3.14}$$

where the minmod function returns zero if the arguments have different signs, otherwise returning the argument with the smallest magnitude. Similar equations apply for $\Delta U^{y,n}_{i,j,k}$ (using U_j, U_{j+1}, U_{j-1}) and $\Delta U^{z,n}_{i,j,k}$. We set $\beta = 2$. Finally, we calculate the evolved edge states (at $n = 1/2$) by

$$\begin{aligned} U_i^{R,n+1/2} &= U_i^R - \frac{\Delta t}{2\Delta x}[F(U_i^R) - F(U_i^L)] \\ U_i^{L,n+1/2} &= U_i^L - \frac{\Delta t}{2\Delta x}[F(U_i^R) - F(U_i^L)], \end{aligned} \tag{3.15}$$

with similar equations for the other four evolved edge states.

We emphasize that our choice of scheme and divergence cleaning was not motivated by GPU hardware or software constraints. Other MHD algorithms which use staggered-mesh formulations to enforce the divergence constraint to machine precision (e.g. (Evans & Hawley 1988; Balsara & Spicer 1999; Dai & Woodward 1998; Stone et al. 2008; Lee & Deane 2009)) should also be easily ported to GPUs. Additionally, other Riemann solvers, e.g. Roe-type (Roe 1981), HLLC (Toro et al. 1994; Batten et al. 1997) and HLLD (Miyoshi & Kusano 2005), should work well with GPUs.

3.3 The MPI+CUDA Programming Methodology

Here, we review some relevant aspects of the CUDA/MPI paradigm. Further detail can be found in NVIDIA White Paper (2009).

The main code, written in C, runs on the CPU host and uses CUDA library functions to allocate/initialize GPU memory and launch kernels to manipulate the data on the GPU. The lowest level of execution within is called a "thread". Threads execute the individual

instructions within the kernels; however, they do not act on their own. Instead, threads are organized in groups of 32 called "warps". When an instruction is issued, the warp receives it and the threads execute it concurrently. However, if conditional statements are present (e.g. "if-then" blocks) and the threads in a warp are split between the statements, then the warp will have to execute each conditional statement sequentially. The next levels of granularity are the "thread block" and "thread grid", programmer-defined constructs which help conceptually organize the grid calculations on the GPU. When a kernel is called, the programmer specifies the number of threads to be executed by specifying a 1D, 2D, or a 3D grid of thread blocks with some number of threads per block. Typically, the number of threads per block is divisible by 32, the number of threads in a warp, so that there are no ununsed threads when the warp executes. We organize our simulations such that one thread is responsible for evolving the solution state in one computational cell.

On a higher level, the GPU is comprised of subunits called streaming multiprocessors (SM), which consist of CUDA cores (which execute arithmetical instructions), load/store units (which perform memory operations), and warp schedulers (which issue instructions to warps). When a kernel executes, the SM receives thread blocks, which it then partitions into warps. Warps are then given to the scheduler for instructions and execution.

In a typical implementation of a MHD program with CUDA, each thread is responsible for updating a single computational cell within the grid domain. Each cell update involves executing many arithmetic operations and memory accesses, but the GPU is able to hide latency by issuing instructions to warps that are waiting to execute. If the GPU is underfilled, however, the smaller number of active warps makes it harder for the SM to find a ready warp (i.e. not currently executing an instruction). The scheduler would then have to wait for a warp to finish before issuing another instruction. For large problems, GPU latencies are better hidden since there are more active warps and the scheduler is

more likely to find warps ready to receive instructions.

With MPI, we can coordinate multiple pairs of CPUs and GPUs each running the main program on their own segment of the simulation. We maintain consistency by broadcasting global data, like the timestep, and copying ghost cell boundary data to the relevant nodes. To do this, however, we need to calculate/copy data on the GPU, transfer it to its partner CPU, and then pass it over the network to the appropriate CPU/GPU pair. This process is relatively slow compared to arithematic calculation, leading to a bottleneck. The development of GPUDirect promises to solve this issue by allowing for the direct copying of data between global memory locations on multiple GPUs.

3.4 GPU Optimizations

The main sources of bottlenecks in multi-GPU codes include data transfers between the CPU and GPU, inter-node communication, memory latency, and memory bandwidth. The GPU's capability for latency hiding means that optimizing arithimetic instruction execution won't have a significant effect in most cases. Provided that the GPU is not underfilled, the warp will typically finish its math operation before the scheduler finishes issuing instructions to other warps. However, memory operations take much longer to execute.

We find that our greatest gains in optimization involve maximizing memory coalescing, that is, minimizing the number of memory reads actually performed by the warps. When the threads in a warp access memory, the GPU loads information from global memory into a cache. Each thread checks the cache for its desired index. If there is a miss (i.e. the thread's memory request is not in the cache), the GPU has to make another memory access and load the cache from a different part of global memory.

Since the first index in CUDA, x, is the fastest-varying, adjacent memory locations

typically have a common y and z index. In order to minimize memory calls, threads in a warp should share a common y and z-index when possible. In our first version of the code, we used 3D thread blocks with a 3D grid. Consequently, thread blocks were comprised of different rows, columns, and ledges of the grid, i.e. there were several different x, y, and z memory indices present within one warp. This made it likely that contiguous threads in a warp did not access contiguous locations in memory, leading to more cache misses and more global memory loads per warp.

The solution to this is to map the threads representing 3D grid cells to 1D thread blocks using a macro, making sure that x is the fastest varying index. As a result, threads within a warp are more likely to access contiguous locations in memory, and the kernel executes faster.

Currently, in order to pass data (e.g. ghost cell boundaries) between nodes, we need to copy the arrays from the GPU to its associated CPU, use MPI to pass the array to the appropriate CPU, then do another CPU-GPU data transfer. We minimize data transfers between the CPU and GPU by keeping the data on the GPU as much as possible, copying arrays back to the CPU only when absolutely necessary. Almost every function is executed on the GPU; the CPU is only used for kernel invocation, message passing, and data output.

Another strategy is to use pinned memory. Pinned memory (also known as page-locked host memory) allows for concurrent data transfer/kernel execution and a higher memory bandwidth when copying data between the CPU and GPU. However, there is a limited amount available, so we have to be careful not to use too much. Fortunately, the ghost cell data is perfect for pinning, since it uses a relatively small amount of data and it must be transferred every timestep. We get a modest speedup of 2-4% with pinning the ghost cell arrays in memory. This may not be needed in the future with the advent of GPUDirect, which will allow us to directly pass data from one GPU to the next without

having to explicitly copy the data to the associated CPUs.

A final strategy is to minimize the number of kernels called by the program. This reduces the overhead from setting up the thread grid and preparing the SM for issuing/executing instructions. Previously, we had separate kernels for performing operations on the ghost cells and the real cells. We did this since it was easier to ensure that each thread in a warp was doing some operation, i.e. there were no underutilized warps. For example, if we have a 128^3 3D grid with two ghost cells in each direction (overall data array size 132^3), we handle the real cells (128^3 grid) in one kernel and each ghost cell face (2x128x128, 132x2x128, 132x132x2) in other kernels. By choosing proper values for the grid dimensions (i.e. powers of 2), we ensure that the total number of real cells is always divisible by 32 threads/warp, meaning that there are no unused cells per warp. However, the increased overhead from creating/destroying kernels outweighs the inefficiency of underutilized warps. It is faster to use a single kernel to handle the entire array and have some unused, leftover threads in a warp than to have one kernel for the real cells and one for each ghost face, all with perfect thread usage.

The final result from this round of optimization is that our kernels are able to acheive a memory bandwidth peak of 70-80% maximum. For a "realistic use case" with the GPU filled to capacity, we find that memory bandwidth is the bottleneck for our code. This is good for weak scaling, since most of the program wallclock time will be spent performing calculations rather than on memory transfers and MPI communication. The most important factor in our optimization is increasing the memory coalescing, since reading from GPU memory is quite slow. Minimizing the number of global memory accesses both directly (with shared memory and registers in the kernels) and indirectly (with memory coalescing) should be the first priority in any GPU optimization. Finally, we emphasize that profiling is very important for determining which kernels to focus on in optimization. A 1% speedup in a

kernel called many times can result in a faster program than a 50% speedup in a kernel scarcely called.

3.5 Benchmarks

To check the accuracy and correctness of our algorithm, we benchmark our code with several simple tests, namely a 1D and 2D whistler wave (Tóth et al. 2008), a Orszag-Tang vortex (Dai & Woodward 1998), and a 3D blast wave (e.g. Balsara & Spicer (1999)).

N	1D Error	Error Ratio
16	0.267	-
32	0.0665	4.01
64	0.0156	4.27
128	0.00372	4.19
256	0.000906	4.11
512	0.000224	4.04

Table 3.1 : Whistler wave errors as calculated in Equation 3.17 for 1D whistler wave simulations. N is the number of cells used in the simulation and Error Ratio is the result of dividing the error in the above row by the error in the current row. It is clear that doubling the resolution leads to an error about four times smaller, verifying our second-order algorithm.

3.5.1 Whistler Wave

Following Tóth et al. (2008) (henceforth referred to as T08), we initialize a 1D whistler wave in the x-direction with right hand circular polarization. The grid domain is $-100 < x < 100$ with periodic boundary conditions. The constant initial conditions are $\rho = 1$, $v_x = 0$, $B_x = 100$, and $p = 1$. We set the wave length $\lambda = 200$, so $k_x = 2\pi/\lambda = \pi/100$. The phase speed of this whistler wave relative to the fluid is:

$$v_\phi = \frac{\delta_i v_a k}{2} + \sqrt{v_a^2 + \frac{\delta_i^2 v_a^2 k^2}{4}} \tag{3.16}$$

N	2D Error	Error Ratio
16^2	0.245	-
32^2	0.0616	3.98
64^2	0.0144	4.28
128^2	0.00345	4.17
256^2	0.000842	4.10

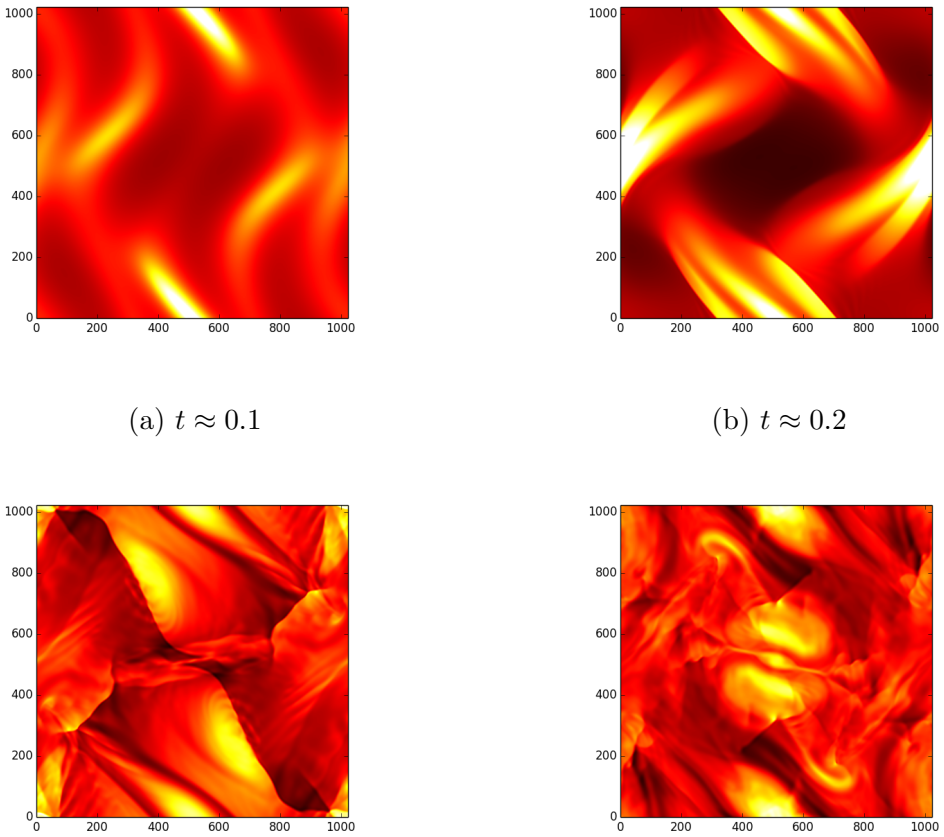
Table 3.2 : Whistler wave errors as calculated in 3.17 for 2D whistler wave simulations. N is the number of cells used in the simulation and Error Ratio is the result of dividing the error in the above row by the error in the current row. It is clear that doubling the resolution leads to an error four times smaller, verifying our second-order algorithm.

where $v_a = |B_x|/\rho = 100$ is the Alfvén speed and δ_i is the ion inertial length. In order to match the phase velocity in T08, we set our normalized $\delta_i = 35.1076$ to get $v_\phi \approx 169.345$ and a period $\tau = \lambda/v_\phi \approx 1.18102$. Then, for the other variables, we set $v_y = -\delta_v \cos k_x$, $v_z = +\delta_v \sin k_x$, $B_y = +\delta_B \cos k_x$, and $B_z = -\delta_B \sin k_x$, with $\delta_B = 0.001$ and $\delta_v = \delta_B * |B_x|/(v_\phi \rho) = 0.00059$. Note that T08 appears to have a typo in their eqns (56) and (57) for their value of δ_v. To check the accuracy of our code, we follow T08's methodology and calculate the numerical error from the difference between the initial and final B_z:

$$E_n = \frac{\Sigma_{i=1}^{n} |B_{z,i}(\tau) - B_{z,i}(0)|}{\Sigma_{i=1}^{n} |B_{z,i}(0)|} \tag{3.17}$$

where i is the grid cell index. There is no special reason for using B_z; any transverse component of the velocity or magnetic field can be used. Our error results for different resolutions are presented in Table 3.1; it is clear that our scheme is second-order accurate.

For the 2D whistler wave in the xy-plane, we rotate the initial conditions by $\alpha = \tan^{-1} 0.5 = 26.56°$ to obtain a (1,2) translational symmetry. The boundaries are periodic in x, and a sheared zero gradient in the (1,2) direction is applied in y. To maintain periodicity in the x-direction, the grid domain is lengthened to $-100/\cos\alpha < x, y < 100/\cos\alpha$. The



(a) $t \approx 0.1$

(b) $t \approx 0.2$

(c) $t \approx 0.4$

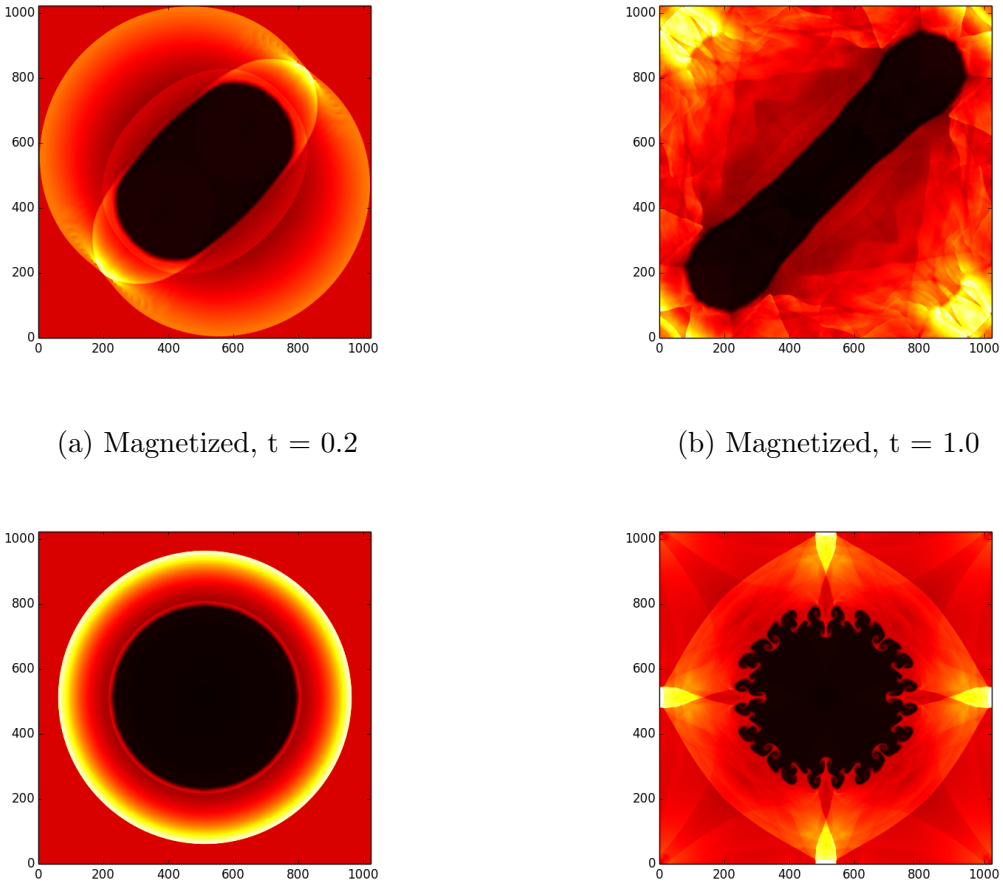
(d) $t \approx 0.8$

Figure 3.1 : Evolution of the Orszag-Tang vortex for a grid size of 1024^2. Density is plotted on a linear scale from 0.05 to 0.5. The ripples caused by whistler waves act to smooth out the shock wave interactions throughout the evolution.

same error calculation as in Equation 3.17 is performed for all $i, j = 1...n_x, 1...n_y$; the results are presented in Table 3.2. Again, it is clear that our scheme is second-order accurate.

3.5.2 Orszag-Tang Vortex

To test the code's handling of complex shocking regions, we simulated a Orszag-Tang vortex with the following initial conditions (e.g. Dai & Woodward (1998)): $\rho = 25/(36\pi)$, $p = 5/(12\pi)$, $v_x = -\sin(2\pi z)$, $v_y = 0.0$, $v_z = \sin(2\pi x)$, $b_x = \sin(2\pi z)/(4\pi)$, $b_y = 0.0$, and $b_z = \sin(2\pi x)/(4\pi)$. We set δ_i to be 10 computational



(a) Magnetized, t = 0.2

(b) Magnetized, t = 1.0

(c) Unmagnetized, t = 0.2

(d) Unmagnetized, t = 1.0

Figure 3.2 : Evolution of the magnetized and unmagnetized 2D blast waves at $t = 0.2$ and $t = 1.0$ with a simulation box of 1024^2. Density is plotted on a linear scale between 0.1 and 3.0. Note the presence of the Richtmyer-Meshkov instability in the unmagnetized run and lack thereof in the magnetized run.

cells long (e.g. 10/1024 for a 1024^2 simulation). The evolution of the vortex for a 1024^2 grid is shown in Figure 3.1. Adding the Hall term to the MHD equations introduces whistler waves to the vortex; their ripples can be seen throughout the grid. In ideal MHD, these ripples are not present and the overall structure is much smoother (e.g. Figure 2.3).

3.5.3 Blast Wave

Finally, to test the code's handling of shocks in 2D and 3D, we simulate a blast wave resulting from an overpressurized region at the center of a grid with $-0.5 < x, y, z < 0.5$. Initially, $\rho = 1$, and $v = 0$. We set $p = 10$ within a radius of 0.1 from the center of the grid and $p = 1$ elsewhere. For the 2D magnetized blast wave, $B_x = B_y = 1/\sqrt{2}$ and $B_z = 0$; for the 3D version we set $B_x = B_y = B_z = 1/\sqrt{3}$. The results are seen in Figures 3.2 and 3.3. Notably, Figure 3.2c, from the unmagnetized blast wave, shows the symmetric wave at early times and Figure 3.2d shows the Richtmyer-Meshkov instability fingers at a later stage of evolution. Figures 3.2a and 3.2b show the results from the magnetized blast wave at the same times; the background magnetic field aligns the fluid bubble and suppresses the instability.

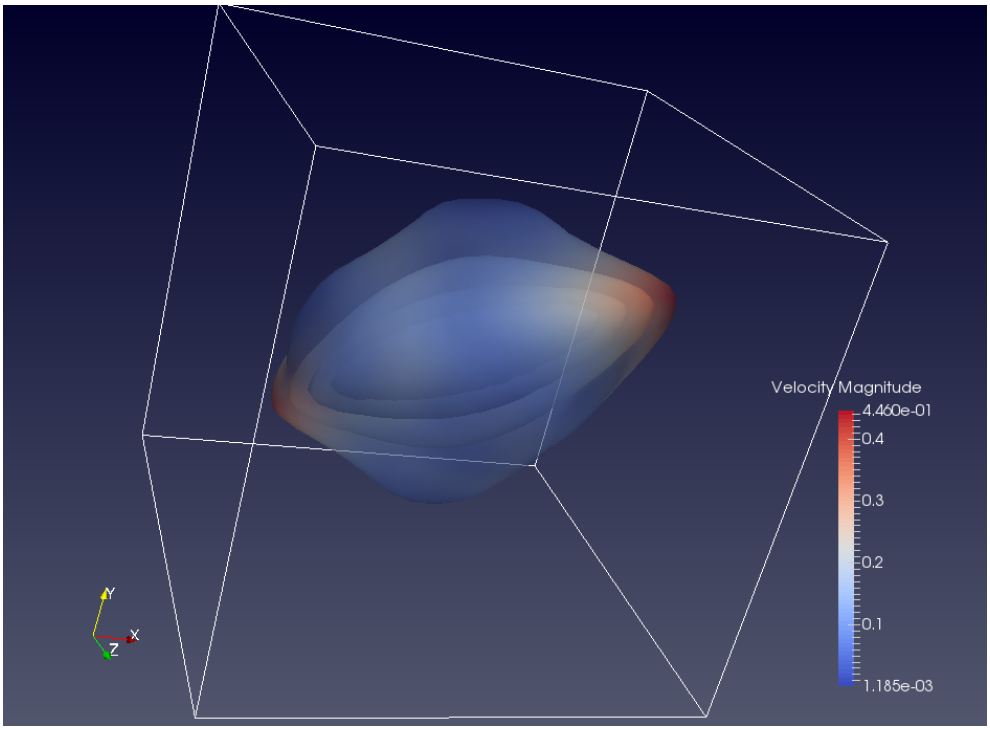


Figure 3.3 : Sample output from the 3D blast wave benchmark, taken at $t = 0.057$. Contours represent 5 levels of density from 0.1 to 0.9, logarithmically spaced. The contours are colored with velocity.

Overall, these benchmarks demonstrate the validity and accuracy of our code algorithm for Hall MHD.

3.6 Scaling Results

In order to ascertain that our code scales well to large numbers of GPUs, we benchmarked it at the Keeneland Initial Delivery System (KIDS) at Georgia Tech. The GPUs at KIDS are M2090s (Fermi architecture) with 6 GB of internal memory each and a NVIDIA compute capability of 2.0. We ran the Orszag-Tang test problem for 300 timesteps, starting the timer at the beginning of the program and ending after the final output. The simulation size ranged from a few ten-thousand cells (32^3) to more than a hundred million cells (512^3), while the number of GPUs ranged from 1 to 128. Due to memory constraints, the larger problem sizes were not able to run on smaller numbers of GPUs. Timing results are presented in Table 3.3; we also present strong-scaling results (Figure 3.4) and weak-scaling results (Figure 3.5).

$N_{cells} \downarrow (N_{GPU} \rightarrow)$	1	2	4	8	16	32	64	128
32^3	2.082	1.497	1.206	0.951	0.931	0.850	0.924	0.950
64^3	11.727	6.939	4.323	2.792	1.908	1.385	1.098	1.010
128^3	85.763	45.775	24.679	13.683	7.814	4.870	2.963	2.067
256^3	-	-	-	92.281	48.948	26.740	14.602	8.443
512^3	-	-	-	-	-	-	93.206	50.090

Table 3.3 : Timing results from KIDS. All simulations ran a Orszag-Tang vortex for 300 timesteps, starting the timer at the beginning of the program and ending after the final output. N_{cells} is the number of cells in the simulation, and N_{GPU} is the number of GPUs the simulation ran on. A "-" indicates that the problem was not able to run on that particular number of GPUs due to memory constraints. These results are visualized in Figures 3.4 and 3.5.

The strong scaling efficiency,

$$\epsilon_s = \frac{100\ t_1}{N_{\mathrm{GPU}}\ t_N}, \tag{3.18}$$

measures how the program run time scales when more processors are used on a fixed problem size. t_1 is the run time for a particular problem size on a single GPU and t_N is the run time on N_{GPU} GPUs. In analyzing our strong-scaling results, we see a steep decline in efficiency for small problems and a more gradual decline in larger problems. The lack of good strong scaling efficiency at smaller problem sizes is due to the relatively larger overhead of inter-node communication, more specifically the GPU→CPU→CPU→GPU data transfer flow. The code must execute two CPU-GPU data transfers every time a message is passed between GPUs. When each GPU is responsible for a smaller and

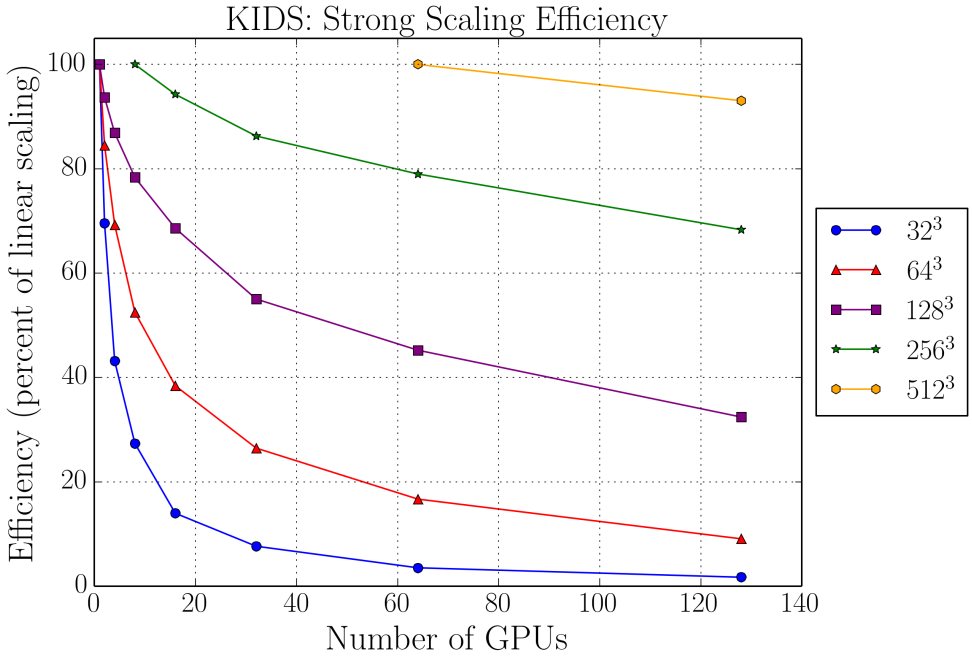


Figure 3.4 : Strong scaling efficiency results from KIDS. Larger problem sizes were not able to run on small numbers of GPUs, so the efficiency was calculated relative to the smallest possible number of GPUs for that size. The decrease in strong scaling efficiency is caused by a larger fraction of clock time devoted to MPI communications instead of GPU calculations.

smaller segment of the computational grid, the GPU kernels execute faster, and the MPI communications take up a increasingly larger proportion of runtime. Additionally, the GPU threads become underutilized and the overall calculation becomes more inefficient. Fortunately, for production runs, we will be pushing against the limits of GPU memory capacity (fully utilizing the GPU threads) so the poor strong scaling at small problem sizes will not be an issue.

The weak scaling efficiency

$$\epsilon_w = \frac{100\ t_1}{t_N}, \tag{3.19}$$

measures how well a program scales when the amount of work per processor stays constant as the number of processors increases. As in the strong-scaling case, underfilling the GPU increases the fraction of wallclock time devoted to inter- and intra-node communication and decreases scaling efficiency. Again, we expect that for production runs utilizing most of the GPU memory, the largest problem sizes will exhibit excellent weak scaling ($\approx 90\%$

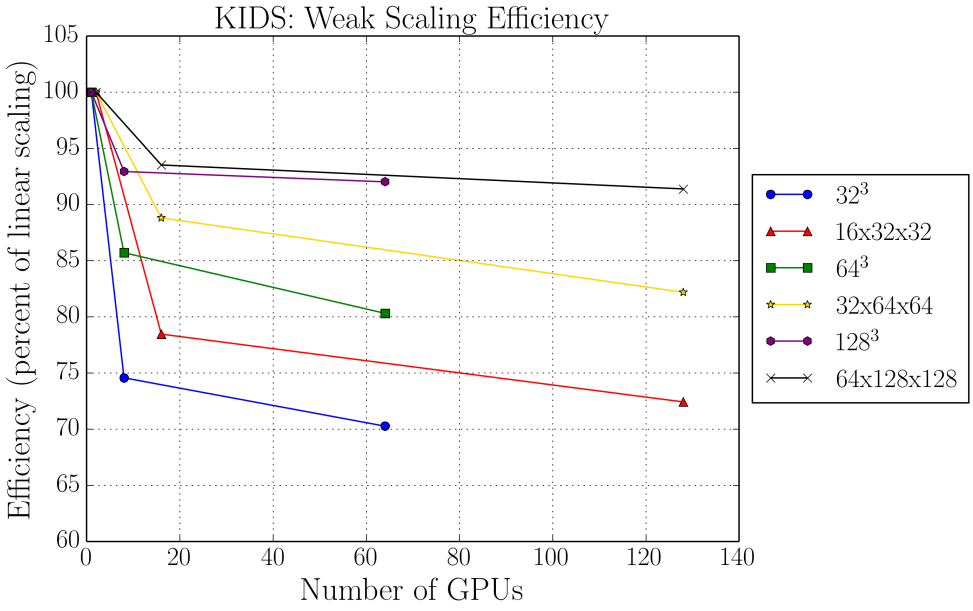


Figure 3.5 : Weak scaling efficiency results from KIDS.

efficiency; c.f. Figure 3.5). We do not need to worry about the lack of scaling at small problems; this simply confirms that GPU memory needs to be filled to see proper scaling and speedups.

At our current level of scaling ($\leq$ 128 GPUs), we note that the memory bandwith of our GPUs is the bottleneck preventing further optimization. However, for larger numbers of GPUs, the fraction of time spent in MPI internode communication will increase. It is uncertain whether MPI or memory bandwidth will be the chief bottleneck for future massively parallel simulations running on thousands of GPUs. The next iteration of GPUDirect (version 3.0), can pass data through Infiniband between GPU memory locations, helping with the internode communication bottleneck. After this is implemented, we are optimistic that our code will exhibit excellent scaling over several hundred to thousands of GPUs. We note that Wong et al. (2014) have implemented a hybrid MPI-GPUDirect approach to reduce communication overhead in the data exchange; however, GPUDirect in their case (version 2.0) is only used between GPUs on a single node and not between GPUs on separate nodes.

3.7 Conclusions

In this paper, we have described our experience in developing a three-dimensional multi-GPU Hall MHD code using CUDA+MPI. We find that the most important optimization for our GPU code was making sure that memory accesses were coalesced. The best way to do this is ascertaining that adjacent threads in a warp access adjacent memory locations; we accomplished this by mapping the three-dimensional thread grid (where one thread is responsible for one computational cell) into one-dimensional thread blocks. Additional optimization strategies we implemented were minimizing the number of kernel function invocations and inter-GPU data transfers.

We also validated our code with benchmarking tests. The one- and two-dimensional whistler wave tests demonstrated the validity of our code's second-order Hall MHD algorithm. The Orszag-Tang vortex and the two- and three-dimensional blast wave tests showed its capability to handle shocks and complex turbulence.

The long term goal for this code is to simulate planetary magnetospheres, such as Earth's, which will require using many hundreds or thousands of GPUs. In preparation for this, we presented strong- and weak-scaling results for intermediate numbers (≤ 128) of GPUs on the supercomputer *KIDS* at Georgia Tech. Though our code does not scale well when the GPUs are underutilized, filling the memory capacity on each GPU results in excellent scaling at large problem sizes. We are optimistic that our code will run well on the large numbers of GPUs required to simulate large planetary magnetospheres, especially if future advances in GPU computing are incorporated in its programming.

3.8 Acknowledgement

This research used resources of the Keeneland Computing Facility at the Georgia Institute of Technology, which is supported by the National Science Foundation under Contract OCI-0910735. CB was supported by the NASA GSRP Fellowship (NASA Grant NNX11AK70H). We thank Homa Karimabadi, who allowed us access to his allocation on the Keeneland/KIDS GPU cluster.

References

Balsara, D. S., & Spicer, D. S. 1999, Journal of Computational Physics, 149, 270

Bard, C. M., & Dorelli, J. C. 2014, Journal of Computational Physics, 259, 444

Batten, P., Clarke, N., Lambert, C., & Causon, D. M. 1997, SIAM J. Sci. Comput., 18, 1553

Belleman, R. G., Bédorf, J., & Zwart, S. F. P. 2008, New Astronomy, 13

Birn, J., Drake, J. F., Shay, M. A., et al. 2001, J. Geophys. Res., 106, 3715

Bowers, K. J., Albright, B. J., Yin, L., Bergen, B., & Kwan, T. J. T. 2008, Physics of Plasmas, 15, 055703

Bowers, K. J., Albright, B. J., Yin, L., et al. 2009, Journal of Physics Conference Series, 180, 012055

Byers, J. A., Cohen, B. I., Condit, W. C., & Hanson, J. D. 1978, Journal of Computational Physics, 27, 363

Courant, R., Friedrichs, K., & Lewy, H. 1928, Mathematische Annalen, 100, 32

Dai, W., & Woodward, P. R. 1998, Astrophys. J., 494, 317

Dawson, J. M. 1983, Reviews of Modern Physics, 55, 403

Dedner, A., Kemm, F., Kröner, D., et al. 2002, J. Computational Phys., 175, 645

Evans, C. R., & Hawley, J. F. 1988, Astrophys. J., 332, 659

Hewett, D. W., & Nielson, C. W. 1978, Journal of Computational Physics, 29, 219

Kunz, M. W., Stone, J. M., & Bai, X.-N. 2014, Journal of Computational Physics, 259, 154

Kylasa, S. B., Aktulga, H. M., & Grama, A. Y. 2014, Journal of Computational Physics, 272, 343

Lee, D., & Deane, A. E. 2009, Journal of Computational Physics, 228, 952

Miyoshi, T., & Kusano, K. 2005, Journal of Computational Physics, 208, 315

Nitadori, K., & Aarseth, S. J. 2012, MNRAS, 424, 545

NVIDIA White Paper. 2009, NVIDIA's Next Generation CUDA Compute Architecture: Fermi, Tech. rep., NVIDIA Corporation

Roe, P. L. 1981, Journal of Computational Physics, 43, 357

Schive, H.-Y., Tsai, Y.-C., & Chiueh, T. 2010, Astrophys. J. Suppl., 186, 457

Sovinec, C., Glasser, A., Gianakon, T., et al. 2004, J. Comp. Phys., 195, 355

Stone, J. M., Gardiner, T. A., Teuben, P., Hawley, J. F., & Simon, J. B. 2008, Astrophys. J., Suppl. Ser., 178, 137

Toro, E. F. 1999, Riemann solvers and numerical methods for fluid dynamics: A practical introduction (Springer-Verlag)

Toro, E. F., Spruce, M., & Speares, W. 1994, Shock Waves, 4, 25

Tóth, G., Ma, Y., & Gombosi, T. I. 2008, Journal of Computational Physics, 227, 6967

van Leer, B. 1984, SIAM J. Sci. Statist. Comput., 5, 1

Winske, D., Yin, L., Omidi, N., & et al. 2003, in Lecture Notes in Physics, Berlin Springer Verlag, Vol. 615, Space Plasma Simulation, ed. J. Büchner, C. Dum, & M. Scholer, 136–165

Wong, H.-C., Wong, U.-H., Feng, X., & Tang, Z. 2011, Comp. Phys. Comm., 182, 2132

Wong, U.-H., Aoki, T., & Wong, H.-C. 2014, Computer Physics Communications, 185, 1901

Chapter 4

Effect of a magnetic field on massive star winds I: mass-loss and velocity for a dipole field

A version of this chapter has been submitted to the Monthy Notices of the Royal Astronomical Society

Bard and Townsend 2016, MNRAS, submitted

What you learn today, for no reason at all, will help you discover all the wonderful secrets of tomorrow.

Norman Juster

Abstract

We generalize the Rigid-Field Hydrodynamic equations to accommodate arbitrary magnetic field topologies, resulting in a new Arbitrary Rigid-Field Hydrodynamic (ARFHD) formalism. We undertake a critical point calculation of the steady-state ARFHD equations with a CAK-type radiative acceleration and determine the effects of a dipole magnetic field on the usual CAK mass-loss rate and velocity structure. Enforcing the proper optically-thin limit for the radiative line-acceleration is found to decrease both the mass-loss and wind acceleration, while rotation boosts both properties. We define optically-thin-correction and rotation parameters to quantify these effects on the global mass-loss rate and develop scaling laws for the surface mass-flux as a function of surface colatitude. These scaling laws are found to agree with previous laws derived from magnetohydrodynamic simulations of magnetospheres. The dipole magnetosphere velocity structure is found to differ from a global beta-velocity law, which contradicts a central assumption of the previously-developed XADM model of X-ray emission from magnetospheres.

4.1 Introduction

In the last decade, spectropolarimetric surveys of OB stars have revealed that about 5-10% of these massive stars have large-scale, organized magnetic fields (Magnetism in Massive Stars: Wade et al. 2014; B-Fields in OB Stars: Morel et al. 2015). Such detectable magnetic fields ($B \gtrsim 100\,\mathrm{G}$) have a significant effect on the stellar wind, both channelling and trapping plasma within a stellar magnetosphere. This accumulated plasma produces extrastellar emission in optical (e.g. Howarth et al. 2007, Bohlender & Monin 2011, Grunhut et al. 2012 and references therein), infrared (Eikenberry et al. 2014), radio (Linsky et al. 1992; Chandra et al. 2015), and X-ray (Nazé et al. 2014, 2015). Furthermore, this emission exhibits a rotational modulation as the plasma is forced by the magnetic field to co-rotate with the star.

Similar advances in magnetosphere theory have also followed, starting with the pioneering magnetohydrodynamics (MHD) simulations of ud-Doula & Owocki (2002). They developed a "wind magnetic confinement parameter" to characterize the interplay between the stellar magnetic field and flow:

$$\eta_* \equiv \frac{B_{\rm eq}^2 R_*^2}{\dot{M}_{B=0} v_{\infty,B=0}}, \tag{4.1}$$

with $\dot{M}_{B=0}$ and $v_{\infty,B=0}$ being the stellar mass-loss rate and terminal velocity if the star had no magnetic field.

The confinement parameter η_* has become the canonical value adopted in scaling relations to explain the size (ud-Doula & Owocki 2002), the mass-loss (ud-Doula et al. 2008), the spin-down (ud-Doula et al. 2009), and, with the critical rotation fraction ω, the classification (Petit et al. 2013) of magnetospheres. However, η_* itself depends on non-

magnetic values, ignoring any effects of the magnetic field. How does the magnetic field change the mass-loss rate and velocity? Can we use these new values to make a better confinement parameter?

Traditionally, $\dot{M}$ and v_∞ have been determined by analyzing the equation of motion for a line-driven wind (Castor et al. 1975; hereafter CAK) and solving for the so-called "critical point". Over the years, various modifications to the base CAK model (finite-disk effect: Friend & Abbott 1986, Pauldrach et al. 1986; depth-dependent force multiplier parameters: Kudritzki 2002) have led to more realistic predictions of the mass-loss and terminal velocities. Other methods have been developed to improve on these estimates, such as a Monte Carlo method (Vink et al. 2000; Noebauer & Sim 2015) and a scattering source function technique (Sundqvist & Owocki 2015). For now, we use the CAK line-driving force in order to take the first steps towards understanding the effect of a dipole field on a stellar wind.

In this paper, we present and study the Arbitrary Rigid-Field Hydrodynamics (ARFHD) equations, an extension of Rigid-Field Hydrodynamics (RFHD) (Townsend et al. 2007) to account for non-dipole magnetic geometries. RFHD was originally developed as an extension of the Rigidly Rotating Magnetosphere (RRM) model (Townsend & Owocki 2005) for centrifugal magnetospheres, whose large magnetic fields make MHD simulations very impractical. In this ansatz, the magnetic fields are assumed to be completely rigid ($\eta_* \to \infty$), channeling the stellar wind along quasi-one-dimensional flux tubes. This allows each field line to be studied and simulated independently from one another, though this does miss important multi-dimensional effects present in the MHD simulations. In essence, the MHD studies approach the subject of massive-star magnetospheres from the regime of low magnetic confinement; ARFHD approaches this subject from the opposite regime of strong magnetic confinement. By blending both studies, we can set limits on the behavior

of magnetospheres.

In Section 4.2, we present the reformulated ARFHD equations and define all the terms, including external sources of acceleration and cooling. Following this, we develop the critical point equations for an arbitrary magnetic configuration in Section 4.3 and an algorithm for determing the critical point location in Section 4.4. Section 4.5 details the implementation and application of an aligned magnetic dipole radiation-driven wind model which includes the effect of stellar rotation. We present analytic scalings of the surface mass-flux in Section 4.6 and model results for the critical point location (Section 4.7), velocity structure (Section 4.8), and, finally, the global mass-loss rate (Section 4.9).

4.2 Arbitrary Rigid-Field Hydrodynamic Equations

Following Townsend et al. (2007) (hereafter T07), we extend the Rigid-Field Hydrodynamics (RFHD) model to incorporate arbitrary magnetic field line configurations, creating an Arbitrary Rigid-Field Hydrodynamics (ARFHD) model. In this section, we recap the key assumptions and equations of RFHD, with additional commentary pertaining to ARFHD when relevant.

In the model, the key assumption is that the magnetic field is sufficiently strong to be effectively rigid (corresponding to $\eta_* \to \infty$). This rigid field is tethered to the star and co-rotates with it. Additionally, since the magnetic Reynolds number in the magnetosphere is so large ($\sim 10^{15}$), we assume that the "frozen flux" condition of ideal MHD applies. As a result, the stiff magnetic field channels magnetospheric plasma flows along the field lines. These trajectories are pre-determined from the chosen stellar magnetic topology, though the plasma state (density, velocity, temperature, etc.) is determined by the hydrodynamics of the flow along each magnetic field line.

The field lines are approximated as quasi-one-dimensional flux tubes, with "quasi-"

referring to their varying cross-sectional area. Under the requirement that local magnetic flux is conserved ($\nabla \cdot \mathbf{B} = \mathbf{0}$), the cross-sectional areas vary inversely with the local magnetic flux density $B \equiv |\mathbf{B}|$. Along these tubes, the plasma flow is subject to both internal (pressure gradients) and external (gravity, centrifugal, radiative driving) forces. Interestingly, in the rigid-field approximation, magnetic and Coriolis forces do not directly influence the dynamics of the flow along field lines since they are always directed perpendicular to the instantaneous velocity vector $\mathbf{v}$.

4.2.1 Euler equations

We can characterize these 1D plasma flows with the conservation form of the quasi-1D Euler equations:

$$\frac{\partial \rho}{\partial t} + \frac{1}{A}\frac{\partial}{\partial s}(A\rho v) = 0 \ , \tag{4.2a}$$

$$\frac{\partial \rho v}{\partial t} + \frac{1}{A}\frac{\partial}{\partial s}(A\rho v^2) + \frac{\partial P}{\partial s} = \rho(g_{\mathrm{eff},s} + g_{\mathrm{rad},s}) \ , \tag{4.2b}$$

$$\frac{\partial \rho \epsilon}{\partial t} + \frac{1}{A}\frac{\partial}{\partial s}(Av(\rho\epsilon + P)) = \rho v(g_{\mathrm{eff},s} + g_{\mathrm{rad},s}) + \Lambda \ , \tag{4.2c}$$

where the independent variables are t, the time, and s, the arc distance along the field line (relative to an arbitrary zero-point). The dependent variables are density ρ, velocity v, pressure P, and total energy per unit mass ϵ. The external sources of energy and momentum are the combined gravitocentrifugal acceleration $g_{\mathrm{eff},s}$ (Section 4.2.5), the radiative driving acceleration $g_{\mathrm{rad},s}$ (Section 4.2.6), and the volumetric energy loss rate $\Lambda \equiv \Lambda_{\mathrm{cc}} + \Lambda_{\mathrm{rc}}$ due to both radiative cooling (rc) and inverse Compton scattering (cc) (Section 4.2.7).

4.2.2 Grid geometry

The "arbitrary" aspect of ARFHD comes from allowing the imposition of any magentic topology, provided that the field lines are physically consistent (i.e. no intersections or discontinuities). This is an improvement over the original RFHD, which allowed only a

dipole topology.

In the reference Cartesian grid comprising the magnetosphere, we define $(0, 0, 0)$ as the center of the star and the z-axis as the stellar rotation pole. Each field line is a three-dimensional space curve $\mathbf{r}(s)$ parameterized by the arc distance s from the northern footpoint, chosen so that the tangent vector $\hat{\mathbf{s}} = d\mathbf{r}/ds$ is everywhere parallel to the local magnetic field vector $\mathbf{B}$. We use the sign of the velocity to indicate the direction of flow; positive (negative) means that the plasma is flowing in the direction of increasing (decreasing) s.

The creation of magnetic topologies is outside the scope of this paper, though there has recently been great success in reconstructing magnetic fields of OB stars using surface spectropolarimetry and source-surface reconstruction (Donati et al. 2006; Kochukhov et al. 2011).

4.2.3 Equations of state and energy

In ARFHD, we assume an ideal gas:

$$P = \frac{\rho k_b T}{\bar{\mu}} \tag{4.3}$$

with the Boltzmann constant k_b and $\bar{\mu} \equiv \mu u_{\rm amu}$ with $u_{\rm amu}$ the atomic mass unit. The mean molecular weight μ is determined by an expression appropriate to a fully ionized mixture:

$$\mu = \left[2X_H + \frac{3}{4}(1 - X_H - Z) + \frac{Z}{2}\right]^{-1} \tag{4.4}$$

with X_H and Z the usual hydrogen and metal mass fractions. Similarly, for a fully ionized plasma, we define a mean molecular weight per hydrogen atom

$$\bar{\mu}_p = u_{\rm amu}/X_H, \tag{4.5}$$

and a mean molecular weight per free electron

$$\bar{\mu}_e = 2u_{\rm amu}/(1 + X_H). \tag{4.6}$$

The electron scattering opacity is

$$\kappa_e = \sigma_T/\bar{\mu}_e, \tag{4.7}$$

with σ_T the Thomson scattering cross-section.

The accompanying equation for the total energy per unit mass is:

$$\epsilon = \frac{v^2}{2} + \frac{P}{\rho(\gamma - 1)} \ , \tag{4.8}$$

with γ the usual ratio of specific heats, 5/3 for a monatomic gas.

4.2.4 Stellar surface properties

Due to rotation, the stellar surface is centrifugally distorted. In the Roche approximation, with the assumptions of a point-like mass distribution and uniform stellar rotation, the surface is an equipotential whose radius R_* varies with rotational colatitude θ as:

$$\frac{R_*}{R_p} = \frac{3}{\omega \sin\theta} \cos\left[\frac{\pi + \cos^{-1}(\omega \sin\theta)}{3}\right] \ . \tag{4.9}$$

Here, after defining Ω as the angular rotation frequency, M_* the stellar mass, and R_p the stellar polar radius,

$$\omega \equiv \Omega \sqrt{\frac{27 R_p^3}{8GM_*(1 - \Gamma_{\rm el})}} \tag{4.10}$$

is the normalized rotation angular frequency, with $\omega = 1$ corresponding to critical rotation. $\Gamma_{\rm el}$ is the Eddington parameter, defined in the next section.

4.2.5 Gravitocentrifugal acceleration

The effective gravity, $g_{\rm eff,s}$, is calculated as the combined gravitocentrifugal acceleration as derived from a scalar effective potential $\Phi_{\rm eff}$ and projected along the field line. The effective gravity vector is

$$\mathbf{g}_{\rm eff} = -\nabla \mathbf{\Phi}_{\rm eff} \ . \tag{4.11}$$

Within the Roche approximation, this effective potential is given by

$$\Phi_{\rm eff} = -(1-\Gamma_{\rm el})\frac{GM_*}{r} - \frac{1}{2}\Omega^2 \bar{r}^2 \ , \tag{4.12}$$

where we take into account the effective reduction in gravity due to the outward force from the electron scattering continuum through the Eddington parameter $\Gamma_{\rm el} \equiv \kappa_e L_*/(4\pi c G M_*)$. In the centrifugal force term, $\bar{r} = |\bar{\mathbf{r}}|$ with $\bar{\mathbf{r}} = [x, y, 0]$ the vector drawn from the rotation axis to the position at $\mathbf{r}$.

In order to obtain $g_{\rm eff,s}$ in Equation 4.2b, we need to translate $\mathbf{g}_{\rm eff}$ into an acceleration *along* the field line:

$$\begin{aligned} g_{\rm eff,s} = \mathbf{g}_{\rm eff} \cdot \hat{\mathbf{s}} &= -(1-\Gamma_{\rm el})GM_* \frac{\psi}{r^2} + \Omega^2 \bar{r}\bar{\psi} \\ &= (1-\Gamma_{\rm el})GM_* \left(\frac{-\psi}{r^2} + \frac{8\omega^2}{27 R_p^3} \bar{r}\bar{\psi} \right) , \end{aligned} \tag{4.13}$$

with $\psi \equiv \hat{\mathbf{r}} \cdot \hat{\mathbf{s}}$ and likewise $\bar{\psi} \equiv \hat{\bar{\mathbf{r}}} \cdot \hat{\mathbf{s}}$. Here, $\hat{\mathbf{r}}$ is the unit radial vector, and $\hat{\bar{\mathbf{r}}}$ is the unit vector parallel to $\bar{\mathbf{r}}$.

In our rotation analysis (Section 4.6.3), we do not take into account the effect of rotational gravity darkening on stellar luminosity (Gayley & Owocki 2000). This will be deferred to future studies.

4.2.6 Radiative driving

The chief mechanism for wind acceleration is radiation line-driving. To quantify this, we implement the Owocki et al. (1988) version of the usual CAK formalism for line-driven

stellar winds. Assuming that the star is a point source of radiation, the acceleration is:

$$\mathbf{g}_{\rm rad} = \frac{\kappa_e \bar{Q} L_*}{4\pi r^2 c} \frac{(1+\tau_{\rm sob})^{1-\alpha} - 1}{(1-\alpha)\tau_{\rm sob}} \, \hat{\mathbf{r}} \, , \tag{4.14}$$

where $\bar{Q}$ is the dimensionless line strength parameter introduced by Gayley (1995), α is the CAK-power law index, and

$$\tau_{\rm sob} \equiv \frac{c\rho\kappa_e\bar{Q}}{|\delta_v|} \tag{4.15}$$

is the Sobolev optical depth.

For δ_v, the local velocity gradient, we follow the same procedure as T07 (see their Section 2.5) and assume that the polar velocity derivative vanishes. Thus, we adopt the approximation $\delta_v \approx \partial v/\partial s$.

At low $\tau_{\rm sob}$, Equation 4.14 correctly reduces to the optically-thin line force. This is an improvement over the previous RFHD implementation, which led to an infinitely large radiative acceleration at zero density (see T07 Equation 25). Finally, we take $g_{{\rm rad},s} = \mathbf{g}_{\rm rad} \cdot \hat{\mathbf{B}}$ to get the radiative acceleration along the field line, giving us a final expression:

$$g_{{\rm rad},s} = \frac{\kappa_e \bar{Q} L_*}{4\pi r^2 c} \frac{(1+\tau_{\rm sob})^{1-\alpha} - 1}{(1-\alpha)\tau_{\rm sob}} \, \psi \, . \tag{4.16}$$

This is a rather simplistic view of line-driven winds, but we emphasize that we are not making any unique insights into the inherent nature of line-driven acceleration. Rather, we are taking the first steps into understanding how a magnetic field affects a line-driven wind. For more detailed massive-star wind models, see e.g. Kudritzki (2002) (modified CAK [mCAK]); Müller & Vink (2008) (Monte Carlo technique); Sundqvist & Owocki (2015) (scattering).

4.2.7 Cooling

The volumetric cooling rate Λ is evaluated as the sum of an inverse Compton cooling term $\Lambda_{\rm cc}$ and a radiative cooling term $\Lambda_{\rm rc}$. We calculate $\Lambda_{\rm cc}$ from the electron pressure

$n_e k_b T$ as per Equation 4 of White & Chen (1995):

$$\Lambda_{\rm cc} = \frac{-4\sigma_T}{m_e c} n_e k_b T U_{\rm rad} \tag{4.17}$$

with $U_{\rm rad} = L_*/(4\pi r^2 c)$ the stellar radiation energy density and n_e the electron number density. We calculate $\Lambda_{\rm rc}$ as:

$$\Lambda_{\rm rc} = -n_e n_p \Lambda_{\rm rad} = -\frac{\rho^2 \Lambda_{\rm rad}}{\bar{\mu}_e \bar{\mu}_p} = -\rho^2 \Lambda_m \tag{4.18}$$

where n_p is the proton number density. $\Lambda_{\rm rad}$ is the optically thin cooling function, typically obtained in tabular form from a plasma emission code (Schure et al. 2009), and we define a mass-weighed form $\Lambda_m \equiv \Lambda/\bar{\mu}_e\bar{\mu}_p$.

4.3 Steady-state wind analysis

Before we analyze the ARFHD equations presented above (Section 4.2.1), we first simplify using several assumptions relevant to the wind-driving region close to the star. We assume the wind is isothermal, and, following Drew (1989), set the temperature T equal to the stellar effective temperature $T_{\rm eff}$. Also, we assume that the wind remains optically thick ($\tau_{\rm sob} \gg 1$) and that it has reached a steady state.

In deriving the magnetospheric wind equation of motion, it is convenient to use the primitive variable form of Equation 4.2a-Equation 4.2c. Under our stated assumptions, these equations reduce to

$$\frac{\partial}{\partial s}(A\rho v) = 0, \tag{4.19}$$

$$v\frac{\partial v}{\partial s} + \frac{1}{\rho}\frac{\partial P}{\partial s} = \rho g, \tag{4.20}$$

$$P = c_{\rm s}^2 \rho, \tag{4.21}$$

where $\lambda = \partial A/\partial s/A$ is the areal gradient term and $c_{\rm s}^2 = P/\rho = k_b T_{\rm eff}/\bar{\mu}$ is the isothermal

sound speed. We can then derive an equation of motion:

$$vv'\left(1-c_{\rm s}^2/v^2\right)-g_{{\rm eff},s}-c_{\rm s}^2\lambda-g_{{\rm rad},s}=0, \tag{4.22}$$

where we define $v'\equiv\partial v/\partial s$.

In the optically thick limit, $g_{{\rm rad},s}$ reduces to

$$g_{{\rm rad},s}=\frac{\kappa_e\bar{Q}L_*}{4\pi r^2c}\frac{\tau_{\rm sob}^{-\alpha}}{1-\alpha}\psi, \tag{4.23}$$

which is equivalent to Equation 25 of T07. Substituting in the expression (4.15) for $\tau_{\rm sob}$, and then eliminating the explicit dependence on density via the continuity equation (4.19), we obtain after some algebra

$$g_{{\rm rad},s}=\Delta\left(\frac{A}{A_*}\right)^{\alpha}\frac{\psi}{r^2}|vv'|^{\alpha}. \tag{4.24}$$

Here,

$$\Delta\equiv\frac{\left(\bar{Q}\Gamma_{\rm el}GM_*\right)^{1-\alpha}}{1-\alpha}\left(\frac{L_*}{4\pi\dot{m}_*c^2}\right)^{\alpha}. \tag{4.25}$$

parameterizes the mass-loss rate, with A_* the area of the flux tube at the stellar surface, and $\dot{m}_*$ is the mass flux into the tube.

Now that we have derived a equation of motion (4.22), we can solve for the values at the critical point. For simplicity, we shall neglect the Parker term $c_{\rm s}^2\lambda$ since it is typically of order $c_{\rm s}^2/v_{\rm esc}^2\approx 0.001$ relative to the gravitational acceleration term, where $v_{\rm esc}=[2GM_*(1-\Gamma_{\rm el})/R_*]^{1/2}$ is the escape velocity at the stellar surface. Defining

$$F[s,y,u]\equiv y(1-1/u^2)-g_{{\rm eff},s}-\Delta\left(\frac{A}{A_*}\right)^{\alpha}\frac{\psi}{r^2}|y|^{\alpha}, \tag{4.26}$$

where $u\equiv v/c_{\rm s}$ and $y\equiv vv'$, the equation of motion can be written as

$$F[s,y,u]=0. \tag{4.27}$$

Following CAK and Abbott (1980), we fix the wind critical point by the singularity condition

$$\frac{\partial F}{\partial y} = 0. \tag{4.28}$$

In order that the velocity gradient $\mathrm{d}v/\mathrm{d}s$ remain bounded at the critical point, it is also necessary that the regularity condition

$$\frac{\partial F}{\partial s} + \left(\frac{y}{c_{\mathrm{s}}^2 u}\right)\frac{\partial F}{\partial u} = 0 \tag{4.29}$$

be satisfied at the critical point (this can be derived by taking the total derivative of Equation 4.27).

For a given choice of the parameter Δ, Equations (4.27–4.29) can in principle be solved to find the unknowns (s, y, u) at the critical point (if solutions exist). However, in his analysis of spherical wind outflows, Bjorkman (1995) found that the critical point location was quite sensitive to the surface mass-flux. He concluded that it was much easier to fix the location and then solve for the mass flux. Following his lead, we therefore treat the critical point location s_{crit} as a free parameter, and solve for Δ, y and s at $s = s_{\mathrm{crit}}$. We defer until later (Section 4.4) the question of how to choose s_{crit} appropriately.

We leave the mathematical derivation of the critical values to Appendix B. In this derivation, we obtain a special function

$$\frac{y_c}{u_c^2} = \Phi \equiv \pm\sqrt{\frac{\partial g_{\mathrm{eff},s}}{\partial s} - \frac{g_{\mathrm{eff},s}}{1-\alpha}\left[\alpha\lambda + \frac{1}{\psi}\frac{\partial \psi}{\partial s} - \frac{2\psi}{r}\right]}. \tag{4.30}$$

Due to our sign convention (Section 4.2.2), Φ can be positive or negative, corresponding to a positive/negative ψ. This is because plasma accelerating away from the stellar surface flows in the direction of increasing (decreasing) s for positive (negative) ψ.

Either way, we solve for the critical velocity:

$$u_c^2 = 1 \mp \frac{\sqrt{2}\alpha}{(1-\alpha)c_{\mathrm{s}}\Phi} g_{\mathrm{eff},s}, \tag{4.31}$$

with the top (bottom) resulting from the positive (negative) root of Equation 4.30. Similarly, we obtain the critical y:

$$y_c = \pm\frac{c_s\Phi}{\sqrt{2}} - \frac{\alpha}{1-\alpha}g_{\mathrm{eff},s}. \tag{4.32}$$

Remembering that our $y = vv'$, this critical value is nearly identical to CAK Equation 45 in the zero sound-speed limit ($\Phi \to 0$), with differences coming from a factor ψ resulting from non-radial paths and from rotational acceleration.

Finally, we can solve for our critical eigenvalue and surface mass-flux:

$$\Delta_c = -\frac{g_{\mathrm{eff},s}}{(1-\alpha)}\left(\frac{A_*}{A}\right)^{\alpha}\frac{r^2}{\psi|y_c|^{\alpha}}, \tag{4.33}$$

$$\dot{m}_* = \left(\frac{(\bar{Q}\Gamma_{\mathrm{el}}GM_*)^{1-\alpha}}{1-\alpha}\right)^{1/\alpha}\frac{L_*}{4\pi c^2}\frac{1}{\Delta_c^{1/\alpha}}, \tag{4.34}$$

A similar procedure for the general critical point values is presented in Appendix C.

As noted by Marlborough & Zamir (1984) and Curé & Rial (2004), this type of analysis implies a range to the allowable critical point position. Here, we require that $\Phi^2 > 0$ and $\Delta_c > 0$ at s_{crit} for a trans-critical steady-state wind.

Although our ansatz assumes a magnetic star, we can pretend there is no field by defining a radial magnetic topology with a spherically expanding cross-section. This results in $\psi = 1$ and $A_*/A = R_*^2/r^2$. In the limit of zero rotation and zero sound speed ($\Phi \to 0$),

Table 4.1 : Stellar and wind parameters used throughout this paper to represent a typical magnetic B-type star with a centrifugal magnetosphere and an O-type star with a dynamical magnetosphere. Values are taken from Table 2 in Bjorkman (1995) for the B star and ud-Doula et al. (2014) for the O star.

Type	M_*	R_p	T_{eff}	α	Γ_{el}	$\bar{Q}$	B_p	η_*
B	9.0 $M_\odot$	4.5 $R_\odot$	21000 K	0.56	9.27×10^{-3}	1025.14	11 kG	4.29×10^5
O	50 $M_\odot$	19 $R_\odot$	41860 K	0.6	0.5	500	3.715 kG	100

making these subsitutions in Equation 4.33 yields

$$\begin{aligned}\Delta_c &= \frac{(1-\Gamma_{\rm el})GM_*}{(1-\alpha)r^2}\frac{R_*^{2\alpha}}{r^{2\alpha-2}}\frac{|r^{2\alpha}|}{|\frac{\alpha}{1-\alpha}(1-\Gamma_{\rm el})GM_*|^\alpha} \\ &= \frac{1}{[\alpha/(1-\alpha)]^\alpha}\frac{[GM_*(1-\Gamma_{\rm el})]^{1-\alpha}R_*^{2\alpha}}{1-\alpha},\end{aligned} \tag{4.35}$$

and a surface mass-flux

$$\dot{m}_* = \frac{\alpha}{1-\alpha}\left(\frac{\bar{Q}\Gamma_{\rm el}}{1-\Gamma_{\rm el}}\right)^{\frac{1-\alpha}{\alpha}}\frac{L_*}{4\pi R_*^2 c^2}. \tag{4.36}$$

As defined by Gayley (1995), using his $\bar{Q}$ formalism, the CAK mass-loss rate is

$$\dot{M}_{\rm CAK} \equiv \frac{\alpha}{1-\alpha}\left(\frac{\bar{Q}\Gamma_{\rm el}}{1-\Gamma_{\rm el}}\right)^{\frac{1-\alpha}{\alpha}}\frac{L_*}{c^2}, \tag{4.37}$$

and we see that our derived surface mass-flux is $\dot{m}_* = \dot{M}_{\rm CAK}/(4\pi R_*^2)$. This demonstrates that our general equations correctly reproduce the usual CAK mass-loss rate in the proper limit.

4.4 Critical point calculation

The critical point location, $s_{\rm crit}$, is required to accurately calculate the surface mass-flux, which, through the density, sets the level of radiative driving and emission throughout the magnetosphere. Since $s_{\rm crit}$ is a free parameter in the above critical point calculation, we must provide a boundary condition to obtain $s_{\rm crit}$ and the resulting critical surface mass-flux. This is especially important for calculating a dipole star's mass-loss rate, since the critical mass-flux is sensitive to the critical point location (Section 4.6.1). Following Equation 24 of Bjorkman (1995), we define the boundary density such that the resultant electron scattering optical depth $\tau_{es} \approx 1$ at the stellar surface. Thus,

$$\rho_0 = \frac{\tau_{es}}{\kappa_e H} \approx \frac{(1-\Gamma_{\rm el})GM_*}{\kappa_e R_*^2 c_{\rm s}^2} \tag{4.38}$$

where $H = c_s^2/g = c_s^2 R_*^2/[(1-\Gamma_{el})GM_*]$ is the pressure scale height, corrected for the electron scattering acceleration. Using the continuity equation and the critical surface mass flux, we can solve for the boundary velocity:

$$v_0 = \frac{\rho_* v_*}{\rho_0} = \frac{\dot{m}_*}{\rho_0} \tag{4.39}$$

where $\dot{m}_* \equiv \rho_* v_*$ is defined as the surface mass-flux into the field line. Since this introduces a dependence on s_{crit} for both the boundary and critical point velocities, we must use an iterative algorithm to satisfy both conditions simultaneously.

The set of equations we use for the integration are the wind equation of motion (Equation 4.22), the steady-state continuity equation $d\rho/ds = -\rho[\lambda + (dv/ds)/v]$, and the isothermal approximation $dP/ds = c_s^2 d\rho/ds$. Equation 4.22 is not easily solved for v', however, since there are multiple roots. There are usually three roots: two positive and one negative. This differs slightly from Kudritzki (2002), who found two positive roots. We find the same two roots and an extra one since τ_{sob} has a dependency on the absolute value of v'. Inside of the sonic point, though, there are instead one positive and two negative roots. We always choose a positive root in order to enforce an accelerating outflow; the negative root corresponds to an accretion flow towards the star. When multiple positive roots exist, we choose the smaller root in the subcritical region and the larger root beyond the critical point (Cassinelli 1979; Abbott 1980).

Our iterative algorithm for calculating s_{crit} is as follows:

1. Choose trial critical point location.

2. Calculate velocity, mass-flux, density at the critical point.

3. Integrate to boundary, taking the smallest positive root of the possible velocity derivatives.

4. If resulting boundary velocity is too high, move $s_{\rm crit}$ out. Else if too low, move $s_{\rm crit}$ in.

5. Repeat from step 2 until correct boundary values are reached.

As a check, we calculate the critical radius and mass-flux for a straight, spherically-diverging flux tube ($\psi = 1$; $A_*/A = R_*^2/r^2$) for the stellar parameters chosen in Bjorkman (1995) (Table 4.1). We calculate $r_c = s_{\rm crit} = 1.5589 R_*$, which matches well with Bjorkman (1995)'s derived value of 1.5594. The resulting mass-flux, $\dot{m}_* = 9.2516 \times 10^{-8}\,{\rm g/cm^2}$ also fits with his derived surface mass-flux $\dot{M}/4\pi R_p^2 = 9.249 \times 10^{-8}\,{\rm g/cm^2}$.

4.5 Magnetic dipole model

Now that we have developed our general critical equations, we now derive the critical values for a wind channeled by a magnetic dipole whose pole is aligned with the rotation axis. This field forces the plasma to co-rotate with the star (i.e. a magnetosphere). Instead of assuming a radial outflow, we force the plasma to flow along the magnetic flux tubes. Additionally, we define the combined gravitocentrifugal force in the same manner as the ARFHD formulation (Section 4.2.5), i.e. with rigid-body rotation.

While it is possible to solve the critical point equations with s as the independent variable, it is relatively more convenient here to parameterize the spatial variables with $\tilde{\theta}$, the magnetic colatitude. We do this because although the plasma flows along the magnetic field line, most of our external forces are dependent on r. It is easier to set $\tilde{\theta}$ as the spatial variable rather than have to solve for r in terms of s.

In an aligned dipole, the magnetic pole is parallel to the rotational pole (the z-axis in our coordinate system; Section 4.2.2), so we take $\tilde{\theta} = \theta$, where θ is the rotational colatitude.

First, we start with the definition of an aligned dipole field (e.g. T07):

$$\mathbf{B} = \frac{B_0}{2(r/R_p)^3}\left(2\cos\theta\hat{r} + \sin\theta\hat{\theta}\right), \tag{4.40a}$$

$$\hat{B} = \frac{\mathbf{B}}{|\mathbf{B}|} = \frac{2\cos\theta\hat{r} + \sin\theta\hat{\theta}}{\sqrt{1+3\cos^2\theta}}. \tag{4.40b}$$

We note that R_p is the polar radius of the star, not the stellar surface radius R_* (Equation 4.9). From the parametric equation of a dipole field line (e.g. Babel & Montmerle 1997), we have

$$r(\theta) = R_p L \sin^2\theta = r_m \sin^2\theta, \tag{4.41}$$

where $r_m \equiv R_p L$ is the maximum extent of the field line and L is the magnetic shell parameter. Each individual line exists over the range $\theta_m < \theta < \pi - \theta_m$, with $\theta_m = \sin^{-1}\sqrt{R_*/(R_p L)}$ marking the northern magnetic footprint and $\pi - \theta_m$ marking the southern. Each field line can be uniquely identified by L and its magnetic azimuthal coordinate which denotes the half-plane containing that line. For our aligned dipole model, we will place every individual line in the same half-plane and assume azimuthal symmetry. Thus, knowing L or θ_m is sufficient for identifying a particular line.

We can obtain the path length s along the line with

$$ds^2 = dr^2 + r^2 d\theta^2 = r_m^2 \sin^2\theta(1+3\cos^2\theta)d\theta^2, \tag{4.42}$$

which, after integrating, yields

$$s = -\frac{r_m}{2}\left[\frac{\sinh^{-1}(\sqrt{3}\cos\theta)}{\sqrt{3}} + \cos\theta\sqrt{1+3\cos^2\theta}\right] + \text{const.} \tag{4.43}$$

We select our constant of integration to enforce $s = 0$ where the magnetic field comes out of the stellar surface: the northern footprint ($\theta = \theta_m$).

With these definitions of r and s, we can write all of the spatial variables as functions of θ (Appendix D). These can be then be used to solve the critical point values derived in

Section 4.3. The general critical point values (Appendix C) can also be parameterized in this manner, using the same spatial variables.

With our general critical point algorithm established, we now turn our attention to how an aligned dipole magnetic field affects the stellar wind, namely its mass-loss rate and terminal velocity. Additionally, we will study how the stellar rotation rate influences the CAK critial point and resulting wind properties. For this analysis, we generate a grid of 500 dipole field lines with footprints covering the northern hemisphere of the star ($0 < \theta < \pi/2$) in linear space. This is repeated for several critical rotation fractions $\omega = [0.0, 0.2, 0.35, 0.5, 0.65, 0.8]$, giving a total of 3000 dipole lines. Field lines with $L > 100R_p$ are arbitrarily truncated at $R = 100R_p$; the rest of the lines extend to the magnetic equator. This truncation does not affect the critical value calculations since that only depends on the boundary condition and the field line geometry inside of the critical point. Additionally, as we later show, this trunctation radius is larger than the "closure radius" of our model magnetospheres, so our apex velocity calculations will not be affected.

We calculate the critical point location and resulting surface mass-fluxes using the procedure described in Section 4.4. After finding the critical values, we integrate from the critical point to the apex of the field line to obtain the apex velocity, $v_{\rm apex}$. We do this for two different sets of stellar parameters, one representing a centrifugal magnetosphere and one a dynamical magnetosphere (Table 4.1). For the centrifugal magnetosphere star, we use parameters from Bjorkman (1995) representing an early-B star similar to the archetype σ Orionis E.

For the other star, we follow the MHD simulations of ud-Doula & Owocki (2002) and choose a ζ Puppis analogue, representing a dynamical magnetosphere: $M_{\rm eff} = 25\ M_\odot$, $R = 19\ R_\odot$, $T_{\rm eff} = 41860$ K (such that $L_{\rm star} = 10^6 L_\odot$), $\alpha = 0.6$, $\bar{Q} = 500$, and we take solar values for the mass fractions X_H and Z. Since the stellar mass above is an effective mass

and already takes into account the factor of two reduction below the Newtonian mass due to the electron scattering continuum force, we take $M = 50 M_\odot$ and $\Gamma_{\rm el} = 0.5$ in our model.

Since our ansatz assumes an infinite magnetic confinement, the actual magnitude of the dipole field (i.e. B_0 in Equation 4.40a) only matters when estimating which lines are in the closed magnetosphere (Equation 4.75).

4.6 Surface mass flux

4.6.1 Zero rotation in the optically-thick limit

Since magnetic dipole field lines do not come straight out of the stellar surface, the surface mass-flux is tilted relative to a radial mass-flux. Inspired by the MHD simulations presented in ud-Doula & Owocki (2002), Owocki & ud-Doula (2004) (hereafter OD04) used a simple, one-dimensional flow analysis to calculate that the radial mass-flux, $\dot{m}_r$, scales as

$$\dot{m}_r = \mu_B \dot{m}_* = \mu_B^2 \dot{m}_{\rm CAK}, \tag{4.44}$$

where the CAK surface mass-flux is defined as

$$\dot{m}_{\rm CAK} = \frac{\dot{M}_{\rm CAK}}{4\pi R_p^2} = \frac{L_*}{4\pi R_p^2 c^2} \frac{\alpha}{1-\alpha} \left(\frac{\bar{Q}\Gamma_{\rm el}}{1-\Gamma_{\rm el}} \right)^{(1-\alpha)/\alpha}, \tag{4.45}$$

and $\mu_B = \hat{\mathbf{n}} \cdot \hat{\mathbf{B}}$ with $\hat{\mathbf{n}}$ the unit vector normal to the stellar surface. For a non-rotating star, $\hat{\mathbf{n}} = \hat{\mathbf{r}}$. One factor of μ_B results from the geometric projection of the tilted flow onto the stellar surface normal ($\dot{m}_r = \mu_B \dot{m}_*$). The other factor results from projecting a radial radiative line force along the field line. The tension in the magnetic field line negates any acceleration normal to the line, further lowering the critical mass-flux.

We now check this scaling analysis with our dipole model. For simplicity, we note

that $c_s\Phi/\sqrt{2} \ll \alpha/(1-\alpha)\ g_{\mathrm{eff},s}$ and take

$$y_c \approx -\frac{\alpha}{1-\alpha} g_{\mathrm{eff},s}. \tag{4.46}$$

From Equation D29 we obtain

$$\left(\frac{A_*}{A}\right)^{\alpha} = \left(\frac{R_*}{r_c}\right)^{3\alpha} \left(\sqrt{\frac{1+3\cos^2\theta_c}{1+3\cos^2\theta_m}}\right)^{\alpha} \tag{4.47}$$

where θ_c is evaluated at the critical radius r_c for a given field line and R_* is the stellar radius at the footprint colatitude θ_m.

Combining our eigenvalue relation (Equation 4.33) with the above equations, we obtain

$$\Delta_c \approx \frac{-g_{\mathrm{eff},s}|g_{\mathrm{eff},s}|^{-\alpha}}{(1-\alpha)(\frac{\alpha}{1-\alpha})^{\alpha}} \frac{R_*^{3\alpha} r_c^{2-3\alpha}}{\psi_c} \left(\sqrt{\frac{1+3\cos^2\theta_c}{1+3\cos^2\theta_m}}\right)^{\alpha}, \tag{4.48}$$

where ψ_c is evaluated at the critical point. Finally, our surface mass-flux is

$$\begin{aligned}\dot{m}_* &\approx \frac{\alpha}{1-\alpha}\frac{L_*}{4\pi c^2}|g_{\mathrm{eff},s}| \left[\frac{(\bar{Q}\Gamma_{\mathrm{el}} GM_*)^{1-\alpha}}{-g_{\mathrm{eff},s}}\right]^{1/\alpha} \\ &\quad \times \frac{\psi_c^{1/\alpha} r_c^{3-2/\alpha}}{R_*^3} \sqrt{\frac{1+3\cos^2\theta_m}{1+3\cos^2\theta_c}}.\end{aligned} \tag{4.49}$$

For zero rotation, $g_{\mathrm{eff},s} = -(1-\Gamma_{\mathrm{el}})GM_*\psi/r_c^2$ and $R_* = R_p$:

$$\begin{aligned}\dot{m}_* &\approx \frac{\alpha}{1-\alpha}\frac{L_*}{4\pi c^2} \left[\frac{(\bar{Q}\Gamma_{\mathrm{el}} GM_*)^{1-\alpha}}{[(1-\Gamma_{\mathrm{el}})GM_*]^{1-\alpha}}\right]^{1/\alpha} \\ &\quad \times \frac{\psi_c r_c}{R_p^3} \sqrt{\frac{1+3\cos^2\theta_m}{1+3\cos^2\theta_c}} \\ &\approx \dot{m}_{\mathrm{dip}} \psi_c \sqrt{\frac{1+3\cos^2\theta_m}{1+3\cos^2\theta_c}},\end{aligned} \tag{4.50}$$

where we will define $\dot{m}_{\mathrm{dip}} = \dot{m}_{\mathrm{dip}}(r_c)$ as the zero-tilt, zero-rotation, optically-thick surface mass-flux for a magnetic dipole:

$$\begin{aligned}\dot{m}_{\mathrm{dip}} &\equiv \frac{\alpha}{1-\alpha}\frac{L_*}{4\pi c^2} \left(\frac{\bar{Q}\Gamma_{\mathrm{el}}}{1-\Gamma_{\mathrm{el}}}\right)^{\frac{1-\alpha}{\alpha}} \frac{r_c}{R_p^3} \\ &= \dot{m}_{\mathrm{CAK}} \frac{r_c}{R_p}.\end{aligned} \tag{4.51}$$

We can thus think of $\dot{m}_{\rm dip}$ as the CAK surface mass-flux corrected for dipole divergence.

We can generalize this straight-line base term for any magnetically-induced areal expansion with

$$\dot{m}_{\rm gen} \equiv \dot{m}_{\rm CAK} \left(\frac{r_c}{R_p} \right)^{q-2}, \tag{4.52}$$

when the areal expansion is proportional to r^q, $q = 3$ for a dipole.

For our model B star, we calculate $\dot{m}_{\rm dip} \approx 9.53 \times 10^{-8}\,{\rm g/cm^2}$ at the pole, with $r_c = 1.033\ R_p$. If we keep the Φ term instead of neglecting it (Equation 4.32), the model calculated mass-flux ($\dot{m}_{\theta=0}$) is boosted by about 2%, to $\dot{m}_{\theta=0} = 9.709 \times 10^{-8}\,{\rm g/cm^2}$. For our model O star, the polar values are $\dot{m}_{\rm dip} \approx 1.98 \times 10^{-5}\,{\rm g/cm^2}$ (with $r_{c,theta=0} \approx 1.054\ R_p$) and $\dot{m}_{\theta=0} = 2.05 \times 10^{-5}\,{\rm g/cm^2}$, a difference of about 3%. The O-type star has a larger correction than the B star because of its faster sound speed.

We can reproduce Equation 4.44 with several simplifications, which end up canceling each other out. First, we take $r_c = r_{{\rm c},\theta=0}$ as constant for every field line (justified in Section 4.7). Next, we assume that the critical radius is very close to the star ($r_{{\rm c},\theta=0} - R_p \ll R_p$), which allows us to take $\theta_c \approx \theta_m$ and $\psi_c \approx \psi_m = \mu_B$. Finally, we correct for neglecting the Φ term by replacing $\dot{m}_{\rm dip}$ with $\dot{m}_{\theta=0}$ to obtain the scaling relation:

$$\dot{m}_* \sim \mu_B \dot{m}_{\theta=0}. \tag{4.53}$$

Rather conveniently, it turns out that replacing θ_c and ψ_c with the surface values θ_m and ψ_m produces opposite effects which nearly cancel each other out. Overall, we are able to reproduce OD04's general scaling at zero rotation (Figure 4.1), though keeping the exact angular expressions with the constant r_c assumption gives an even better fit.

4.6.2 Zero rotation in the general case

In the general line-force critical analysis (Appendix C), an important parameter arises:

$$\chi \equiv (1 + \tau_{\rm sob})^{-\alpha} \, . \tag{4.54}$$

Physically, it represents the "transparency" of the wind where low χ means a optically thick wind and high χ means a optically thin wind. Alternatively, it represents a "optically-thin correction level" parameter which characterizes the relative importance of using the general form of $g_{{\rm rad},s}$ rather than its optically-thick limit (Equation 4.23). For optically-thick winds, $\tau_{\rm sob} \gg 1$ and the line-acceleration reduces to $g_{\rm rad,CAK} \propto \tau_{\rm sob}^{-\alpha}$. For optically-thin winds,

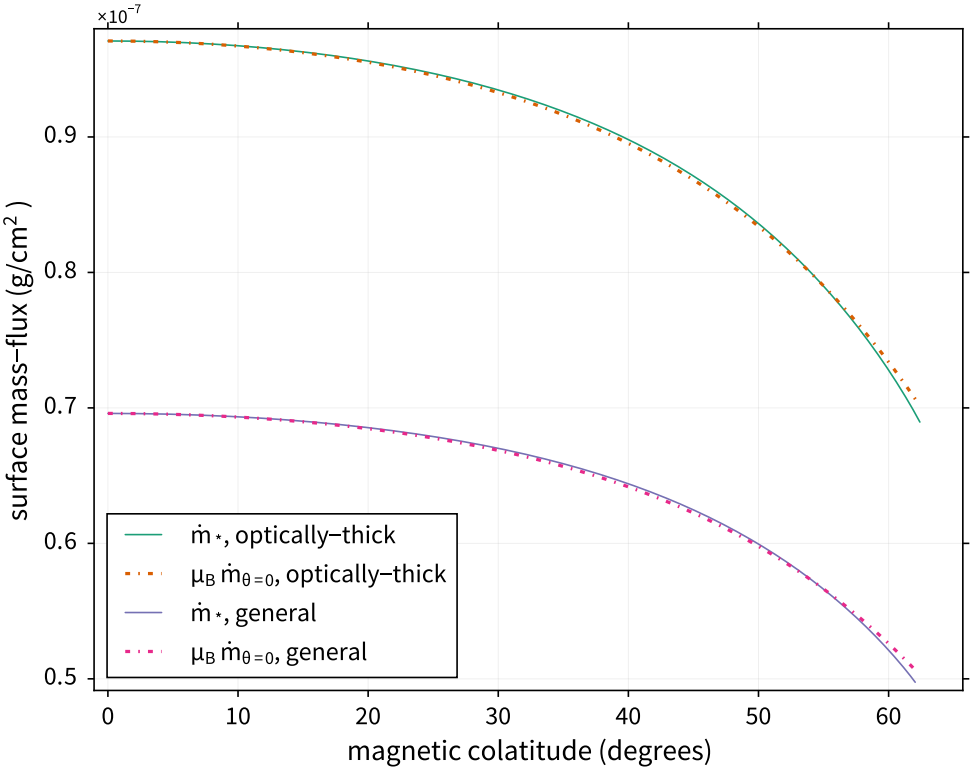


Figure 4.1 : Surface mass-flux ($\dot{m}_*$) as a function of line surface colatitude calculated using both the general and optically-thick line-acceleration for a B-type star (Table 4.1). $\dot{m}_*$ is compared to a general scaling $\mu_B \dot{m}_{\theta=0}$ as derived by OD04 (Equation 4.44), with a different $\dot{m}_{\theta=0}$ for both cases. We find an excellent agreement between the scaling and model, with similar results for the O-type star not shown.

$\tau_{\rm sob} \to 0$ and $\chi \to 1$.

We can repeat the previous section's scaling analysis for a general line force (derivation in Appendix E). With the same approximations as above and zero rotation, we get the scaling

$$\dot{m}_* \approx \dot{m}_{\rm dip}(r_c)\Sigma_0\psi_c\sqrt{\frac{1+3\cos^2\theta_m}{1+3\cos^2\theta_c}}, \tag{4.55}$$

where we have defined an optically-thin-correction (OTC) parameter

$$\Sigma_0 \equiv \dot{m}_{\rm dip,OTC}(r_c)/\dot{m}_{\rm dip}(r_c) = \frac{\left|1-\alpha-\left[\frac{1-\chi_0^{1/\alpha-1}}{1-\chi_0^{1/\alpha}}\right]\right|}{\alpha}, \tag{4.56}$$

with χ_0 as the value of χ evaluated at the critical point for zero rotation.

Notably, the critical value of χ is set by the ratio between the non-radiative external forces (gravity, centrifugal, areal gradient) and the optically thin radiative force (c.f. Equation C18). With zero rotation, $\chi_0 = (1-\Gamma_{\rm el})/(\Gamma_{\rm el}\bar{Q})$ is independent of surface colatitude, which allows us to use a constant Σ_0 across the stellar surface.

Essentially, Σ_0 results from the error in assuming an optically-thick wind. In O stars, the increased luminosity drives a much higher surface mass-flux, leading to a more optically-thick wind than in B stars. Thus, χ_0 is smaller and Σ_0 is closer to unity for more massive stars.

Σ_0 allows us to correct our mass-flux estimates, though the critical radius (and thus the base mass-flux) will be different between the general and optically-thick cases (Section 4.7). For an optically thick wind, $\chi \to 0$ and $\Sigma_0 \to 1$, reproducing $\dot{m}_{\rm dip}$. For an optically thin wind, using l'Hôpital's rule yields

$$\lim_{\chi\to 1}\Sigma_0 = \frac{|1-\alpha-(1-\alpha)|}{\alpha} = 0, \tag{4.57}$$

which is expected since the optically-thick line force goes to infinity as the density goes to zero.

For our model B star, we calculate $\Sigma_0 \approx 0.725$ and use the polar critical radius $r_{c,\theta=0} \approx 1.0367$ to obtain $\dot{m}_{\rm dip} \approx 9.56 \times 10^{-8}\,{\rm g/cm^2}$. The resulting estimated mass-flux in the general case, $\Sigma_0 \dot{m}_{\rm dip} \approx 6.90 \times 10^{-8}\,{\rm g/cm^2}$ compares well with the model-calculated $\dot{m}_{\theta=0} \approx 6.96 \times 10^{-8}\,{\rm g/cm^2}$. Similarly, for the O star, we calculate $\Sigma_0 \approx 0.974$, $\dot{m}_{\rm dip} \approx 1.98 \times 10^{-5}\,{\rm g/cm^2}$ (with $r_{c,\theta=0} \approx 1.054\ R_p$), and $\Sigma_0 \dot{m}_{\rm dip} \approx 1.93 \times 10^{-5}\,{\rm g/cm^2}$. The model polar mass-flux is $\dot{m}_{\theta=0} \approx 2 \times 10^{-5}\,{\rm g/cm^2}$.

Simplifying Equation 4.55 with $r_{c,\theta=0} - R_p \ll R_p$, $\theta_c \approx \theta_m$, and $\psi_c \approx \psi_m = \mu_B$ yields the scaling

$$\dot{m}_* \approx \mu_B \Sigma_0 \dot{m}_{\rm dip}(r_c). \tag{4.58}$$

We can correct for approximations made in deriving this equation (Appendix E) by using the model-calculated $\dot{m}_{{\rm gen},\theta=0}$ instead of $\Sigma_0 \dot{m}_{\rm dip}$:

$$\dot{m}_* \sim \mu_B \dot{m}_{{\rm gen},\theta=0}. \tag{4.59}$$

We use $\dot{m}_{{\rm gen},\theta=0}$ rather than $\Sigma_0 \dot{m}_{\theta=0}$ because the critical point location is different between the two cases. As in the optically-thick case, our model magnetosphere shows excellent agreement for this OD04-type scaling (Figure 4.1).

4.6.3 With rotation in the optically-thick limit

The previous scaling results, however, depend on zero rotation. There are two ways rotation changes the above scaling: μ_B and the location of r_c (c.f. Section 4.7). For an oblate star, the surface normal unit vector is:

$$\hat{n} = \frac{\hat{r} - R_*'/R_*\ \hat{\theta}}{\sqrt{1 + (R_*'/R_*)^2}} \tag{4.60}$$

where $R_*' \equiv \partial R_*/\partial\theta$. Using Equation 4.40b, the resulting μ_B is

$$\mu_B = \frac{2\cos\theta_m - \sin\theta_m R_*'/R_*}{\sqrt{(1+3\cos^2\theta_m)(1+(R_*'/R_*)^2)}}, \tag{4.61}$$

where, from taking the derivative of Equation 4.9,

$$\frac{1}{R_p}\frac{\partial R_*}{\partial \theta} = \frac{\cot\theta \sin\left\{\frac{1}{3}[\pi + \arccos(\omega \sin\theta)]\right\}}{\sqrt{1-\omega^2\sin^2\theta}} - \frac{3\cot\theta\csc\theta\cos\left\{\frac{1}{3}[\pi + \arccos(\omega\sin\theta)]\right\}}{\omega}. \tag{4.62}$$

For a non-rotating star, $R'_* = 0$ and μ_B is identical to OD04. μ_B is affected most at the middle colatitudes, where the stellar surface normal tilts the farthest from the radial direction (Figure 4.2).

However, even though μ_B gets smaller with increased rotation at the middle colatitudes, the surface mass-flux does not decrease in the same manner. Instead, the rotation of the star boosts the mass-flux above this naive scaling, and, for sufficiently high rotation, actually causes the flux to *increase* as one moves towards the middle colatitudes. This is

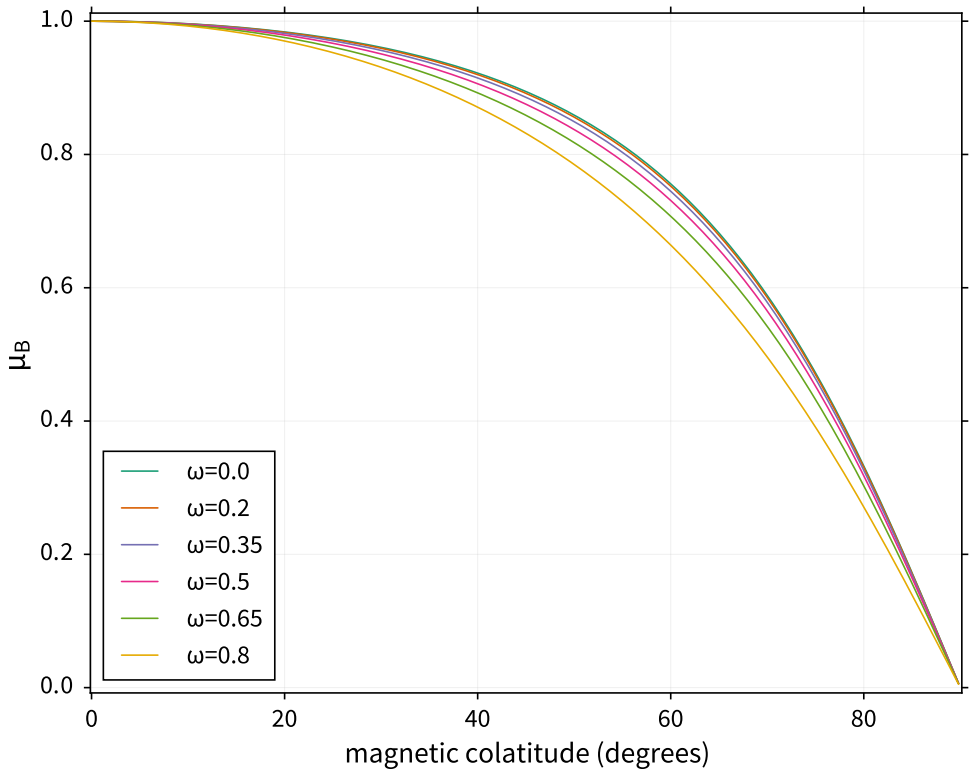


Figure 4.2 : Effect of rotation on μ_B, the dot product between the surface normal unit vector and the surface magnetic field unit vector (Equation 4.61).

chiefly due to the decreased $g_{\mathrm{eff},s}$ as the centrifugal acceleration increases.

We can derive a simple scaling for how $\dot{m}_*$ depends on rotation. We follow the same procedure as in Section 4.6.1, but this time we keep the rotation. First, we rewrite the effective gravity (Equation 4.13) using our dipole magnetosphere parameterizations on θ (Appendix D), specifically $\bar{r} = r\sin\theta = R_p L \sin^3\theta$ and $\bar{\psi}/\psi = 3\sin\theta/2$:

$$\begin{aligned} g_{\mathrm{eff},s} &= -(1-\Gamma_{\mathrm{el}})\frac{GM_*\psi}{r^2}\left(1 - \frac{8\omega^2}{27R_p^3}\frac{\bar{r}r^2\bar{\psi}}{\psi}\right) \\ &= -(1-\Gamma_{\mathrm{el}})\frac{GM_*\psi}{r^2}\left(1 - \frac{4}{9}\omega^2 L^3 \sin^8\theta\right) \\ &= -(1-\Gamma_{\mathrm{el}})\frac{GM_*\psi}{r^2}\aleph, \end{aligned} \tag{4.63}$$

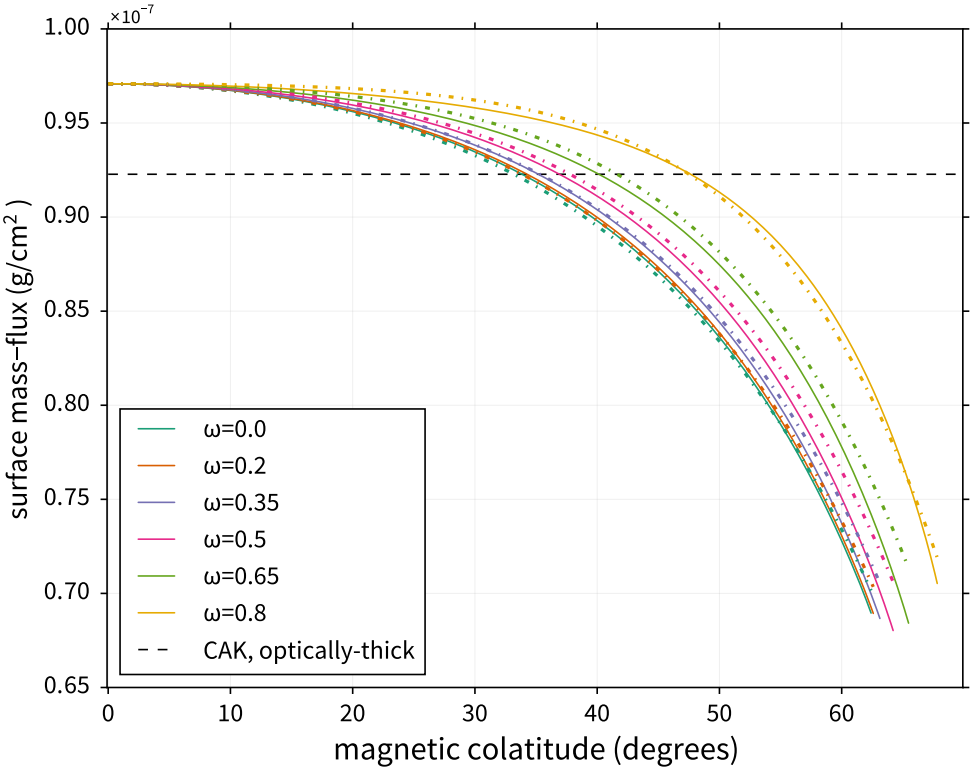


Figure 4.3 : Surface mass-flux ($\dot{m}_*$; solid line) as a function of line surface colatitude for an optically-thick line force and a B-type star (Table 4.1). $\dot{m}_*$ is compared to the rotation scaling (dot-dashed line) derived in Equation 4.70 and the CAK mass-flux (Equation 4.45; dashed line)

where we have defined a rotation effect parameter

$$\aleph \equiv 1 - \frac{8\omega^2}{27R_p^3}\frac{\bar{r}r^2\bar{\psi}}{\psi} \tag{4.64a}$$

$$= 1 - \frac{4}{9}\omega^2 L^3 \sin^8\theta, \tag{4.64b}$$

where the first line is the general definition and the second is specifically for an aligned dipole. We note that for zero rotation, $L = 1/\sin^2\theta_m$, but the stellar oblateness due to rotation means that this equation no longer applies. Instead, we combine Equation 4.9 and Equation 4.41 into

$$L = \frac{3}{\omega \sin^3\theta_m}\cos\left[\frac{\pi + \cos^{-1}(\omega\sin\theta_m)}{3}\right], \tag{4.65}$$

and get an aligned dipole rotation parameter

$$\aleph \equiv 1 - \frac{12\cos^3\left[\frac{1}{3}(\pi + \cos^{-1}(\omega\sin\theta_m))\right]\sin^8\theta}{\omega\sin^9\theta_m}. \tag{4.66}$$

The above analysis simply multiplies each instance of $g_{\mathrm{eff},s}$ in Equation 4.49 by a factor $\aleph$, and yields a mass-flux estimate

$$\dot{m}_* \approx \dot{m}_{\mathrm{dip}}(r_c)\left(\frac{R_p}{R_*}\right)^3 \aleph_c^{1-1/\alpha}\psi_c\sqrt{\frac{1+3\cos^2\theta_m}{1+3\cos^2\theta_c}}, \tag{4.67}$$

where $\aleph_c$, ψ_c and θ_c are evaluated at the critical point and we take into account stellar oblation due to rotation. Comparing the above approximation against numerical results, we find an error of only $\approx 3.5\%$ (6%) error in the middle colatitudes for a B-type (O-type) star at $\omega = 0.8$. This increase in error relative to the non-rotating case comes from neglecting Φ, which is larger at faster rotation rates.

Since the critical radius is no longer constant with colatitude in the rotating cases, we will need to know the critical radius for each magnetic footprint (Section 4.7) in order to get precise estimates. Interestingly, despite this dependence on r_c, we can still take OD04-type approximations to get a reasonable mass-flux estimate for different rotation rates

independent of r_c! We take $\psi_c(R_p/R_*)^3 \approx \mu_B$ (μ_B given in Equation 4.61) and $\theta_c \approx \theta_m$ such that $\aleph$ is evaluated at the stellar surface:

$$\aleph_m \equiv 1 - \frac{12\cos^3\left[\frac{1}{3}(\pi + \cos^{-1}(\omega \sin\theta_m))\right]}{\omega \sin\theta_m}. \tag{4.68}$$

The resulting scaling relation is then

$$\dot{m}_* \approx \dot{m}_{\rm dip}(r_{c,\theta=0})\mu_B \aleph_m^{1-1/\alpha}. \tag{4.69}$$

As before, we correct for neglecting Φ by using the model-calculated $\dot{m}_{\theta=0}$ instead of the approximation $\dot{m}_{\rm dip}$:

$$\dot{m}_* \sim \dot{m}_{\theta=0}\mu_B \aleph_m^{1-1/\alpha}. \tag{4.70}$$

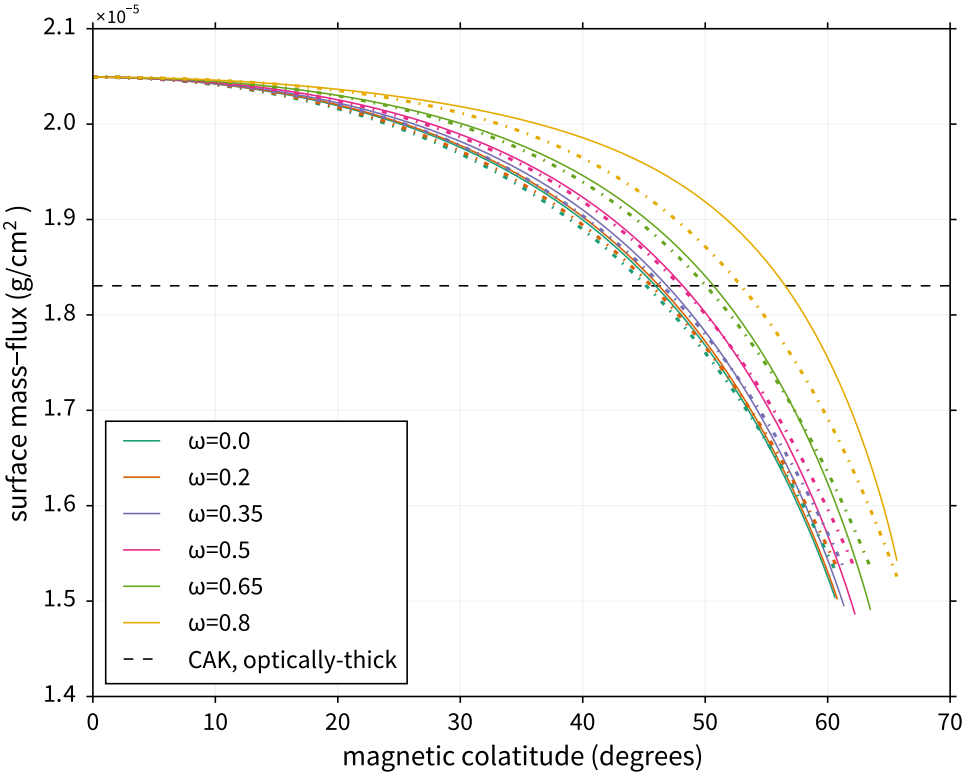


Figure 4.4 : Surface mass-flux ($\dot{m}_*$; solid line) as a function of line surface colatitude for an optically-thick line force and an O-type star. $\dot{m}_*$ is compared to the rotation scaling (dot-dashed line) derived in Equation 4.70 and the CAK mass-flux (dashed line).

Figure 4.3 and Figure 4.4 show the fortunate agreement of Equation 4.70 with our model calculations, despite the questionable approximations. Again, we get larger differences between model and scaling for the faster rotation rates.

4.6.4 With rotation in the general case

Finally, we combine the effects of rotation and the OTC parameter. With a similar derivation as the previous sections, we obtain

$$\dot{m}_* \approx \dot{m}_{\rm dip}(r_c)\Sigma_{\rm rot}\left(\frac{R_p}{R_*}\right)^3 \aleph^{1-1/\alpha}\psi_c\sqrt{\frac{1+3\cos^2\theta_m}{1+3\cos^2\theta_c}}, \tag{4.71}$$

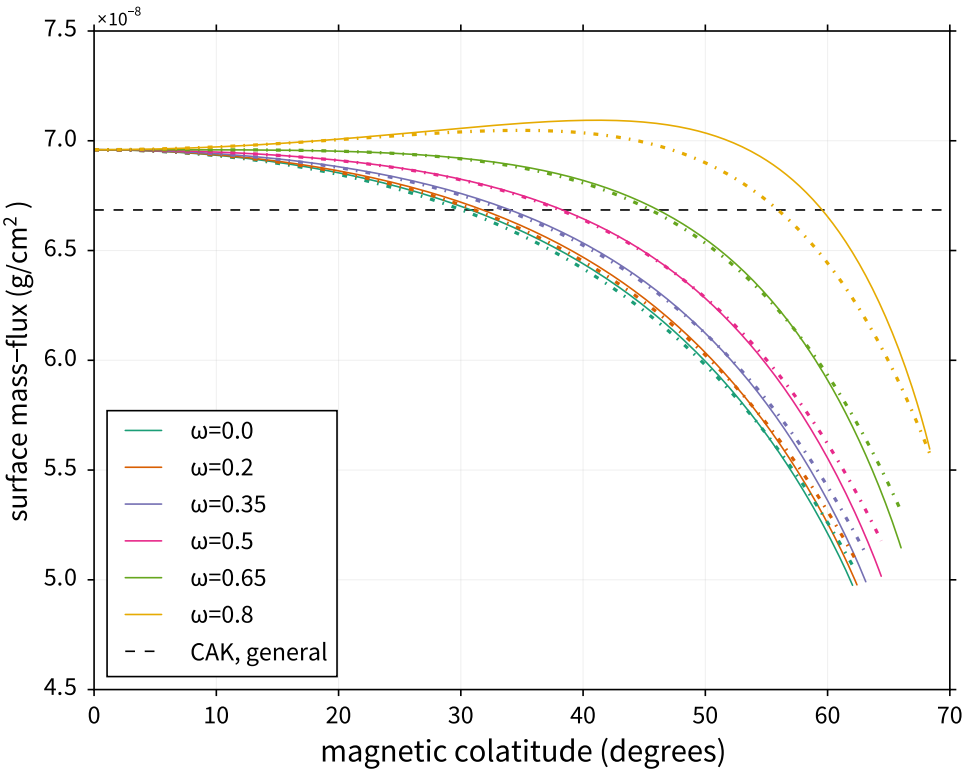


Figure 4.5 : Surface mass-flux ($\dot{m}_*$; solid line) as a function of line surface colatitude for a general line force and a B-type star (Table 4.1). $\dot{m}_*$ is compared to the rotation scaling (dot-dashed line) derived in Equation 4.74 and $\Sigma_0\dot{m}_{\rm CAK}$, the corrected CAK mass-flux (dashed line).

where, again, $\dot{m}_{\rm dip}$ is the base mass-flux from Equation 4.51. Unlike before, however, the OTC parameter now has a dependency on rotation:

$$\Sigma_{\rm rot} = \frac{\left|1 - \alpha - \left[\frac{1-\chi_0^{1/\alpha-1}\aleph^{1/\alpha-1}}{1-\chi_0^{1/\alpha}\aleph^{1/\alpha}}\right]\right|}{\alpha} \tag{4.72}$$

At zero rotation, $\aleph = 1$ and $\Sigma_{\rm rot}$ reduces to Σ_0. With rotation, $\chi_0\aleph$ decreases and $\Sigma_{\rm rot}$ moves towards 1. Physically, this occurs because the rotation-boosted mass-flux in the wind further reduces the error from assuming an optically-thick $g_{{\rm rad},s}$.

In more massive stars, rotation has less of an effect on the OTC parameter. This is because the error from assuming an optically-thick wind is already small, so increasing the density in the wind does not have a relatively large effect.

We can simplify the scaling relation using OD04-type approximations:

$$\dot{m}_* \approx \mu_B \dot{m}_{\rm dip}(r_{c,\theta=0})\Sigma_{\rm rot,m}\aleph_m^{1-1/\alpha}, \tag{4.73}$$

where $\Sigma_{\rm rot,m}$ is calculated at the stellar surface (i.e. Equation 4.72 with $\aleph_m$ instead of $\aleph$). We correct this for approximations made in Appendix E by using the model-calculated $\dot{m}_{{\rm gen},\theta=0}$ instead of $\Sigma_0\dot{m}_{\rm dip}$:

$$\dot{m}_* \sim \frac{\Sigma_{\rm rot,m}}{\Sigma_0}\mu_B\aleph_m^{1-1/\alpha}\dot{m}_{{\rm gen},\theta=0}. \tag{4.74}$$

Figure 4.5 and Figure 4.6 show the good agreement of our scaling (Equation 4.74) with our model calculations for both the B- and O- type stars. Again, we get larger differences between the model and the scaling relation at faster rotation rates. At $\omega = 0.8$, we get about a 3% difference for the B star and 4% for the O star.

4.7 Critical point locations

The critical point location, r_c, depends not only on the boundary condition, but also on the inherent properties of the magnetosphere and wind. In CAK theory, this location

sets the value of the surface mass-flux, so anything that moves this point influences the amount of material being accelerated off the stellar surface. There has been some discussion in the literature about the physicality of the CAK critical point (e.g. Lamers & Cassinelli 1999; Lucy 2007) and its validity in setting the critical mass-flux. These authors prefer using the sonic point to set the critical mass-flux (e.g. in the models of Vink et al. 2000). For now, we defer discussion of this issue to future studies.

In the point-star zero sound-speed limit, CAK found that the entire wind is critical; this degeneracy means that the critical radius (r_c) is ill-defined for this case. Including the small sound-speed term barely breaks this degeneracy, but the finite-disk correction allows

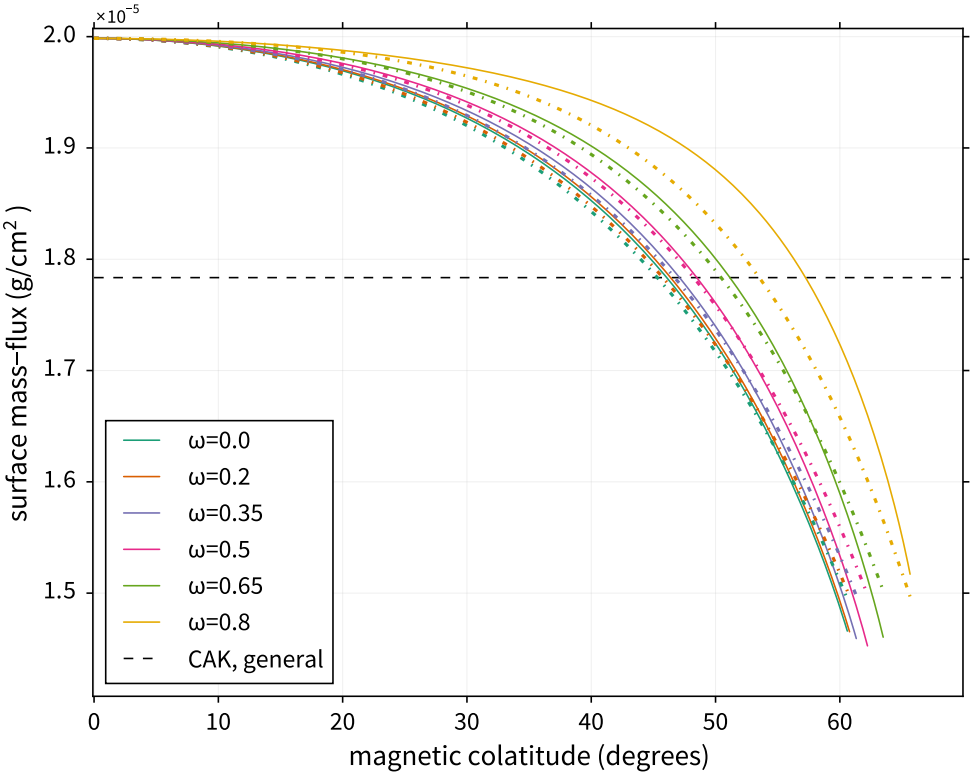


Figure 4.6 : Surface mass-flux ($\dot{m}_*$; solid line) as a function of line surface colatitude for a general line force and an O-type star. $\dot{m}_*$ is compared to the rotation scaling (dot-dashed line) derived in Equation 4.74 and $\Sigma_0 \dot{m}_{\rm CAK}$, the optically-thin corrected CAK mass-flux (dashed line).

r_c to be well-defined by allowing the critical velocity, its derivative, and the mass-loss rate to vary with radius (Kudritzki et al. 1989; Madura et al. 2007). This results in only one radius which satisfies the critical and boundary conditions simultaneously.

Interestingly, dipole divergence also breaks this degeneracy! Although it does not change the critical velocity and its derivative (Equation 4.46), the faster-than-spherical expansion induces a mass-flux dependency on the critical radius (Equation 4.51). There is then only one radius which allows a self-consistent critical mass-flux. This critical location is close to the star ($r_c \approx 1.033 R_p$ for the B star; $\approx 1.054 R_p$ for the O star), much like

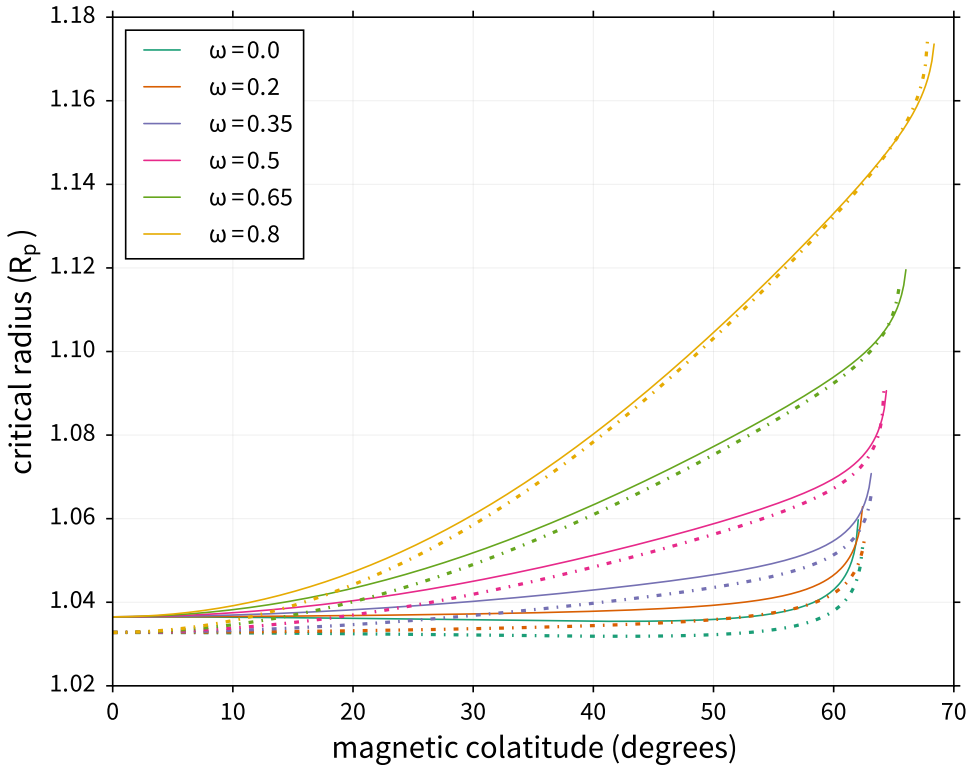


Figure 4.7 : Critical point location r_c in the general (solid) and optically-thick (dot-dashed) cases as a function of critical rotation fraction ω and dipole line surface colatitude θ_m for a B-type star. In general, both rotation and the optically-thin correction move the critical point away from the center of the star, though the rotation effect is caused by the increased stellar radius of an oblate star (Figure 4.8). Similar results for the O star not shown.

the finite-disk corrected critical radius ($\approx 1.02R_p$ in Pauldrach et al. (1986) with different stellar properties for their O star).

Next, we study two factors which influence the critical location, the optically-thin correction and stellar rotation. Both these effects work in the same manner, causing the smaller root of Equation 4.22 and, equivalently, the wind acceleration in the subcritical region to decrease. Even though these factors also lower the critical velocity (Equation 4.31), they have different effects on r_c. The optically-thin wind takes a longer distance to accelerate from the boundary to the critical point, which pushes r_c out. However, rotation lowers the critical velocity sufficiently that the wind is able to accelerate over a slightly shorter distance, pulling r_c in relative to the stellar surface. Stellar oblation, though, will push the critical radius out relative to the center of the star.

Our results for a dipole magnetosphere (Figure 4.7/Figure 4.8) show how r_c moves out due to rotation and the OTC parameter. We note that the increase in critical radius due to rotation is almost entirely caused by the stellar oblation. In fact, we see that the radial distance of the critical point from the stellar surface is nearly constant with colatitude and rotation. For more massive stars, the error from the optically-thick assumption is reduced (see Section 4.6.2 for discussion), so there is a smaller difference in r_c between the general and optically-thick cases than for later-type stars.

Finally, we note that the critical radius does not exist for every field line. In Figure 4.7, we see a clear tendency for the footprint colatitude to have a limit. Another interesting aspect is that different rotation rates have different θ_m limits, though as we will see, this limit may actually be determined by the field line shell parameter (L). The starting footprint locations of our six model magnetospheres (one for each rotation fraction) are identical, but the different stellar surfaces resulting from rotation (c.f. Equation 4.60) yield different L values for a single θ_m. Faster rotation results in a greater L for a given surface

colatitude. When we evaluate $L_{\rm min}$ corresponding to the maximum θ_m for each rotation rate, we get similar values: $L_{\rm min} \approx 1.27 - 1.3 R_*$ for a B star, $L_{\rm min} \approx 1.32 - 1.33 R_*$ for a O star, with slight differences between the general and optically-thick cases. We stress that these limits are only approximate due to the division of our model stellar surface into 500 discrete magnetic footprints.

It is uncertain exactly why this limit, if there is one, exists. For lines close to this limit, there do exist possible critical points, but none satisify the boundary condition. This occurs because the magnetic tension due to line tilt (represented by ψ) neuters the wind acceleration so that the flow cannot pass through the critical point for the given boundary condition. For lines much closer to the equator, there are no possible critical points for any boundary condition since $\Phi^2 < 0$ (Section 4.3).

4.8 Velocity structure

Here, we focus on the wind velocity as it is accelerated along a dipole field line and qualify its behavior.

4.8.1 Is there a beta-velocity law?

In the CAK zero sound-speed limit, the radial velocity structure is found to be $v = v_\infty (1 - R_*/r)^\beta$, with $\beta = 1/2$ for this specific case. With the finite-disk correction, the velocity can still be well-represented by this beta-velocity law (Pauldrach et al. 1986), which greatly simplifies analytic considerations of the finite-disk correction factor by allowing the factor to be represented as an explicit spatial function (e.g. Madura et al. 2007). However, for a magnetosphere, there are two problems with assuming a beta-velocity law: the terminal velocity and the effect of rotation.

The faster-than-spherical divergence of the field will lead to higher terminal velocities

(Owocki & ud-Doula 2004) than for the spherically-diverging case. With the escape velocity defined as $v_{\rm esc}^2 = 2(1-\Gamma_{\rm el})GM_*/R_p$, we calculate the polar $v_\infty \approx 1.97\ v_{\rm esc}$ ($5.9\ v_{\rm esc}$) for our B-type (O-type) OTC wind. We note that observed terminal velocities of magnetic stars are unlikely to ever reach these limits since the faster-than-spherical divergence of the wind will not continue indefinitely through space. For comparison, the mCAK terminal velocity, which includes the finite-disk correction, is 3000 km s^{-1} $= 4.23\ v_{\rm esc}$ for our O star (ud-Doula et al. 2014); however, this is based on the optically-thick $g_{{\rm rad},s}$ which does not properly reduce to the optically-thin limit at low densities. Observations of non-magnetic stars give lower terminal values $v_\infty \approx 1.4 v_{\rm esc}$ for B stars and $v_\infty \approx 2-3\ v_{\rm esc}$ for O stars (Kudritzki & Puls 2000).

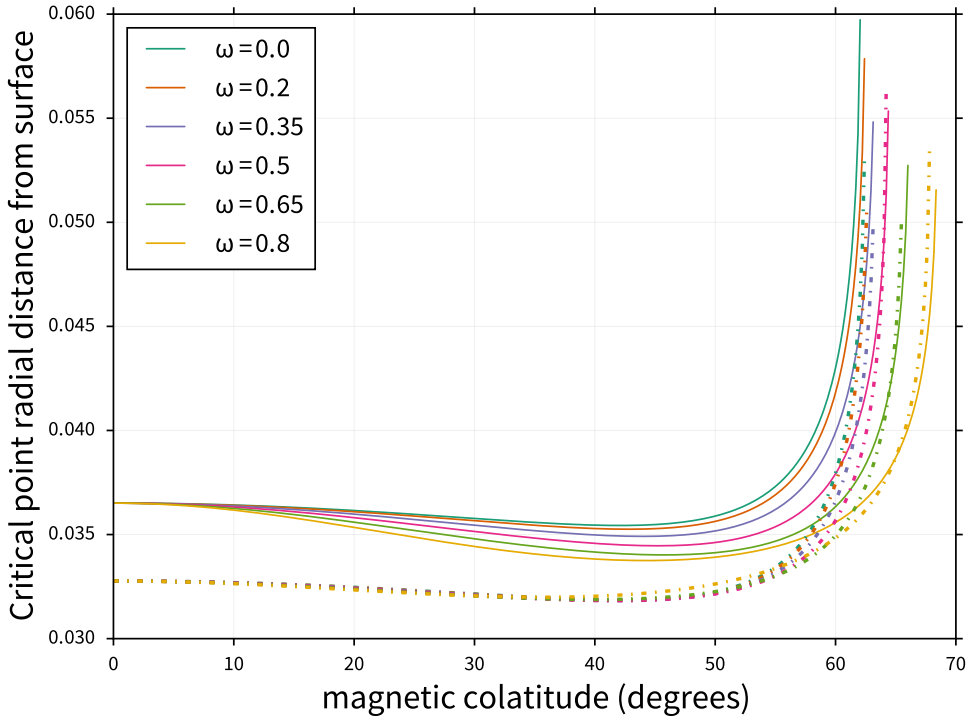


Figure 4.8 : Critical point location realative to the stellar surface ($r_c - R_*$) in the general (solid) and optically-thick (dot-dashed) cases as a function of critical rotation fraction ω and dipole line surface colatitude θ_m for a B-type star. The optically-thin correction tends to move the critical radius away from the star, but rotation moves the critical radius slightly closer to the star. Similar results for the O star not shown.

Of course, closed field lines cannot have terminal velocities. We find that that there is no easily-defined global beta-velocity law for the magnetosphere. Instead, each line has an independent velocity structure which depends on its own geometry. To quantify this, we use nonlinear least squares to fit a beta-velocity law to the numerically-calculated velocity structure of each line. For a non-rotating magnetic dipole, the best-fit value of both v_∞ and β varies throughout the magnetosphere (Figure 4.9). As one moves toward more polar colatitudes, this best-fit v_∞ approaches the asymptotic limit of the straight-line dipole terminal velocity. The magnetic field geometry also affects how quickly the wind accelerates; higher tilt relative to the surface reduces both the "terminal velocity" and how long it takes the wind to reach that limit (as represented by a decreasing β).

Interestingly, the best-fit β values are quite large for our model O star ($\beta > 2$) and not within the usual non-magnetic range $0.7 \lesssim \beta \lesssim 1$ (Kudritzki et al. 1989). The model B star shows similar behavior, though the β range fits better with non-magnetic values. Both

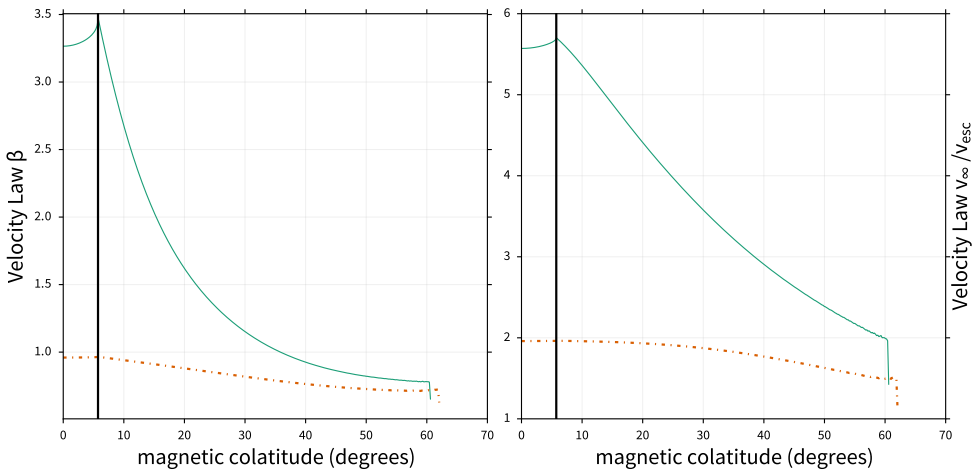


Figure 4.9 : Best-fit velocity-law betas (left) and corresponding terminal velocities (right) for our non-rotating O star (solid) and B star (dot-dashed) models. Fit parameters were determined using nonlinear least-squares fitting. The solid black line represents the surface colatitude for $L = 100\ R_p$, our truncation radius; results from more polar colatitudes should be ignored since those model lines were truncated well before reaching an apex.

cases imply that the field-line tilt has a large effect on throttling the wind acceleration even as the dipole divergence works to boost it.

We note that that finite-disk effect will change these velocity results, but it may not have a large effect. Compared to the point-star CAK model, the finite-disk effect leads to a higher terminal velocity because less mass is driven off the star and this lower-density wind sees more of the stellar surface as it accelerates out. However, the mCAK models use the optically-thick version of $g_{\mathrm{rad},s}$, which artifically boosts this low-density acceleration and leads to higher terminal velocities. We will implement the finite-disk correction in Paper II and characterize its effect on the velocity structure there.

The second issue with a global magnetospheric beta-velocity law is the acceleration from rigid-body rotation. Beyond a certain point, centrifugal acceleration will exceed the inward gravitational pull and cause the wind to accelerate; there is no asymptotic limit. This breaks the beta-velocity law assumption and makes it difficult to characterize the velocity structure with a general equation (Figure 4.10).

We note that for the more equatorial lines in our model, the wind did not actually accelerate all the way from the critical point to the line apex. Instead, a kink solution occurs since the equation of motion (Equation 4.22) stops admitting positive roots for the velocity derivative at some location while still allowing the negative roots (c.f. Cranmer & Owocki 1996; Madura et al. 2007). In order to fit a beta-velocity law to these lines, we ignored the deceleration after the kink and only fit the portion of the line from the critical point to the deceleration point. This implicity assumes that shocks along each line will prevent the wind from ever reaching a kink, so these fits will represent the behavior of the wind velocity up to the shock.

4.8.2 Closure radius

There is always a struggle between the wind and the magnetic field within magnetospheres. As quantified by η_* (Equation 4.1), the field dominates the wind if its energy is larger than the wind kinetic energy. On the other hand, the wind will escape if its velocity exceeds the local Alfvén velocity $v_A = B/\sqrt{4\pi\rho}$. We can thus understand the Alfvén radius (R_A), the typical length scale of the magnetosphere, as the point where $v = v_A$. Additionally, we can approximate the maximum extent of closed loops in the magnetosphere, the closure radius (R_c), as the $L_{\rm shell}$ for which $v_{\rm apex} = v_A$. This will only be a lower bound in the context of this model ($\eta_* \to \infty$) since the shocks produced by colliding wind flows will

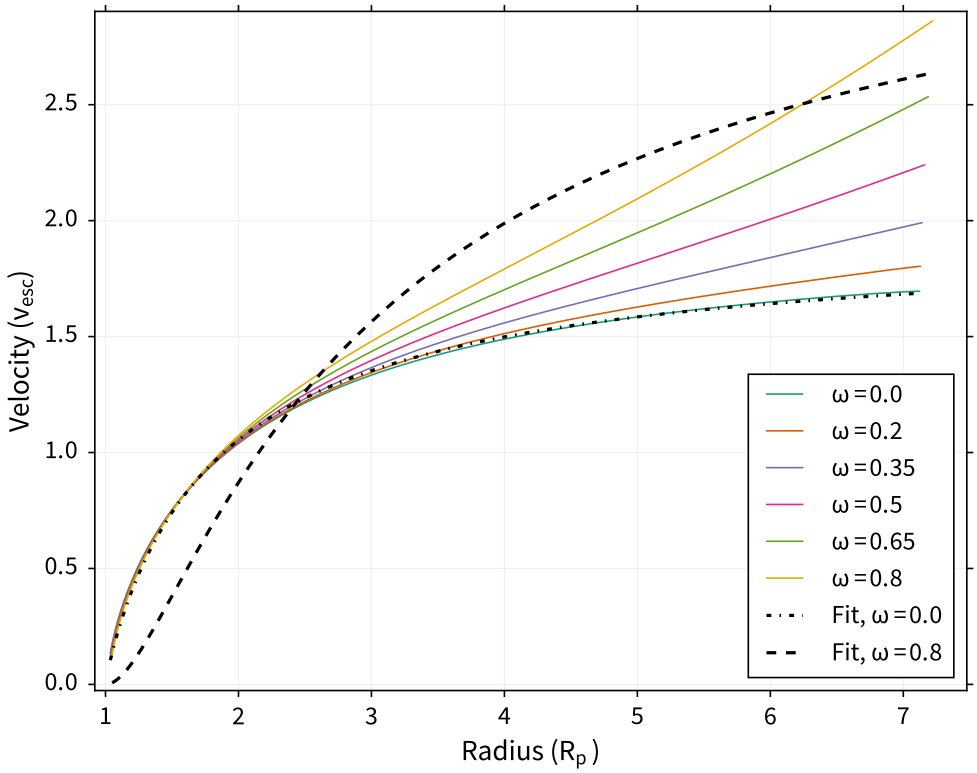


Figure 4.10 : Velocity in terms of the stellar escape velocity along one model B star magnetic dipole line ($\theta_m = 22°$; $L = 7.12 - 7.22\ R_p$) for several stellar rotation fractions ω. The best-fit beta-laws from nonlinear least-squares fitting for zero rotation (dot-dashed) and high rotation (dashed) are also shown.

not allow the wind to fully accelerate all the way to the line apex.

We can compare this to the MHD-derived closure radius scaling (ud-Doula & Owocki 2002)

$$R_c \approx R_* + 0.7(R_A - R_*), \tag{4.75}$$

with the dipole Alfvén radius given by

$$\frac{R_A}{R_p} \approx 0.3 + \eta_*^{1/4} \tag{4.76}$$

characterizing the maximum radius at which the magnetic field still dominates the wind.

For our model stars, we assume the same non-magnetic v_∞ and $\dot{M}$ as the previous subsection. We take the σ Ori E-like value $B_{\rm eq} = 5500\,{\rm G}$ for our B star and take $B_{\rm eq} = 1857.5\,{\rm G}$ such that $\eta_* = 100$ for our O star. This results in a MHD-predicted closure radius $R_c = 18.4R_p$ ($2.72R_p$) for the B star (O star).

Not surprisingly, we obtain larger closure radii than MHD predictions (Table 4.2). This is due to our rigid-field assumption; in reality, the wind will stretch out the polar field lines radially (ud-Doula & Owocki 2002) and accelerate more rapidly when ψ moves towards unity. The resulting faster velocities would move R_c towards the MHD-approximated closure radius.

We also see a clear trend of rotation boosting both the apex wind and Alfvén velocities. The Alfvén velocities increase since, by the conservation of mass, the faster

Table 4.2 : Estimated closure radii (in units R_p) for the model B star centrifugal magnetosphere and O star dynamical magnetosphere at different rotation rates. The MHD scaling estimate (Equation 4.75) is also included for comparison.

Type	$\omega = 0.0$	0.2	0.35	0.5	0.65	0.8	MHD
B	74.1	54.9	48.0	44.2	42.4	39.2	18.4
O	5.02	4.96	4.91	4.87	4.78	4.65	2.72

wind velocities result in lower densities. The overall effect is to produce smaller closure radii at faster rotation rates.

4.9 Global mass loss

Here, we study the effect of the dipole field on the global mass-loss rate for different rotation rates. We can find the global mass-loss rate, $\dot{M}_{\rm global}$, by integrating the mass-flux over the stellar surface:

$$\dot{M}_{\rm global} = \int \dot{m}_r \; dA = 2\pi \int R_*^2 \mu_B \dot{m}_* d\mu_* \; , \tag{4.77}$$

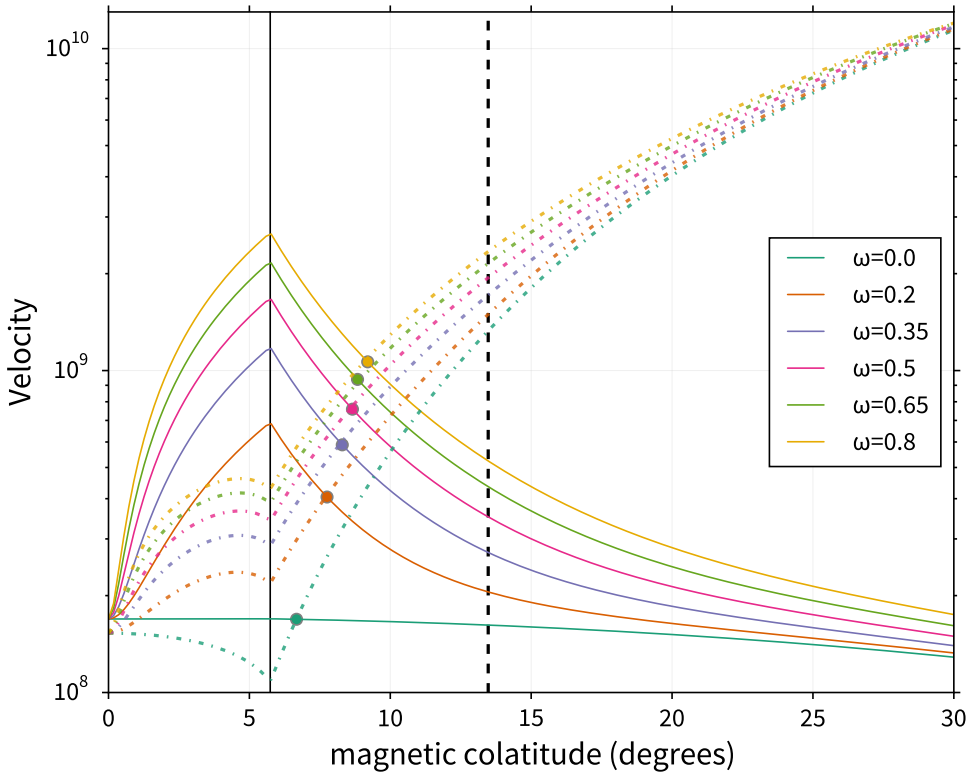


Figure 4.11 : Comparison of wind velocity (solid) vs. Alfvén velocity (dot-dashed) at apex of each magnetic field line for our optically-thin corrected B star model. Colored dots indicate approximation of closure colatitude for each rotation rate. The dashed line indicates the non-rotating MHD-predicted closure colatitude. The solid black line represents the surface colatitude for $L = 100\ R_p$, our truncation radius; results from more polar colatitudes should be ignored since those model lines were truncated well before reaching an apex.

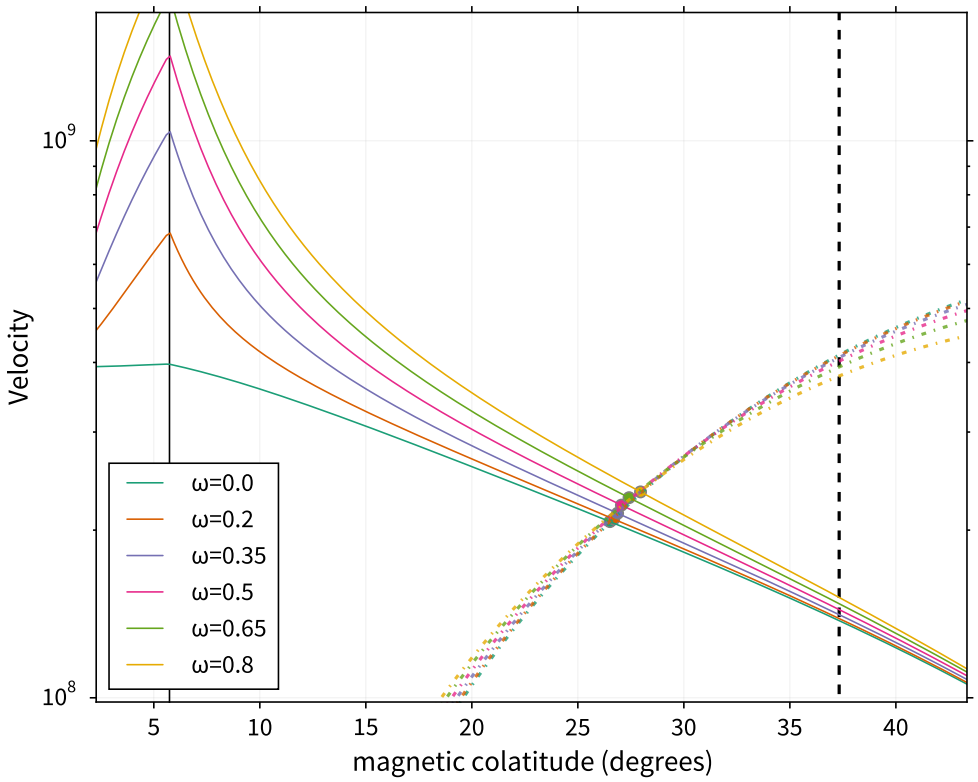


Figure 4.12 : Same as Figure 4.11, except for our O star model.

Table 4.3 : Mass-loss rates (in units of $10^{-9}\ M_\odot/\,\mathrm{yr}$) for a B-type star ($\eta_* = 4.28 \times 10^5$) as calculated from our model, assuming infinite magnetic confinement where applicable. "No B" indicates a CAK-type mass-loss rate calculated from a non-rotating radial flow with spherical divergence. The other mass-loss rates are calculated from a dipole magnetosphere with the given rotation fraction ω. "Optically-Thick" indicates the mass-loss calculated from using the optically-thick $g_{\mathrm{rad},s}$; the rest use the general $g_{\mathrm{rad},s}$. "Open" is the mass-loss into open field lines ($L > R_c$), R_c given by the MHD-estimated value (Equation 4.75). "Disk" is the mass-loss into field lines with a centrifugally supported disk ($R_K < L < R_c$). Numbers in parentheses next to a mass-loss rate represent the ratio of that particular rate to the "General" mass-loss at its rotation fraction ω.

	No B	$\omega = 0.0$	0.2	0.35	0.5	0.65	0.8
Optically-Thick	1.81	1.00	1.01	1.02	1.04	1.08	1.12
General	1.29	0.718	0.724	0.738	0.762	0.797	0.845
Open	...	0.036(0.05)	0.036(0.05)	0.036(0.049)	0.037(0.049)	0.037 (0.046)	0.037(0.044)
Disk	...	...	0.21(0.29)	0.33 (0.45)	0.46 (0.6)	0.61 (0.77)	0.8 (0.95)
Effective	...	0.036 (0.05)	0.24(0.33)	0.37(0.5)	0.49 (0.64)	0.64 (0.8)	0.84 (0.99)

with $\mu_* \equiv \cos\theta_m$. Since not every field line has a calculated critical mass-loss rate, we will assume that the various scaling relations for $\dot{m}_*$ derived in Section 4.6 hold for the entire stellar surface. For the non-magnetic case, we take the mass-flux for a straight line with spherical divergence ($\dot{m}_{B=0}$) and integrate over the stellar surface:

$$\dot{M}_{\rm global,B=0} = \dot{m}_{B=0} \int dA \ = 4\pi R_p^2 \dot{m}_{B=0}. \tag{4.78}$$

We stress that the mass-loss rate in Equation 4.77 is not a "true" mass-loss rate; the plasma flowing along closed field lines does not easily escape the magnetosphere (ud-Doula et al. 2008; Townsend et al. 2013). In our model, our assumption of infinite magnetic confinement means that, technically, none of the stellar wind escapes the magnetic field. The amount of mass that escapes depends on the closure radius (Equation 4.75), which itself depends on the stellar magnetic field strength. Despite this, however, $\dot{M}_{\rm global}$ is still an useful value to calculate, as it will give a better estimate for the wind magnetic confinement parameter (ud-Doula & Owocki 2002), depending on rotation.

Additionally, we calculate an "disk" mass-loss rate into a centrifugally-supported disk. Such disks are created because maintaining rigid-body rotation away from the star eventually leads to a balance between the gravitational and centrifugal forces at the Kepler radius (e.g. ud-Doula et al. 2008)

$$R_K = \frac{GM_*}{v_\phi^2} = \omega^{-2/3} R_p. \tag{4.79}$$

For lines inside the Kepler radius, the lack of centrifugal support results in a pattern of outflow and infall that leads to a long-term average mass-loss of zero (ud-Doula & Owocki 2002). Similarly, we can calculate a "true" mass-loss rate using the MHD-derived closure radius (Equation 4.75) to approximate which lines are considered open. Combining the "disk" and "true" rates can give us an "effective" mass-loss; the star loses mass if it will

not return to the surface, either because it settles in a magnetospheric disk or escapes through open field lines.

The results for our model magnetospheres are presented in Table 4.3 (B star) and Table 4.4 (O star). For zero rotation, the ratio between the general and optically-thick $\dot{M}_{\rm global}$ is well explained by the OTC parameter, Σ_0. For our B star, we calculate a ratio $7.18 \times 10^{-10}/1 \times 10^{-9} \approx 0.718$, which compares well to our approximated $\Sigma_0 \approx 0.725$. For the O star, the model $\dot{M}_{\rm global}$ ratio is $3.67 \times 10^{-6}/3.77 \times 10^{-6} \approx 0.973$, which fits with our $\Sigma_0 \approx 0.974$. Additionally, the ratio between the CAK-type mass-loss rates ("No B" in Table 4.3/Table 4.4) between the general and optically-thick cases can be approximated by Σ_0. The reason why the ratios differ from the actual value of Σ_0 is because r_c is different between the general and optically-thick cases, leading to different base mass-fluxes (Equation 4.51). Nonetheless, multiplying the base mass-loss by the OTC parameter gives an excellent approximation.

The effective mass-loss rates for both our B and O stars imply that most of the plasma falls back to the star at low rotation (about 2/3rds at $\omega = 0.2$) and nearly none of it falls back at high rotation (1% at $\omega = 0.8$). The mass loss is reduced to about 20-65% of the non-magnetic, non-rotating CAK value. This reduction will affect the eventual evolution of these stars; changing the total amount of mass lost over the stellar lifetime will change what the star becomes at the end of its life (Maeder & Meynet 2012), and possibly lead to different supernovae properties than non-magnetic stars (Smartt 2009).

Since we use Gayley (1995)'s $\bar{Q}$ parameterization for the line-acceleration, we must be careful when comparing calculated mass-loss rates with other models which use the more traditional CAK force multipler paradigm (Abbott 1982). Puls et al. (2000) show that using $\bar{Q}$ in $g_{{\rm rad},s}$ instead of the CAK k parameter requires an ansatz that does not hold for $T_* < 35000K$. In our model, assuming a wind temperature equal to the stellar

effective temperature means that B star winds will be below this cutoff; the result is an overestimated mass-loss rate (c.f. Table 2 in Puls et al. 2000) by about a factor of 2. Future studies will need to determine the wind temperature to check the validity of the $T = T_{\rm eff}$ assumption.

Further improvements to our global mass-loss rates will require consideration of the finite-disk effect. We will implement this term in Paper II, but for now we can estimate the finite-disk corrected $\dot{M}_{\rm global}$ by dividing our results by two.

4.10 Summary and future work

In this paper, we presented a critical point analysis of the Arbitrary Rigid-Field Hydrodynamic Equations, which represent a CAK-type wind within an arbitrary, infinitely-strong magnetic field. This differs from the usual CAK wind model by including the proper optically-thin maximum line-force, a rigid-body centrifugal acceleration, and a dipole areal divergence. After finding the general critical point values for the mass-flux, velocity and velocity derivative, we confirmed that they reduced to the proper values for a traditional CAK wind, i.e. a non-rotating, non-optically-thin corrected, radial flow with spherical divergence. These benchmarked general critical point equations were then applied to an aligned magnetic dipole field in order to calculate critical point locations and surface mass-fluxes. By integrating from these critical point locations, the velocity structure within the magnetosphere was quantified and studied. Finally, we obtained global mass-loss rates and found that the dipole field effectively reduces the overall mass-loss to 20-65% of the non-magnetic, non-rotating CAK value.

The key results are summarized as follows:

1. We are able to approximately confirm the Owocki & ud-Doula (2004) scaling for the influence of a magnetic dipole on the surface mass-flux, $\dot{m}_r \approx \mu_B^2 \dot{m}_{\rm CAK}$. While

this scaling does not need much improvement, we provide a more accurate scaling equation (Equation 4.50) and detail which approximations are required to reproduce the OD04 scaling.

2. The effect of a optically-thin corrected line-force can be encapsulated in a OTC parameter, which we call Σ_0 (Equation 4.56). Including this does not have much of an effect for O-type stars, since their increased wind density means that there will be less difference in the corrected and uncorrected line-forces. B-type stars, on the other hand, have their surface mass-flux reduced by approximately 25-30% when the optically-thin correction is taken into account.

3. The effect of rotation can be similarly represented with a rotation-effect parameter, which we call $\aleph$ (Equation 4.64a). The amount of rotational boosting of the mass-flux is found to depend on both the rotational colatitude and the magnetic obliquity angle.

4. The effects of rotation and the optically-thin correction can not be decoupled, however. We find a different OTC parameter in the case of rotation, $\Sigma_{\rm rot}$ (Equation 4.72). Rotation is found to reduce the correction by driving a higher surface mass-flux.

5. The velocity structure within a magnetosphere cannot be described by a global beta-velocity law. However, at least for zero rotation, we can well-fit each line with individual beta-velocity laws. The best-fit v_∞ and β do vary from line to line, however. With rotation, the beta-velocity law assumption breaks down.

6. The global mass-loss rate for a optically-thin corrected line-force can be accurately estimated by multiplying the optically-thick mass-loss by the OTC parameter, Σ. We find "effective" magnetospheric mass-loss rates, in which the plasma does not fall

back to the star, to be approximately 20-65% of the non-magnetic, non-rotating CAK mass-loss rate.

Overall, we have quantified the effect of a magnetic dipole on a massive star wind with an eye towards better understanding of massive star magnetospheres. Next steps include adding the finite-disk correction parameter and quantifying its effect on the magnetospheric mass-loss and velocity (Paper II). Paper III will add colliding wind shocks and the subsequent "cooling" region to each line in order to better quantify the level of X-ray emission coming from each line. This will provide accurate initial conditions for hydrodynamical simulations of centrifugal magnetospheres.

4.11 Acknowledgments

CB acknowledges support from the NASA GSRP Fellowship (NASA Grant NNX11AK70H). CB and RHDT acknowledge support from NASA ATP Grant NNX12AC72G. The authors thank S. Owocki and J. Cassinelli for helpful discussions and comments.

References

Abbott, D. C. 1980, ApJ, 242, 1183

—. 1982, ApJ, 259, 282

Table 4.4 : Same as Table 4.3, except for an O-type star with $\eta_* = 100$. All mass-loss rates are given in 10^{-6} $M_\odot$/yr. Numbers in parentheses next to a mass-loss rate represent the ratio of that particular rate to the "General" mass-loss with the same rotation.

	No B	$\omega = 0.0$	0.2	0.35	0.5	0.65	0.8
Optically-Thick	6.61	3.77	3.79	3.83	3.91	4.01	4.11
General	6.44	3.67	3.70	3.74	3.81	3.91	4.03
Open	...	1.33(0.36)	1.35(0.36)	1.35(0.37)	1.37(0.36)	1.41(0.36)	1.44(0.36)
Disk	...	...	...	0.52 (0.14)	1.12 (0.29)	1.78(0.46)	2.56(0.64)
Effective	...	1.33 (0.36)	1.35 (0.36)	1.87(0.5)	2.49(0.65)	3.19(0.82)	3.99 (0.99)

Babel, J., & Montmerle, T. 1997, A&A, 323, 121

Bjorkman, J. E. 1995, ApJ, 453, 369

Bohlender, D. A., & Monin, D. 2011, AJ, 141, 169

Cassinelli, J. P. 1979, ARA&A, 17, 275

Castor, J. I., Abbott, D. C., & Klein, R. I. 1975, ApJ, 195, 157

Chandra, P., Wade, G. A., Sundqvist, J. O., et al. 2015, MNRAS, 452, 1245

Cranmer, S. R., & Owocki, S. P. 1996, ApJ, 462, 469

Curé, M., & Rial, D. F. 2004, A&A, 428, 545

Donati, J.-F., Howarth, I. D., Jardine, M. M., et al. 2006, MNRAS, 370, 629

Drew, J. E. 1989, ApJS, 71, 267

Eikenberry, S. S., Chojnowski, S. D., Wisniewski, J., et al. 2014, ApJ, 784, L30

Friend, D. B., & Abbott, D. C. 1986, ApJ, 311, 701

Gayley, K. G. 1995, ApJ, 454, 410

Gayley, K. G., & Owocki, S. P. 2000, ApJ, 537, 461

Grunhut, J. H., Rivinius, T., Wade, G. A., et al. 2012, MNRAS, 419, 1610

Howarth, I. D., Walborn, N. R., Lennon, D. J., et al. 2007, MNRAS, 381, 433

Kochukhov, O., Lundin, A., Romanyuk, I., & Kudryavtsev, D. 2011, ApJ, 726, 24

Kudritzki, R. P. 2002, ApJ, 577, 389

Kudritzki, R. P., Pauldrach, A., Puls, J., & Abbott, D. C. 1989, A&A, 219, 205

Kudritzki, R.-P., & Puls, J. 2000, ARA&A, 38, 613

Lamers, H. J. G. L. M., & Cassinelli, J. P. 1999, Introduction to Stellar Winds

Linsky, J. L., Drake, S. A., & Bastian, T. S. 1992, ApJ, 393, 341

Lucy, L. B. 2007, A&A, 474, 701

Madura, T. I., Owocki, S. P., & Feldmeier, A. 2007, ApJ, 660, 687

Maeder, A., & Meynet, G. 2012, Reviews of Modern Physics, 84, 25

Marlborough, J. M., & Zamir, M. 1984, ApJ, 276, 706

Morel, T., Castro, N., Fossati, L., et al. 2015, in IAU Symposium, Vol. 307, IAU Symposium, ed. G. Meynet, C. Georgy, J. Groh, & P. Stee, 342–347

Müller, P. E., & Vink, J. S. 2008, A&A, 492, 493

Nazé, Y., Petit, V., Rinbrand, M., et al. 2014, ApJS, 215, 10

Nazé, Y., Sundqvist, J. O., Fullerton, A. W., et al. 2015, MNRAS, 452, 2641

Noebauer, U. M., & Sim, S. A. 2015, MNRAS, 453, 3120

Owocki, S. P., Castor, J. I., & Rybicki, G. B. 1988, ApJ, 335, 914

Owocki, S. P., & ud-Doula, A. 2004, ApJ, 600, 1004

Pauldrach, A., Puls, J., & Kudritzki, R. P. 1986, A&A, 164, 86

Petit, V., Owocki, S. P., Wade, G. A., et al. 2013, MNRAS, 429, 398

Puls, J., Springmann, U., & Lennon, M. 2000, A&AS, 141, 23

Schure, K. M., Kosenko, D., Kaastra, J. S., Keppens, R., & Vink, J. 2009, A&A, 508, 751

Smartt, S. J. 2009, ARA&A, 47, 63

Sundqvist, J. O., & Owocki, S. P. 2015, MNRAS, 453, 3428

Townsend, R. H. D., & Owocki, S. P. 2005, MNRAS, 357, 251

Townsend, R. H. D., Owocki, S. P., & Ud-Doula, A. 2007, MNRAS, 382, 139

Townsend, R. H. D., Rivinius, T., Rowe, J. F., et al. 2013, ApJ, 769, 33

ud-Doula, A., Owocki, S., Townsend, R., Petit, V., & Cohen, D. 2014, MNRAS, 441, 3600

ud-Doula, A., & Owocki, S. P. 2002, ApJ, 576, 413

ud-Doula, A., Owocki, S. P., & Townsend, R. H. D. 2008, MNRAS, 385, 97

—. 2009, MNRAS, 392, 1022

Vink, J. S., de Koter, A., & Lamers, H. J. G. L. M. 2000, A&A, 362, 295

Wade, G. A., Grunhut, J., Alecian, E., et al. 2014, in IAU Symposium, Vol. 302, IAU Symposium, 265–269

White, R. L., & Chen, W. 1995, in IAU Symposium, Vol. 163, Wolf-Rayet Stars: Binaries; Colliding Winds; Evolution, ed. K. A. van der Hucht & P. M. Williams, 438

A Equation of motion in general case

Starting with the general $g_{\mathrm{rad},s}$ (Equation 4.16), we eliminate the density with the continuity equation (Equation 4.19):

$$g_{\mathrm{rad},s} = \frac{\bar{Q}\Gamma_{\mathrm{el}} G M_*}{1-\alpha}\frac{\psi}{r^2}\frac{|vv'|}{\xi}\left[(1+\xi/|vv'|)^{1-\alpha}-1\right], \tag{A1}$$

where we've defined the eigenvalue $\xi \equiv c\kappa_e \bar{Q}\dot{m}_* \frac{A_*}{A}$. We then get our equation of motion:

$$\begin{aligned} F_{\mathrm{gen}} \equiv\ & vv'\left(1-c_{\mathrm{s}}^2/v^2\right) - g_{\mathrm{eff},s} \\ & - \frac{\bar{Q}\Gamma_{\mathrm{el}} G M_*}{1-\alpha}\frac{\psi}{r^2}\frac{|vv'|}{\xi}\left[(1+\xi/|vv'|)^{1-\alpha}-1\right] = 0, \end{aligned} \tag{A2}$$

where, as in the optically-thick case, we ignore the Parker term $c_{\mathrm{s}}^2\lambda$.

B Critical values for optically-thick wind

From the equation of motion (Equation 4.22), the CAK singularity condition (Equation B4), and the CAK regularity condition (Equation B5), we now solve for the critical values (Δ_c, u_c, and y_c) as a function of critical point location, s_{crit}. For mathematical simplicity, we will define $\Gamma \equiv (A/A_*)^\alpha\ \psi/r^2$ in the derivation, such that the starting equation of motion is

$$F \equiv y(1-1/u^2) - g_{\mathrm{eff},s} - \Delta\Gamma|y|^\alpha. \tag{B3}$$

We continue with evaluating both CAK critical conditions (Equation 4.28, Equation 4.29), remembering that $g_{\mathrm{eff},s}$ and Γ are wholly functions of s:

$$0 = \frac{\partial F}{\partial y} = (1-1/u^2) - \alpha\Delta\Gamma|y|^\alpha/y \tag{B4}$$

and

$$0 = \frac{\partial F}{\partial s} + \frac{y}{c_{\mathrm{s}}^2 u}\frac{\partial F}{\partial u} = -\frac{\partial g_{\mathrm{eff},s}}{\partial s} - \Delta|y|^\alpha\frac{\partial\Gamma}{\partial s} + \frac{2y^2}{c_{\mathrm{s}}^2 u^4}. \tag{B5}$$

Starting from the singularity condition, we get

$$\partial F/\partial y = (1 - 1/u_c^2) - \alpha\Delta_c\Gamma\frac{|y_c|^\alpha}{y_c} = 0, \tag{B6}$$

$$(1 - 1/u_c^2)y_c = \alpha\Delta_c\Gamma|y_c|^\alpha. \tag{B7}$$

Substituting this into Equation 4.26 yields

$$(1 - \alpha)\Delta_c\Gamma|y_c|^\alpha = -g_{\mathrm{eff},s}. \tag{B8}$$

Combining this and Equation B7 gives us

$$y_c(1 - 1/u_c^2) = -\frac{\alpha}{1 - \alpha}\, g_{\mathrm{eff},s} \tag{B9}$$

From the regularity condition, Equation B5, we obtain

$$\left(\frac{y_c}{u_c^2}\right)^2 = \frac{c_\mathrm{s}^2}{2}\left[\frac{\partial g_{\mathrm{eff},s}}{\partial s} + \Delta_c|y_c|^\alpha\frac{\partial\Gamma}{\partial s}\right]. \tag{B10}$$

Substituting Equation B8 into the above yields

$$\left(\frac{y_c}{u_c^2}\right)^2 = \frac{c_\mathrm{s}^2}{2}\Phi^2, \tag{B11}$$

where we define

$$\Phi^2 \equiv \frac{\partial g_{\mathrm{eff},s}}{\partial s} - \frac{g_{\mathrm{eff},s}}{1 - \alpha}\left(\alpha\lambda + \frac{1}{\psi}\frac{\partial\psi}{\partial s} - \frac{2\psi}{r}\right). \tag{B12}$$

Equation B11 has two possible outcomes: $y_c/u_c^2 = \pm c_\mathrm{s}\Phi/\sqrt{2}$. Since $u_c^2 > 0$, the sign of y_c determines which solution to choose. For the case of a radiation-driven outflow, the magnetospheric plasma accelerates as it flows away from the stellar surface. However, due to our sign convention (Section 4.2.2), outflowing plasma can have either a positive or negative velocity. For an increasing arc length away from the stellar surface ($\psi > 0$), an accelerating outflow has $v > 0$, $dv > 0$, and $ds > 0$, resulting in $y_c > 0$. At the opposite line footprint ($\psi < 0$), if there is one, an accelerating outflow requires $v < 0$, $dv < 0$, and

$ds < 0$, resulting in $y_c < 0$. Thus, we take the positive (negative) root of Equation B11 for positive (negative) ψ.

Next, we solve for the critical velocity using Equation B9 and Equation B11:

$$u_c^2 = 1 \mp \frac{\sqrt{2}\alpha}{(1-\alpha)c_s\Phi}\, g_{\mathrm{eff},s}, \tag{B13}$$

which is Equation 4.31. Substituting Equation B9 for u_c^2 instead allows us to solve for y_c:

$$y_c = \pm\frac{c_s\Phi}{\sqrt{2}} - \frac{\alpha}{1-\alpha}\, g_{\mathrm{eff},s}. \tag{B14}$$

Finally, we solve for Δ_c using Equation B8:

$$\Delta_c = -\frac{g_{\mathrm{eff},s}}{(1-\alpha)\Gamma|\pm\frac{c_s\Phi}{\sqrt{2}} - \frac{\alpha}{1-\alpha}\, g_{\mathrm{eff},s}|^{\alpha}}. \tag{B15}$$

C Critical values for general case

As in Appendix B, we will solve for the critical velocity, velocity derivative, and surface mass-flux as a function of the critical point location. First, from the general equation of motion (Equation A2), we make the substitutions $y = vv'$ and $u = v/c_s$. Next, for mathematical simplicity, we define $\bar{\Gamma} = \psi/r^2$:

$$\begin{aligned} F_{\mathrm{gen}} &= y(1-1/u^2) - g_{\mathrm{eff},s} \\ &\quad - \frac{\bar{Q}\Gamma_{\mathrm{el}}GM_*}{1-\alpha}\bar{\Gamma}\frac{|y|}{\xi}\left[(1+\xi/|y|)^{1-\alpha} - 1\right] = 0. \end{aligned} \tag{C16}$$

Next, we evaluate the CAK singularity condition:

$$\begin{aligned} (1-1/u_c^2)y_c &= \frac{\bar{Q}\Gamma_{\mathrm{el}}GM_*}{1-\alpha}\frac{\bar{\Gamma}}{\xi}|y_c|\left[\left(1+\frac{\xi}{|y_c|}\right)^{1-\alpha} - 1\right] \\ &\quad - \bar{Q}\Gamma_{\mathrm{el}}GM_*\bar{\Gamma}\left(1+\frac{\xi}{|y_c|}\right)^{-\alpha}. \end{aligned} \tag{C17}$$

Substituting Equation C17 into Equation C16 yields:

$$\left(1+\frac{\xi}{|y_c|}\right)^{-\alpha} = -\frac{g_{\mathrm{eff},s}}{\bar{Q}\Gamma_{\mathrm{el}}GM_*\bar{\Gamma}}\ . \tag{C18}$$

Since the right-hand side of Equation C18 is wholly dependent on s_{crit}, we define

$$\chi \equiv (1+\xi/|y_c|)^{-\alpha} = (1+\tau_{\mathrm{sob}})^{-\alpha}\,, \tag{C19}$$

τ_{sob} being the Sobolev optical depth. Additionally, we will define the critical value of χ as

$$\chi_c = \chi_c(s_{\mathrm{crit}}) \equiv -g_{\mathrm{eff},s}r^2/[\bar{Q}\Gamma_{\mathrm{el}}GM_*\psi]. \tag{C20}$$

We can further simplify this to $\chi_c = \chi_0\aleph_c$, where

$$\chi_0 = \frac{1-\Gamma_{\mathrm{el}}}{\Gamma_{\mathrm{el}}\bar{Q}} \tag{C21}$$

is the critical χ value for zero rotation and $\aleph_c$ is the rotation effect parameter (Equation 4.64a) evaluated at the critical point.

We discuss the physical meaning of χ as a "correction level parameter" in Section 4.6.2, but we note that χ_c is set by the ratio of the non-radiative external forces to the optically thin ($\tau_{\mathrm{sob}} \ll 1$) radiative force. Since χ can only be between zero and one (since $\tau_{\mathrm{sob}} > 0$), this implies both that, at the critical point, gravity must be stronger than the centrifugal force and the optically thin radiative force must be stronger than the other combined external forces.

These definitions make possible several substitutions:

$$\left(1+\frac{\xi_c}{|y_c|}\right)^{-\alpha} = \chi_c \tag{C22a}$$

$$\left(1+\frac{\xi_c}{|y_c|}\right)^{1-\alpha} = \chi_c^{\frac{\alpha-1}{\alpha}} \tag{C22b}$$

$$|y_c| = \frac{\xi_c}{\chi_c^{-1/\alpha}-1}. \tag{C22c}$$

Now, we evaluate the CAK regularity condition and substitute Equation C22a - Equation C22c:

$$\frac{2y_c^2}{c_s^2 u_c^4} = \frac{\partial g_{\mathrm{eff},s}}{\partial s} - \frac{g_{\mathrm{eff},s}}{1-\alpha}\left[\frac{1-\chi_c^{1/\alpha-1}}{1-\chi_c^{1/\alpha}}\right]\left(\frac{1}{\bar{\Gamma}}\frac{\partial \bar{\Gamma}}{\partial s} + \lambda\right) + g_{\mathrm{eff},s}\lambda, \tag{C23}$$

where we remember $\lambda = \partial A/\partial s/A = -\partial\xi/\partial s/\xi$ (Section 4.3). We can then define $\bar{\Phi}$ in a parallel manner to the optically-thick case (Φ; Equation 4.30):

$$\begin{aligned}\bar{\Phi}^2 \equiv \frac{\partial g_{\mathrm{eff},s}}{\partial s} - \frac{g_{\mathrm{eff},s}}{1-\alpha}\left[\frac{1-\chi_c^{1/\alpha-1}}{1-\chi_c^{1/\alpha}}\right]\left(\frac{1}{\psi}\frac{\partial\psi}{\partial s} - \frac{2\psi}{r}\right) \\ + \frac{1-\alpha-\left[\frac{1-\chi_c^{1/\alpha-1}}{1-\chi_c^{1/\alpha}}\right]}{1-\alpha} g_{\mathrm{eff},s}\lambda\end{aligned} \tag{C24}$$

As a check, we note that the general case reduces to the optically-thick case for $\xi/|y_c| \gg 1$ (i.e. $\tau_{\mathrm{sob}} \gg 1$). This leads to $\chi_c \ll 1$; the terms in the brackets above reduce to 1. From this, it is easily seen that $\bar{\Phi}^2 \to \Phi^2$.

We can use Equation C16 and $\bar{\Phi}^2 = 2y_c^2/(c_s^2 u_c^4)$ to solve for y_c, remembering that both roots of Equation C24 are valid solutions for an accelerating outflow from the stellar surface (c.f. Section 4.3):

$$y_c = \frac{1-\alpha-\left[\frac{1-\chi_c^{1/\alpha-1}}{1-\chi_c^{1/\alpha}}\right]}{1-\alpha}\, g_{\mathrm{eff},s} \pm \frac{c_s\bar{\Phi}}{\sqrt{2}} \tag{C25}$$

where the top (bottom) term applies for positive (negative) ψ. In the optically-thick limit, the term in brackets is ≈ 1 and the overall equation reduces to the optically-thick y_c (Equation 4.32). Similarly, we can obtain the critical velocity:

$$u_c^2 = 1 \pm \frac{\sqrt{2}}{c_s\bar{\Phi}}\left(\frac{1-\alpha-\left[\frac{1-\chi_c^{1/\alpha-1}}{1-\chi_c^{1/\alpha}}\right]}{1-\alpha}\right) g_{\mathrm{eff},s} \tag{C26}$$

which reduces to the optically-thick value Equation 4.31 in the proper limit. Finally, we solve for the critical mass-flux using Equation C22c:

$$\dot{m}_* = \frac{(\chi_c^{-1/\alpha}-1)|y_c|}{c\kappa_e\bar{Q}\frac{A_*}{A}} \tag{C27}$$

As in the optically-thick case, these critical values imply a range to the allowable critical point location, $s_{\rm crit}$, by requiring that $\bar{\Phi}^2 > 0$ and $\chi_c > 0$. With a boundary condition (e.g. Equation 4.39), we can solve for $s_{\rm crit}$ iteratively using the procedure described in Section 4.4.

D Dipole parameterizations

There are several spatial variables that we need to translate from r or s to the magnetic colatitude $\tilde{\theta}$, which we will do in the following paragraphs. For an aligned dipole, the magnetic axis coincides with the rotational axis, so we take $\tilde{\theta} = \theta$. Since the plasma flows along the field line but most of our external forces (Section 4.2.1) depend on the radial distance, it is convenient to simplify the derivative along the field line as $\partial/\partial s = \partial r/\partial s \; \partial/\partial r$. Towards this end, we calculate $\partial r/\partial s$ from Equation 4.41 and Equation 4.42:

$$\frac{\partial r}{\partial \theta} = 2 r_m \sin\theta \cos\theta \tag{D28a}$$

$$\frac{\partial s}{\partial \theta} = r_m \sin\theta \sqrt{1 + 3\cos^2\theta} \tag{D28b}$$

$$\frac{\partial r}{\partial s} = \frac{2\cos\theta}{\sqrt{1 + 3\cos^2\theta}} \tag{D28c}$$

We also have $\psi = \hat{r} \cdot \hat{s} = \partial r/\partial s$. Next, from the the conservation of magnetic flux ($BA = $ const) and our magnetic field definition Equation 4.40a:

$$\frac{A_*}{A} = \left(\frac{R_*}{r}\right)^3 \frac{\sqrt{1 + 3\cos^2\theta}}{\sqrt{1 + 3\cos^2\theta_m}}, \tag{D29}$$

where the surface radius R_* is defined in Equation 4.9. From this, we simplify $\lambda = \partial A/\partial s/A$:

$$\lambda = \psi \left[\frac{3}{r} + \frac{3}{2 r_m (1 + 3\cos^2\theta)}\right]. \tag{D30}$$

Now, we move on to parameterizing the external forces on the plasma in the

magnetosphere. From Section 4.2.5, we get:

$$g_{\text{eff},s} + c_{\text{s}}^2\lambda = -(1-\Gamma_{\text{el}})\frac{GM_*\psi}{r^2}\left(1 - \frac{8\omega^2}{27R_p^3}\frac{\bar{r}r^2\bar{\psi}}{\psi}\right) + c_{\text{s}}^2\lambda, \tag{D31}$$

where $\bar{r} = r\sin\theta$ is the distance from the rotational axis and $\bar{\psi} = \hat{\bar{r}} \cdot \hat{s}$. Typically, the Parker term $c_{\text{s}}^2\lambda$ is neglected. Parameterizing $\bar{r}$ and $\bar{\psi}$ yields

$$\bar{r} = r\sin\theta = r_m \sin^3\theta \ , \tag{D32}$$

$$\frac{\partial\bar{r}}{\partial\theta} = 3r_m \sin^2\theta\cos\theta \ , \tag{D33}$$

$$\bar{\psi} = \frac{\partial\bar{r}}{\partial s} = \frac{3\sin\theta\cos\theta}{\sqrt{1+3\cos^2\theta}} \ . \tag{D34}$$

Since there are several spatial derivatives in the critical point calculations, we calculate $\partial/\partial s$ of several variables:

$$\frac{\partial\psi}{\partial s} = \frac{-2}{r_m(1+3\cos^2\theta)^2}, \tag{D35}$$

$$\frac{\partial\bar{\psi}}{\partial s} = \frac{3(\cos^2\theta - \sin^2\theta) + \bar{\psi}^2}{r_m\sin\theta(1+3\cos^2\theta)}, \tag{D36}$$

$$\frac{\partial\lambda}{\partial s} = \frac{\partial\psi}{\partial s}\left[\frac{3}{r} + \frac{3}{2r_m(1+3\cos^2\theta)}\right] - \frac{3\psi^2}{r^2} + \frac{9\psi\cos\theta}{r_m^2(1+3\cos^2\theta)^{5/2}}. \tag{D37}$$

Now, we present $\partial g_{\text{eff},s}/\partial s$ as required for Φ in (4.30):

$$\frac{\partial g_{\text{eff},s}}{\partial s} = (1-\Gamma_{\text{el}})GM_*\left[\frac{2\psi^2}{r^3} - \frac{\partial\psi/\partial s}{r^2} + \frac{8\omega^2}{27R_p^3}\left(\bar{\psi}^2 + \bar{r}\frac{\partial\bar{\psi}}{\partial s}\right)\right]. \tag{D38}$$

With all our definitions above, we can easily solve for the critical point values (Equation 4.31 - Equation 4.33), and the critical surface mass-flux (Equation 4.34) for a magnetic dipole. In theory, one could write the full critical value equations wholly in terms of θ, but such a representation would be too muddled and provide no benefit. For practical purposes, it is much easier to calculate values along each input magnetic field line and derive the spatial derivatives numerically.

The procedure for calculating the dipole critical values for a general line acceleration is identical to above, since the spatial variables are the same.

E Mass-flux scaling for general case

We reproduce the scaling of Section 4.6.1 for a general line acceleration. As in the optically-thick case, we ignore both the Parker term $c_s^2\lambda$ and Φ:

$$y_c \approx \frac{1-\alpha-\left[\frac{1-\chi_c^{1/\alpha-1}}{1-\chi_c^{1/\alpha}}\right]}{1-\alpha}\, g_{\mathrm{eff},s} = -\frac{\alpha}{1-\alpha}\Sigma g_{\mathrm{eff},s}, \tag{E39}$$

with χ_c defined in Equation C20 and we define a optically-thin correction parameter

$$\Sigma \equiv \frac{\left|1-\alpha-\left[\frac{1-\chi_c^{1/\alpha-1}}{1-\chi_c^{1/\alpha}}\right]\right|}{\alpha}, \tag{E40}$$

whose utility will become evident later on.

Next, we simplify Equation C27 for an aligned dipole, noting that usually $\chi_c^{-1/\alpha} \gg 1$:

$$\begin{aligned}\dot{m}_* \approx \frac{\alpha}{1-\alpha}\Sigma\frac{L_*}{4\pi c^2}|g_{\mathrm{eff},s}|\left[\frac{(\bar{Q}\Gamma_{\mathrm{el}}GM_*)^{1-\alpha}}{-g_{\mathrm{eff},s}}\right]^{1/\alpha} \\ \times \frac{\psi_c^{1/\alpha} r_c^{3-2/\alpha}}{R_*^3}\sqrt{\frac{1+3\cos^2\theta_m}{1+3\cos^2\theta_c}}\end{aligned} \tag{E41}$$

which is identical to Equation 4.49 except for the OTC parameter. For zero rotation, $g_{\mathrm{eff},s} = -(1-\Gamma_{\mathrm{el}})GM_*\psi/r^2$ and $R_* = R_p$:

$$\dot{m}_* \approx \dot{m}_{\mathrm{dip}}\Sigma_0\psi_c\sqrt{\frac{1+3\cos^2\theta_m}{1+3\cos^2\theta_c}}, \tag{E42}$$

with $\dot{m}_{\mathrm{dip}}$, the optically-thick surface mass-flux, defined in Equation 4.51. Here, Σ_0 is the OTC parameter for zero rotation (as discussed in Section 4.6.2).

Taking the OD04-type simplifications $r_{\mathrm{c},\theta=0} - R_* \ll R_*$, $\theta_c \approx \theta_m$ and $\psi_c \approx \psi_m = \mu_B$ (for zero rotation), we get

$$\dot{m}_* \approx \mu_B\Sigma_0\dot{m}_{\mathrm{dip}}(r_{\mathrm{c},\theta=0}), \tag{E43}$$

which is nearly identical to Equation 4.53 with the addition of the OTC parameter. $r_{c,\theta=0}$ is different between the general and optically-thick cases (Section 4.7), however, so that should be taken into account.

Chapter 5

Effect of a magnetic field on massive-star winds II: finite-disk correction factor

It's deja vu all over again.

Yogi Berra

Abstract

We extend our previous analysis of the steady-state Arbitrary Rigid-Field Hydrodynamics Equations by correcting our prior point-star simplification with a finite-disk correction factor. We undertake a critical point analysis to determine how the finite-disk correction changes our understanding of how an aligned dipole magnetic field affects the CAK mass-loss and velocity structure. We compare our results to the finite-disk correction CAK model and determine new scaling relations for the surface mass-flux with surface colatitude. Confirming previous results, the finite-disk correction is found to reduce the stellar mass-loss and increase the wind acceleration compared to the point star case.

5.1 Introduction

In Paper I (Chapter 4), we undertook a steady-state analysis of a Castor et al. (1975) (hereafter CAK) line-driven wind flowing along dipole magnetic field lines for both the optically-thick formulation of the radiative acceleration (a la CAK) and the more general limit (Owocki et al. 1988). For this simplest case, we assumed a point source of radiation such that the line acceleration was solely radial. Here, in Paper II, we relax this assumption of "radially-streaming" photons and treat the star as a finite disk. The transformation between these two cases is provided by the so-called "finite-disk correction factor" (fdcf), which is multiplied with the radially-streaming acceleration.

CAK originally derived the fdcf assuming a stellar disk with uniform intensity and found a complex relationship for the fdcf with space, velocity, and the velocity gradient. This precludes analytical solutions, but numerical solutions can been calculated (Friend & Abbott 1986; Pauldrach et al. 1986). Although the fdcf can be a three-dimensional vector, these papers assume a spherically-symmetric, purely radial outflow and ignore the fdcf angular components. This, however, does not work for magnetospheres due to the non-radial directions of the flow along magnetic field lines.

Another assumption which simplifies the fdcf calculation is a beta-law velocity structure. This allows the fdcf to be evaluated explicitly as a spatial function (Kudritzki et al. 1989; Madura et al. 2007). However, Paper I showed that the assumption of a beta-law velocity does not hold for rotating magnetospheres; moreover, there does not exist a single, global law which applies to every single magnetospheric field line.

In this paper, we add the fdcf to the Arbitrary Rigid-Field Hydrodynamics (ARFHD) equations and study its effects on the steady-state analysis presented in Paper I. Thus, the

difference between Paper I and Paper II is akin to the difference between CAK and later CAK-type models which include the correction factor (Friend & Abbott 1986; Pauldrach et al. 1986; Kudritzki et al. 1989). We stress that we are not making new claims on the very nature of line-driven winds themselves; rather, we are attempting to understand how a strong, dominant magnetic field changes the properties of the wind. For more detailed massive-star wind models, see e.g. Kudritzki (2002) (modified CAK); Müller & Vink (2008) (Monte Carlo technique); Sundqvist & Owocki (2015) (scattering).

We briefly recap the relevant details from ARFHD in Section 5.2. Section 5.3 discusses the finite-disk correction factor and appropriate simplifications for magnetospheres. Our model setup is presented in Section 5.4 and results are presented concerning the surface mass-flux (Section 5.5), the velocity structure (Section 5.6), the closure radii (Section 5.7), and the global mass-loss (Section 5.8).

5.2 Steady-State Wind Equation

In Paper I, we presented the Arbitrary Rigid-Field Hydrodynamic (ARFHD) equations, an extension of the Rigid-Field Hydrodynamic (RFHD) equations (Townsend et al. 2007) to allow for arbitrary magnetic configurations. Here, however, we will assume the simplest case of a magnetic field aligned with the stellar rotation axis. The key assumption of ARFHD is that the magnetic field is effectively rigid and will completely dominate the wind flow, channeling it along magnetic field lines. The plasma state (density, velocity, energy, etc.) is determined by the hydrodynamics of the flow along each magnetic field line and subject to external sources of momentum and energy (gravity, centrifugal force, radiative driving). The field lines are represented as quasi-one-dimensional flux tubes with varying cross-sectional areas. In order to conserve local magnetic flux, the cross-sectional areas are inversely proportional to the local magnetic flux density. Magnetic forces do not

directly influence the flow since they are always directed perpendicular to the instantaneous velocity vector, but they do influence the flow acceleration by channeling the flow along the field lines.

In the reference Cartesian grid comprising the magnetosphere, we define $(x, y, z) = (0, 0, 0)$ as the center of the star and the z-axis as the stellar rotation pole. Each magnetic field line is a three-dimensional space curve $\mathbf{r}(s)$, with s the arc length along the field line chosen such that $\hat{\mathbf{s}} = d\mathbf{r}/ds$ is everywhere parallel to the local magnetic field vector $\mathbf{B}$. We use the sign of the velocity to indicate the direction of flow: positive (negative) means that the plasma is flowing in the direction of increasing (decreasing) s. We additionally define $\bar{r} = |\bar{\mathbf{r}}|$ with $\bar{\mathbf{r}} = [x, y, 0]$ the vector drawn from the rotation axis to the position at $\mathbf{r}$. $\hat{\mathbf{r}}$ and $\hat{\bar{\mathbf{r}}}$ are the unit vectors of $\mathbf{r}$ and $\bar{\mathbf{r}}$, respectively. Further details (and the full set of ARFHD equations) can be found in Paper I (Chapter 4).

In the radiative driving region of the magnetosphere, we make the approximation that the wind is isothermal with the temperature on the order of the stellar effective temperature (Drew 1989), and in a steady state:

$$vv'\left(1 - c_{\rm s}^2/v^2\right) - g_{{\rm eff},s} - c_{\rm s}^2\lambda - g_{{\rm rad},s} = 0. \tag{5.1}$$

where $v' = \partial v/\partial s$ is the velocity gradient *along* the field line and $\lambda \equiv \partial A/\partial s/A$ is the areal gradient term. $c_s = k_b T/\bar{\mu}$ is the isothermal sound speed with $\bar{\mu} \equiv \mu u_{\rm amu}$ the weighed mean molecular weight and $u_{\rm amu}$ the atomic mass unit. The mean molecular weight μ is calculated with the assumption of a fully ionized mixture:

$$\mu = \left[2X_H + \frac{3}{4}(1 - X_H - Z) + \frac{Z}{2}\right]^{-1} \tag{5.2}$$

with X_H and Z the usual hydrogen and metal mass fractions.

$g_{\mathrm{eff},s}$, the effective gravity, is the combined gravitocentrifugal force along the field line:

$$g_{\mathrm{eff},s} = (1-\Gamma_{\mathrm{el}})GM_* \left(\frac{-\psi}{r^2} + \frac{8\omega^2}{27R_p^3}\bar{r}\bar{\psi} \right), \tag{5.3}$$

with the Eddington parameter $\Gamma_{\mathrm{el}} \equiv \kappa_e L_*/(4\pi cGM_*)$ and M_* the mass of the star. Both $\psi = \hat{\mathbf{r}} \cdot \hat{\mathbf{s}}$ and $\bar{\psi} = \hat{\bar{\mathbf{r}}} \cdot \hat{\mathbf{s}}$ account for the effect of magnetic tilt on the effective gravity. We note that R_p is the polar radius of the star and not to be confused with the stellar surface radius R_*. In the Roche approximation (point-like mass distribution and uniform stellar rotation), R_* varies with rotational colatitude θ as

$$\frac{R_*}{R_p} = \frac{3}{\omega \sin\theta} \cos\left[\frac{\pi + \cos^{-1}(\omega \sin\theta)}{3} \right] . \tag{5.4}$$

In both the above equations, we parameterize the stellar rotation with the normalized rotation angular frequency,

$$\omega \equiv \Omega \sqrt{\frac{27R_p^3}{8GM_*(1-\Gamma_{\mathrm{el}})}}, \tag{5.5}$$

with Ω the angular rotation frequency and $\omega = 1$ corresponding to critical rotation.

The radiative driving along the field line, $g_{\mathrm{rad},s}$, is parameterized with the Owocki et al. (1988) version of the usual CAK formalism

$$g_{\mathrm{rad},s} = f \frac{\kappa_e \bar{Q} L_*}{4\pi r^2 c} \frac{(1+\tau_{\mathrm{sob}})^{1-\alpha} - 1}{(1-\alpha)\tau_{\mathrm{sob}}} \ \psi \ , \tag{5.6}$$

where the finite-disk-correction factor (fdcf) is defined as the ratio between the finite-disk-derived and the point-source-derived $g_{\mathrm{rad},s}$:

$$f \equiv \frac{g_{\mathrm{rad},s,\mathrm{disk}}}{g_{\mathrm{rad},s,\mathrm{point}}}. \tag{5.7}$$

In Equation 5.6, L_* is the stellar luminosity, $\bar{Q}$ is the dimensionless line strength parameter introduced by Gayley (1995), κ_e is the electron scattering opacity, α is the CAK-power law index, and

$$\tau_{\mathrm{sob}} \equiv \frac{c\rho\kappa_e\bar{Q}}{|\delta_v|} \tag{5.8}$$

is the Sobolev optical depth. We follow the same procedure as Townsend et al. (2007) and Paper I and assume that the polar velocity derivative vanishes. Thus, we adopt the approximation $\delta_v \approx \partial v/\partial s$. In the optically-thick limit ($\tau_{\rm sob} \gg 1$), Equation 5.8 reduces to the finite-disk CAK formalism

$$g_{{\rm rad},s} = f\frac{\kappa_e \bar{Q} L_*}{4\pi r^2 c}\frac{\tau_{\rm sob}^{-\alpha}}{1-\alpha}\psi. \tag{5.9}$$

In Equation 5.8, we can eliminate the density dependence with the continuity equation,

$$A\rho v = A_*\rho_* v_* = A_*\dot{m}_*, \tag{5.10}$$

with A the cross-sectional area, $\dot{m}_*$ the surface mass-flux into the flux tube, and the subscript $*$ indicating stellar surface values. We can then rewrite the optically-thick $g_{{\rm rad},s}$ as

$$g_{{\rm rad},s} = f\Delta\left(\frac{A}{A_*}\right)^{\alpha}\frac{\psi}{r^2}|vv'|^{\alpha}, \tag{5.11}$$

where we define $v' = dv/ds$ and parameterize the surface mass flux with

$$\Delta \equiv \frac{\left(\bar{Q}\Gamma_{\rm el} G M_*\right)^{1-\alpha}}{1-\alpha}\left(\frac{L_*}{4\pi\dot{m}_* c^2}\right)^{\alpha}. \tag{5.12}$$

5.3 Magnetospheric Finite-Disk Correction Factor

We first consider the radiative line-acceleration from a finite-disk:

$$g_{{\rm rad},s,{\rm disk}} = \frac{\kappa_e \bar{Q}}{(1-\alpha)c}\oint I_*\frac{(1+\frac{\bar{Q}\rho\kappa_e c}{|\hat{\mathbf{n}}\cdot\nabla(\hat{\mathbf{n}}\cdot\mathbf{v})|})^{1-\alpha}-1}{\frac{\bar{Q}\rho\kappa_e c}{|\hat{\mathbf{n}}\cdot\nabla(\hat{\mathbf{n}}\cdot\mathbf{v})|}}\,\hat{\mathbf{n}}\,d\mu' d\phi', \tag{5.13}$$

The stellar surface intensity, taken as uniform over the surface, is

$$I_* = \begin{cases} 0, & -1 \le \mu' < \mu_* \\ L_*/(\pi\Sigma_{\rm surf}), & \mu_* \le \mu' < 1 \end{cases}, \tag{5.14}$$

with $\mu_* = \cos(\theta'_*)$ marking the edge of the stellar disk (see discussion below). The surface area of an oblate star is given by the power law fit (Cranmer 1996)

$$\Sigma_{\rm surf} = 4\pi R_p^2 \, (1 + 0.91444\omega^2 + 0.28053\omega^4 \\ - 1.9014\omega^6 + 6.8298\omega^8 - 9.5002\omega^{10} + 4.6631\omega^{12}). \quad (5.15)$$

We generalize the one-dimensional Sobolev optical depth (Equation 5.8) with the three-dimensional projected gradient of the projected velocity along each line of sight $\hat{\mathbf{n}}$: $\hat{\mathbf{n}} \cdot \nabla(\hat{\mathbf{n}} \cdot \mathbf{v})$.

The wind, at some spatial location $\mathbf{r}_0 = [x_0, y_0, z_0]$, sees a finite disk of radiation from the stellar surface. Since the photons are no longer streaming purely radially through $\mathbf{r}_0$, the radial line acceleration is diluted. There may even be angular acceleration induced by non-radial wind velocity components and by variations in the stellar surface intensity caused by limb-darkening or rotation. Here, however, we do not consider the effects of limb or gravity darkening.

With the star centered at $(x, y, z) = (0, 0, 0)$ (Section 5.2), we follow Cranmer & Owocki (1995) and define a "wind-centered" coordinate system $\mathbf{r}' = [x', y', z']$ by translating the origin to a wind point $\mathbf{r}_0 = [x_0, y_0, z_0]$ and rotating the axes such that the z'-axis points along $\mathbf{r}$ away from the star and the y'-axis is parallel to the y-axis. The x'-axis is thus tilted by an angle $\theta_0 = \arccos(z_0/r_0)$ (with $r_0 = |\mathbf{r}_0|$) relative to the x-axis, giving a transformation between the two systems:

$$x' = x\cos\theta_0 - z\sin\theta_0$$

$$y' = y$$

$$z' - r_0 = x\sin\theta_0 + z\cos\theta_0. \quad (5.16)$$

Figure 5.1 illustrates the coordinate geometry of the star- and wind-centered frames.

Evaluating $g_{\mathrm{rad},s,\mathrm{disk}}$ requires evaluation of the stellar intensity and a projected gradient of the projected velocity along arbitrary rays $\hat{\mathbf{n}}$, directed from points on the stellar surface to the wind point. The necessary angular integrations are most conveniently done in the wind-centered spherical coordinates, (r', θ', ϕ'), so we express the ray unit vector components as

$$\begin{aligned} \hat{n}_r &= \cos\theta' = \mu', \\ \hat{n}_\theta &= \sin\theta' \cos\phi' = \sqrt{1-\mu'^2}\cos\phi', \\ \hat{n}_\phi &= \sin\theta' \sin\phi' = \sqrt{1-\mu'^2}\sin\phi' \ . \end{aligned} \tag{5.17}$$

In this system, we take the fdcf as the ratio of $g_{\mathrm{rad},s,\mathrm{disk}}$ to $g_{\mathrm{rad,s,point}}$ (Equation 5.6

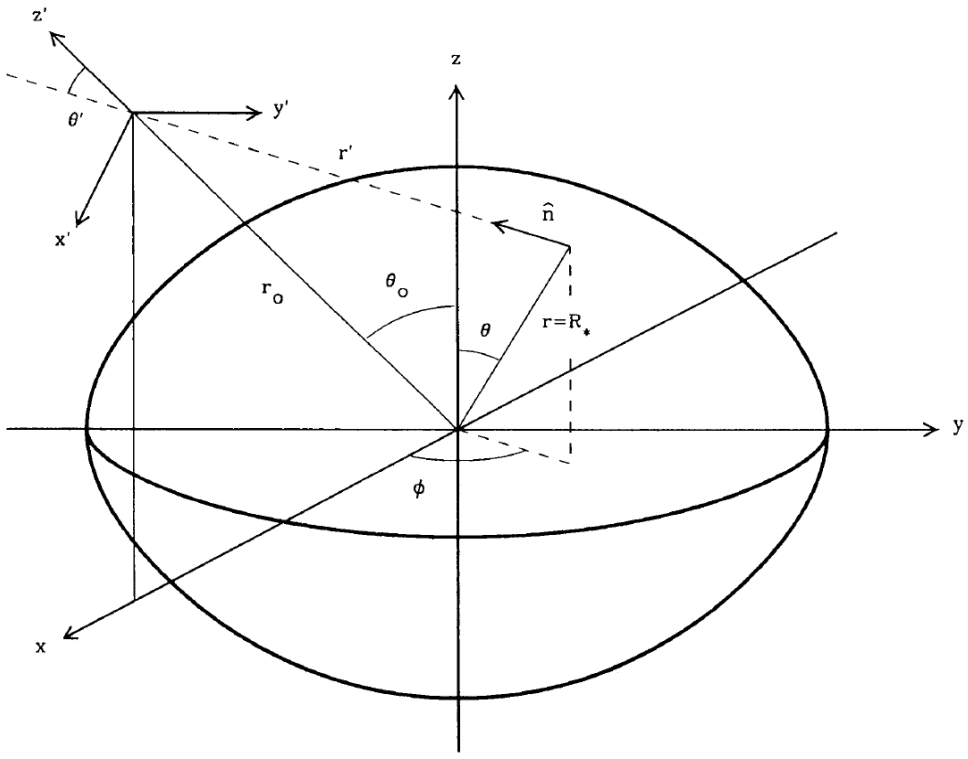


Figure 5.1 : Coordinate geometry for the star-centered (unprimed) and wind-centered (primed) coordinates described in the text, related by the position of a wind point (r_0, θ_0, ϕ_0). Rays $\hat{\mathbf{n}}$ intercept the stellar surface at a distance r' from the field point and are specified by wind-centered angles θ' and ϕ'. Figure taken from Figure 3 of Cranmer & Owocki (1995).

without f):

$$\mathbf{f} = \frac{\oint |\hat{\mathbf{n}} \cdot \nabla(\hat{\mathbf{n}} \cdot \mathbf{v})| \left[\left(1 + \frac{\bar{Q}\rho\kappa_e c}{|\hat{\mathbf{n}} \cdot \nabla(\hat{\mathbf{n}} \cdot \mathbf{v})|} \right)^{1-\alpha} - 1 \right] \hat{\mathbf{n}} \, d\mu' d\phi'}{\pi[\Sigma_{\rm surf}/(4\pi r_0^2)] \, |\partial v/\partial s| \left[\left(1 + \frac{\bar{Q}\rho\kappa_e c}{|\partial v/\partial s|} \right)^{1-\alpha} - 1 \right]}, \tag{5.18}$$

with the integral over the stellar surface as seen from $\mathbf{r}_0$.

Due to rotationally-induced oblateness, each individual ϕ' may have an unique limb angle μ'_* and integration range $[\mu'_*, 1]$ which we have to solve for. For zero rotation, the edge of the star is simply marked by $\mu'_* = \sqrt{1 - (R_p/r_0)^2}$.

Evaluating the integral can be done with any quadrature method (e.g. Simpson's Rule, trapezoidal method); we use Gauss-Legendre quadrature (sample implementation in Press et al. 2002). Further details on the calculation of the full finite-disk factor, including limb- and gravity-darkening, are presented in Thesis Appendix A.

The line-of-sight velocity gradient along each tracing ray requires knowledge of the velocity structure in three-dimensional space. Unfortunately, in the ARFHD ansatz, each line is limited to its own local velocity structure along the line. In the original RFHD paper, Townsend et al. (2007) found that approximating the local magnetospheric velocity gradient with the gradient along the field line ($\partial v/\partial s$) resulted in less than 10% error in the radiatively driven lines along the magnetosphere. Thus, for simplicity, we take only the radial component of the line-of-sight velocity gradient and take $\partial v_r/\partial r \approx \partial v/\partial s$ and $v \approx v_r$:

$$\hat{\mathbf{n}} \cdot \nabla(\hat{\mathbf{n}} \cdot \mathbf{v}) \approx \mu'^2 \frac{\partial v}{\partial s} + (1 - \mu'^2) \frac{v}{r_0}. \tag{5.19}$$

This assumption does not work well in the magnetic equatorial regions ($\theta \gtrsim 60°$) due to non-radial flows induced by the tilt of field lines. However, in the polar and mid-latitude regions, the velocity in the wind-acceleration region is mainly radial, and this approximation

works well. We will leave more realistic magnetospheric line-of-sight velocity gradients to future papers.

We note that the fdcf is actually a vector since variations in the stellar disk intensity and the line-of-sight velocity gradients along each ray can induce non-radial acceleration in the wind. We can break up the components into $f_r,\ f_\theta,\ f_\phi$ by applying the appropriate component of the ray unit vector:

$$f_r = \frac{\oint \ldots\ \hat{n}_r\ d\mu' d\phi'}{\ldots}, \tag{5.20}$$

and similar for f_θ (using $\hat{n}_\theta$) and f_ϕ ($\hat{n}_\phi$). In the magnetosphere, we correct for the magnetic redirection of the wind by taking $f = \mathbf{f} \cdot \hat{\mathbf{s}}$, though our radial approximation of the line-of-sight gradient (Equation 5.19) means that only the radial component of f is nonzero.

With these modifications and assumptions, our final fdcf is

$$f \approx f_r = \frac{\oint \left|\mu'^2 \frac{\partial v}{\partial s} + (1-\mu'^2)\frac{v}{r_0}\right| \left[\left(1 + \frac{\bar{Q}\rho\kappa_e c}{|\mu'^2 \frac{\partial v}{\partial s} + (1-\mu'^2)\frac{v}{r_0}|}\right)^{1-\alpha} - 1\right]\ \mu'\ d\mu' d\phi'}{\pi[\Sigma_{\rm surf}/(4\pi r_0^2)]\ |\partial v/\partial s| \left[\left(1 + \frac{\bar{Q}\rho\kappa_e c}{|\partial v/\partial s|}\right)^{1-\alpha} - 1\right]}\ . \tag{5.21}$$

In the optically thick limit, we recover the CAK-type finite-disk correction:

$$f \approx \frac{\oint \left|\mu'^2 \frac{\partial v}{\partial s} + (1-\mu'^2)\frac{v}{r_0}\right|^{\alpha}\ \mu'\ d\mu' d\phi'}{\pi[\Sigma_{\rm surf}/(4\pi r_0^2)]\ |\partial v/\partial s|^{\alpha}}, \tag{5.22}$$

which, for zero rotation, evaluates to the CAK-derived factor

$$f \approx \frac{(1+\sigma)^{1+\alpha} - (1+\sigma\mu_*^2)^{1+\alpha}}{(1+\sigma)^{\alpha}\sigma(1+\alpha)(1-\mu_*^2)} \tag{5.23}$$

where $\sigma = d\ln v_r / d\ln r - 1$.

5.4 Model Setup

With our equation of motion and the finite-disk correction factor defined, we now follow the procedure outlined in Paper I for calculating the wind values at the CAK critical

point. In the optically-thick limit, we define

$$F[s, y, u] \equiv y(1 - 1/u^2) - g_{\mathrm{eff},s} - f\Delta \left(\frac{A}{A_*}\right)^{\alpha} \frac{\psi}{r^2}|y|^{\alpha}, \tag{5.24}$$

where $u \equiv v/c_{\mathrm{s}}$ and $y \equiv vv'$, such that the equation of motion is

$$F[s, y, u] = 0. \tag{5.25}$$

Following CAK and Abbott (1980), we find the wind critical values by simultaneously satisfying the singularity condition

$$\frac{\partial F}{\partial y} = 0 \tag{5.26}$$

and the regularity condition

$$\frac{\partial F}{\partial s} + \left(\frac{y}{c_{\mathrm{s}}^2 u}\right) \frac{\partial F}{\partial u} = 0. \tag{5.27}$$

Following Paper I and Bjorkman (1995), we treat the critical point location s_{crit} as a free parameter and solve for Δ, y, and s at $s = s_{\mathrm{crit}}$. At the stellar surface, we define the boundary density such that the electron scattering optical depth $\tau_{es} \approx 1$:

$$\rho_0 = \frac{\tau_{es}}{\kappa_e H} \approx \frac{(1 - \Gamma_{\mathrm{el}})GM_*}{\kappa_e R_*^2 c_{\mathrm{s}}^2} \tag{5.28}$$

where $H = c_{\mathrm{s}}^2/g = c_{\mathrm{s}}^2 R_*^2/[(1 - \Gamma_{\mathrm{el}})GM_*]$ is the pressure scale height, corrected with the Eddington parameter. Using the continuity equation and the critical surface mass flux, we can solve for the boundary velocity:

$$v_0 = \frac{\rho_* v_*}{\rho_0} = \frac{\dot{m}_*}{\rho_0} \tag{5.29}$$

where $\dot{m}_* \equiv \rho_* v_*$ is defined as the surface mass-flux into the field line.

This introduces a dependence on s_{crit} for both the boundary and critical point velocities, so we implement an iterative algorithm for choosing s_{crit} appropriately:

1. Choose trial critical point location.
2. Calculate velocity, mass-flux, density at the critical point.
3. Integrate to boundary, taking the smallest positive root of the possible velocity derivatives.
4. If resulting boundary velocity is too high, move $s_{\rm crit}$ out. Else if too low, move $s_{\rm crit}$ in.
5. Repeat from step 2 until correct boundary values are reached.

Our initial critical point guess for the fdcf case is the non-fdcf critical point.

The mathematical derivation of the critical point values is complicated by the finite-disk factor, $f = f(s, y, u)$, which introduces a whole new level of non-linear complexity into the analysis. With the fdcf, we are not able to solve for the critical values wholly in terms of spatial variables as we were able in Paper I. However, we can derive a system of three equations and three unknowns (y, u, Δ) which we can solve using numerical root-finding:

$$y_c = \pm\frac{c_s\Phi_f}{\sqrt{2}} - \frac{\alpha - \frac{y_c}{f_c}\frac{\partial f}{\partial y}}{1-\alpha-\frac{y_c}{f_c}\frac{\partial f}{\partial y}}\, g_{{\rm eff},s} \tag{5.30a}$$

$$u_c^2 = 1 \mp \frac{\sqrt{2}g_{{\rm eff},s}}{c_s\Phi_f}\frac{\alpha - \frac{y_c}{f_c}\frac{\partial f}{\partial y}}{1-\alpha-\frac{y_c}{f_c}\frac{\partial f}{\partial y}} \tag{5.30b}$$

$$\Delta_c = \frac{-g_{{\rm eff},s}}{1-\alpha-\frac{y_c}{f_c}\frac{\partial f}{\partial y}}\frac{r^2}{f_c\psi|y_c|^\alpha(A/A_*)^\alpha} \tag{5.30c}$$

with derivatives evaluated with critical point values, the top (bottom) sign applies for $\psi > 0$ (< 0), and we define Φ_f as the positive square root of

$$\Phi_f^2 = \frac{\partial g_{{\rm eff},s}}{\partial s} - \frac{g_{{\rm eff},s}}{1-\alpha-\frac{y_c}{f_c}\frac{\partial f}{\partial y}}\left[\alpha\lambda + \frac{1}{\psi_c}\frac{\partial\psi}{\partial s} - \frac{2\psi_c}{r_c} + \frac{1}{f_c}\frac{\partial f}{\partial s} + \frac{y_c}{c_s^2 u_c f_c}\frac{\partial f}{\partial u}\right]. \tag{5.31}$$

The surface mass-flux is then obtained from Δ_c through

$$\dot{m}_* = \frac{L_*}{4\pi c^2}\left(\bar{Q}\Gamma_{\rm el}GM_*\right)^{\frac{1-\alpha}{\alpha}}\left(\frac{1}{(1-\alpha)\Delta_c}\right)^{1/\alpha}. \tag{5.32}$$

We can reproduce the results of the non-fdcf optically-thick critical point values by taking $f = 1$ and nullifying all fdcf derivatives. The mathematical details of this derivation are

presented in Appendix A, and a similar derivation for the general $g_{\mathrm{rad},s}$ case is presented in Appendix C.

In these critical point equations, we get a clear picture of how the fdcf affects the critical values. Most obviously, the surface mass-flux scales as $\dot{m}_* \propto \Delta_c^{-1/\alpha} \propto f^{1/\alpha}$, which matches the numerical results of Friend & Abbott (1986), Pauldrach et al. (1986) and the analytical derivation of Kudritzki et al. (1989). Another effect is induced by the fdcf spatial and velocity derivatives in Φ_f^2; Φ_f^2 increases by about a factor of a few, which lowers the velocity at the critical point by roughly a factor of 2. Finally, the fdcf slightly changes the impact of the CAK power-law index α. Typical values of $y_c/f_c\ \partial f/\partial y$ are on the order 0.01, between one and two orders of magnitude smaller than usual values of $\alpha \sim 0.5 - 0.65$ (Puls et al. 2000). This reduces the magnitude of the α fraction term in Equations 5.30a and 5.30b by 5% to 10%.

In order to obtain the critical point values, we must simultaneously solve Equation 5.30a-Equation 5.30c. We do this by defining a set of three equations

$$0 = y_{\mathrm{guess}} - y_c \tag{5.33a}$$

$$0 = u^2_{\mathrm{guess}} - u^2_c \tag{5.33b}$$

$$0 = \dot{m}_{*,\mathrm{guess}} - \dot{m}_*, \tag{5.33c}$$

and solving them with a multi-root solver which combines Newton's method for systems of nonlinear equations with a line-search method to ensure global convergence. We start with the non-fdcf $(y_c, u_c^2, \dot{m}_*)$ as an initial guess, and iterate with the solver until a relative error of $|\mathrm{guess} - \mathrm{crit}|/\mathrm{crit} < 4 \times 10^{-5}$ is reached. The reader is referred to Dennis & Schnabel (1983) for details (and the algorithm) of this method.

In our model, we assume an aligned dipole magnetic field

$$\mathbf{B} = \frac{B_0}{2(r/R_p)^3}\left(2\cos\tilde{\theta}\hat{r} + \sin\tilde{\theta}\hat{\tilde{\theta}}\right), \tag{5.34a}$$

$$\hat{B} = \frac{\mathbf{B}}{|\mathbf{B}|} = \frac{2\cos\tilde{\theta}\hat{r} + \sin\tilde{\theta}\hat{\tilde{\theta}}}{\sqrt{1+3\cos^2\tilde{\theta}}}, \tag{5.34b}$$

with $\tilde{\theta}$ the magnetic colatitude. Each field line can be uniquely identified by its surface colatitude, $\sin\theta_m = \sqrt{R_*/(R_p L)}$, and its magnetic shell parameter L such that

$$r(\theta) = R_p L \sin^2\theta. \tag{5.35}$$

The cross-sectional area A is determined under the requirement that local magnetic flux is conserved (BA = const). Since the field is aligned with the rotational axis, the magnetic and rotational colatitudes are equal: $\tilde{\theta} = \theta$.

We generate 500 dipole field lines with footprints covering the northern half of the star ($0 < \theta < \pi/2$) in linear space, and repeat this for several critical rotation fractions $\omega = [0.0, 0.2, 0.35, 0.5, 0.65, 0.8]$. Field lines with lengths $> 100R_p$ are arbitrarily truncated at $R = 100R_p$; the rest of the lines extend to the magnetic equator.

We use the same two stars as defined in Paper I (Table 5.1), with parameters for the B-type taken from Bjorkman (1995) and the O-type from ud-Doula & Owocki (2002). The B star is analogous to the centrifugal magnetosphere archetype σ Orionis E, while the O star is analogous to ζ Puppis, a star with a dynamical magnetosphere.

Table 5.1 : Stellar and wind parameters used throughout this paper to represent a typical magnetic B-type star with a centrifugal magnetosphere and an O-type star with a dynamical magnetosphere. Values are taken from Table 2 in Bjorkman (1995) for the B star and ud-Doula et al. (2014) for the O star.

Type	M_*	R_p	$T_{\rm eff}$	α	$\Gamma_{\rm el}$	$\bar{Q}$	B_p	η_*
B	9.0 $M_\odot$	4.5 $R_\odot$	21000 K	0.56	9.27×10^{-3}	1025.14	11 kG	4.29×10^5
O	50 $M_\odot$	19 $R_\odot$	41860 K	0.6	0.5	500	3.715 kG	100

We note that the stellar effective temperature, $T_{\rm eff}$ is used to set the stellar luminosity L_* through $L_* = 4\pi R_p^2 \sigma_B T_{\rm eff}^4$. Since we fix L_* independent of rotation rate, the effective $T_{\rm eff}$ will change as the surface area of the star changes due to oblation. L_* is fixed rather than $T_{\rm eff}$ since it is a more directly observable quantity.

5.5 Effect of Correction Factor on Surface Mass-Flux

5.5.1 Review of Paper I results

In Paper I, we derived a scaling relation for the steady-state surface mass-flux as a function of surface colatitude and detailed how a dipole magnetic field affects the wind through rotation and field tilt. Additionally, we parameterized the effect of relaxing the CAK optically-thick assumption in favor of the general Owocki et al. (1988) formulation of the radiative driving. Here, we briefly recap the results before going into detail about the effects of the finite-disk correction factor on the wind mass-flux.

The overall scaling relation for the surface mass-flux is

$$\dot{m}_* \approx \mu_B \dot{m}_{\rm dip}(r_{c,\theta=0}) \Sigma_{\rm rot,m} \aleph_m^{1-1/\alpha}, \tag{5.36}$$

with $\mu_B = \hat{\mathbf{n}} \cdot \hat{\mathbf{B}}$ the cosine of the angle between the stellar surface and the magnetic field vector at the line footprint, $r_{c,\theta=0}$ the critical radius at the pole ($\theta = 0$), and

$$\begin{aligned} \dot{m}_{\rm dip} = \dot{m}_{\rm dip}(r_c) &\equiv \frac{\alpha}{1-\alpha} \frac{L_*}{4\pi c^2} \left(\frac{\bar{Q}\Gamma_{\rm el}}{1-\Gamma_{\rm el}} \right)^{\frac{1-\alpha}{\alpha}} \frac{r_c}{R_p^3} \\ &= \dot{m}_{\rm CAK} \frac{r_c}{R_p}, \end{aligned} \tag{5.37}$$

is the zero-tilt, zero-rotation, optically-thick surface mass-flux for a magnetic dipole. $\dot{m}_{\rm dip}$ is essentially the CAK surface mass-flux

$$\dot{m}_{\rm CAK} = \frac{\dot{M}_{\rm CAK}}{4\pi R_p^2} = \frac{L_*}{4\pi R_p^2 c^2} \frac{\alpha}{1-\alpha} \left(\frac{\bar{Q}\Gamma_{\rm el}}{1-\Gamma_{\rm el}} \right)^{(1-\alpha)/\alpha}, \tag{5.38}$$

corrected for dipole divergence, with $\dot{M}_{\rm CAK}$ the CAK mass-loss rate in the Gayley (1995) $\bar{Q}$ formalism.

The influence of rotation is encapsulated in a "rotation effect parameter"

$$\aleph_m \equiv 1 - \frac{12\cos^3\left[\frac{1}{3}(\pi + \cos^{-1}(\omega \sin\theta_m))\right]}{\omega \sin\theta_m}, \tag{5.39}$$

with θ_m the surface colatitude. At zero rotation ($\omega \rightarrow 0$), $\aleph_m \rightarrow 1$ and its influence disappears.

The difference between using the CAK $g_{{\rm rad},s}$ and the Owocki et al. (1988) $g_{{\rm rad},s}$ is parameterized by an "optically-thin correction"

$$\Sigma_{\rm rot,m} = \frac{\left|1 - \alpha - \left[\frac{1-\chi_0^{1/\alpha-1}\aleph_m^{1/\alpha-1}}{1-\chi_0^{1/\alpha}\aleph_m^{1/\alpha}}\right]\right|}{\alpha} \tag{5.40}$$

with

$$\chi_0 = (1 - \Gamma_{\rm el})/(\Gamma_{\rm el}\bar{Q}). \tag{5.41}$$

χ_0 is the critical point value, at zero rotation, of

$$\chi \equiv (1 + \tau_{\rm sob})^{-\alpha}\,. \tag{5.42}$$

χ is a "wind transparency" parameter. It denotes the importance of using the general form of the radiative driving rather than assuming an optically-thick wind. For optically-thick winds, $\tau_{\rm sob} \gg 1$, $\chi \rightarrow 0$, and $\Sigma_{\rm rot,m} \rightarrow 1$. For optically-thin winds, χ increases and $\Sigma_{\rm rot,m}$ decreases. As an example, our non-rotating model O star has $\chi_0 = 0.002$ and $\Sigma_{\rm rot,m} = 0.974$ while our non-rotating B star has $\chi_0 = 0.104$ and $\Sigma_{\rm rot,m} = 0.724$. This optically-thin correction is not so important for O stars with their high mass-loss rates, but it is important for B stars.

5.5.2 Finite-Disk Factor Scaling

Since the magnetospheric finite-disk correction factor depends on the state of the plasma, it will scale with field line surface colatitude based on the critical point conditions and the perceived size of the oblate stellar disk. At zero rotation, the relationship between the finite-disk factor and surface colatitude is simple. Since the star is not rotating, the critical point location and velocity are nearly constant over the stellar surface (Paper I). Additionally, without oblation, the stellar disk appears as the same size from every wind point. The chief source of variation then comes from the tilt of the magnetic field line through $f = \mathbf{f} \cdot \hat{\mathbf{s}}$:

$$f_c = \psi_c f_{\theta=0}, \tag{5.43}$$

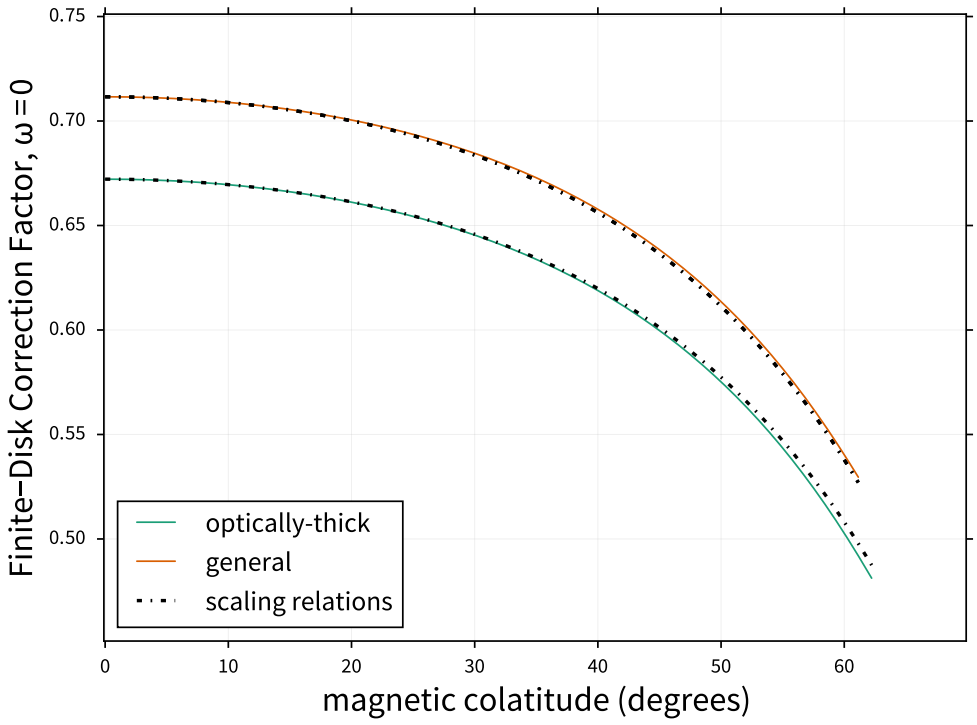


Figure 5.2 : Finite-disk correction factor (solid lines) as a function of surface colatitude for both the general and optically-thick line acceleration for a B-type star (Table 5.1). The calculations are compared to derived scaling relations (dot-dashed; Equation 5.44). Similar results for the O star not shown.

with ψ_c the value of ψ evaluated at each critical point and $f_{\theta=0}$ is the fdcf evaluated at the polar critical radius. We can simplify this by assuming that the critical radius is sufficiently close to the star such that $\psi_c = \mu_B$, leading to a zero-rotation scaling

$$f_c \sim \mu_B f_{\theta=0}. \tag{5.44}$$

Figure 5.2 demonstrates this scaling.

When rotation (or darkening) effects are included, the complexity of the fdcf integral makes analytic solutions untenable. We thus defer any attempt on deriving an approximate fdcf scaling relation for oblate stars for future papers. For now, we use the model-calculated

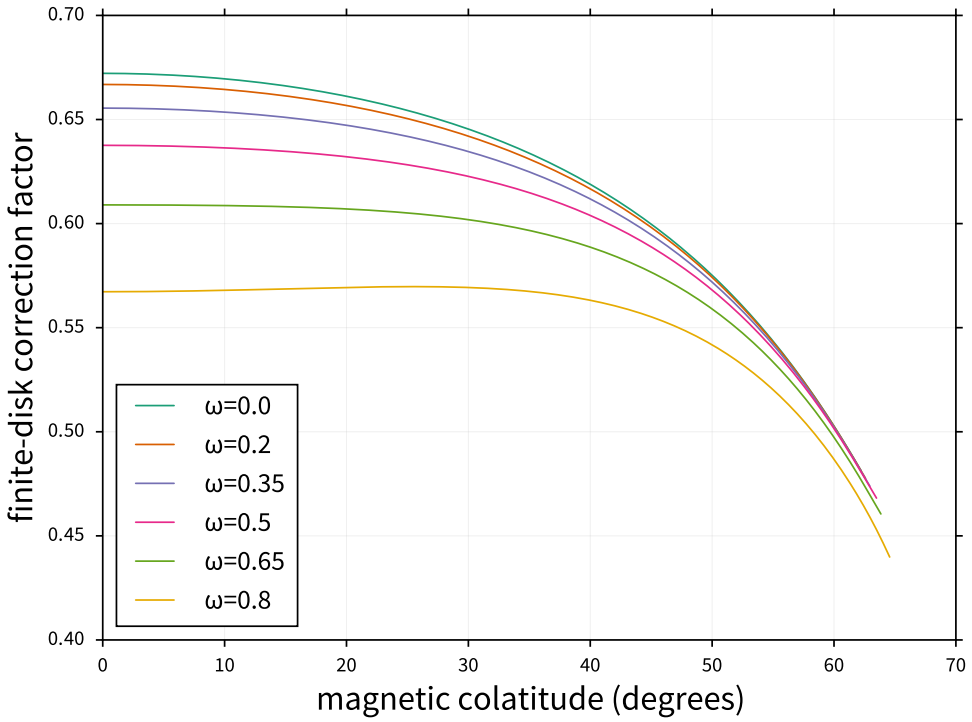


Figure 5.3 : Finite-disk correction factor (solid lines) as a function of surface colatitude and rotation rate ω for a B-type star with optically-thick line acceleration. The decrease in radiative driving at higher rotation rates is caused by the decrease in surface intensity (see text). Similar results for the O star and for a general line-accleration not shown.

fdcf values to define

$$f_{\rm ratio}(\omega, \theta_c) \equiv \frac{f_c(\omega, \theta_c)}{f_{\theta=0}(\omega)} \tag{5.45}$$

such that $f_c = f_{\rm ratio} f_{\theta=0}$ for a given rotation fraction. $f_{\rm ratio}$ can be replaced with any appropriate function which describes the oblate-star fdcf scaling.

Figure 5.3 shows the fdcf values for various rotation rates across surface colatitude. The decrease in fdcf magnitude at higher ω is caused by our surface intensity parameterization (Equation 5.14). Since we keep L_* fixed in our model, the surface intensity decreases as the stellar surface area increases and this reduces the overall radiative driving.

Since we require an analytic scaling in calculating the global mass-loss rates (Section 5.8), we will use nonlinear least-squares to fit $f_{\rm ratio}$ to the equation

$$f_{\rm ratio} = \frac{f_c}{f_{\theta=0}} \approx a_1 \mu_m + a_2 \mu_m^2 + a_3 \mu_m^3 + a_4 \mu_m^4, \tag{5.46}$$

with $\mu_m = \cos\theta_m$. The fit parameters for each rotation rate and type of line-driving are given in Table 5.2; at zero rotation we simply use $f \sim \mu_B f_{\theta=0}$.

5.5.3 Optically-thick limit

We can check our model results with a simple scaling relation derived from our critical point values, a la Paper I. For simplicity, we assume that the Φ_f term $\ll g_{{\rm eff},s}$

Table 5.2 : Best-fit parameters (a_1, a_2, a_3, a_4) for the scaling of the finite-disk correction factor with surface colatitude assuming the equation form given in Equation 5.46. Values for our model B and O stars are given at several rotation rates for both the "general" and optically-thick ("CAK") radiative driving.

	B, CAK	O, CAK	B, general	O, general
$\omega = 0.2$	(2.19, -1.45, 0.08, 0.18)	(2.14, -1.31, -0.049, 0.22)	(2.34, -1.91, 0.57, 0.003)	(2.15, -1.33, -0.026, 0.21)
$\omega = 0.35$	(2.22, -1.43, -0.01, 0.22)	(2.17, -1.27 , -0.17, 0.27)	(2.36, -1.86, 0.45, 0.057)	(2.17, -1.29, -0.15, 0.27)
$\omega = 0.5$	(2.28, -1.4, -0.20, 0.31)	(2.20, -1.16, -0.44, 0.39)	(2.38, -1.73, 0.18, 0.17)	(2.21, -1.17, -0.42, 0.38)
$\omega = 0.65$	(2.34, -1.25, -0.59, 0.50)	(2.22, -0.87, -0.99, 0.64)	(2.38, -1.44, -0.34, 0.40)	(2.22, -0.86, -1.00, 0.64)
$\omega = 0.8$	(2.33, -0.66, -1.67, 0.99)	(2.10, 0.088, -2.48, 1.29)	(2.25, -0.54, -1.69, 0.99)	(2.09, 0.13, -2.52, 1.31)

in Equation 5.30a and $\alpha >> y_c/f_c\ \partial f/\partial y$ such that

$$y_c \approx -\frac{\alpha}{1-\alpha} g_{\mathrm{eff},s}. \tag{5.47}$$

For an aligned dipole, the term A_*/A can be derived under the requirement that local magnetic flux is conserved ($BA = \mathrm{const}$). From this and Equation 5.34a, we obtain

$$\left(\frac{A_*}{A}\right)^{\alpha} = \left(\frac{R_*}{r_c}\right)^{3\alpha} \left(\sqrt{\frac{1+3\cos^2\theta_c}{1+3\cos^2\theta_m}}\right)^{\alpha}. \tag{5.48}$$

Since these values are the same as in the non-fdcf case (Paper I), the only difference in the fdcf case comes from the $1/f$ factor in Δ_c (Equation 5.30c). We thus derive at zero rotation

$$\dot{m}_* \approx f_c^{1/\alpha} \dot{m}_{\mathrm{dip}} \psi_c \sqrt{\frac{1+3\cos^2\theta_m}{1+3\cos^2\theta_c}}, \tag{5.49}$$

where f_c is evaluated at the critical point and $\dot{m}_{\mathrm{dip}}$ is as defined in Equation 5.37. As in Paper I, we assume that the critical radius is sufficiently close to the star such that $\theta_c \approx \theta_m$ and $\psi_c \approx \psi_m = \mu_B$. From Section 5.5.2, we make the additional assumption that the fdcf scales with surface colatitude as $f_c \sim f_{\theta=0}\mu_B$. Our resulting scaling relation is

$$\dot{m}_* \approx \mu_B (\mu_B f_{\theta=0})^{1/\alpha} \dot{m}_{\mathrm{dip}}. \tag{5.50}$$

We can correct for neglecting the Φ_f and $y_c/f_c\ \partial f/\partial y$ terms by using the model-calculated $\dot{m}_{\theta=0}$ in place of $f_{\theta=0}^{1/\alpha}\dot{m}_{\mathrm{dip}}$:

$$\dot{m}_* \sim \mu_B^{1+1/\alpha} \dot{m}_{\theta=0}. \tag{5.51}$$

Figure 5.4 shows the excellent agreement of Equation 5.51 with the steady-state model calculation.

Since the finite-disk effect does not influence the centrifugal force and vice versa, it is straightforward to multiply in the rotation effect parameter, $\aleph$, into Equation 5.49:

$$\dot{m}_* \approx f_c^{1/\alpha} \aleph^{1-1/\alpha} \dot{m}_{\mathrm{dip}} \left(\frac{R_*}{R_p}\right)^3 \psi_c \sqrt{\frac{1+3\cos^2\theta_m}{1+3\cos^2\theta_c}}, \tag{5.52}$$

where the $(R_*/R_p)^3$ term accounts for the increase in stellar radius due to rotation (Equation 5.4). We can take similar approximations as the non-rotating case, but with a different fdcf scaling for rotation: $f_c = f_{\rm ratio} f_{\theta=0}$ (Equation 5.45):

$$\dot{m}_* \approx \mu_B \aleph_m^{1-1/\alpha} (f_{\rm ratio} f_{\theta=0})^{1/\alpha} \dot{m}_{\rm dip}, \tag{5.53}$$

where $\aleph_m$ is the rotation effect parameter evaluated at the stellar surface. Correcting for the neglect of Φ_f and $y_c/f_c\ \partial f/\partial y$, we use $\dot{m}_{\theta=0}$ instead of $f_{\theta=0}^{1/\alpha}\dot{m}_{\rm dip}$ to get the scaling

$$\dot{m}_* \sim \mu_B f_{\rm ratio}^{1/\alpha}\ \aleph_m^{1-1/\alpha} \dot{m}_{\theta=0}. \tag{5.54}$$

Figure 5.5 demonstrates this optically-thick rotation scaling.

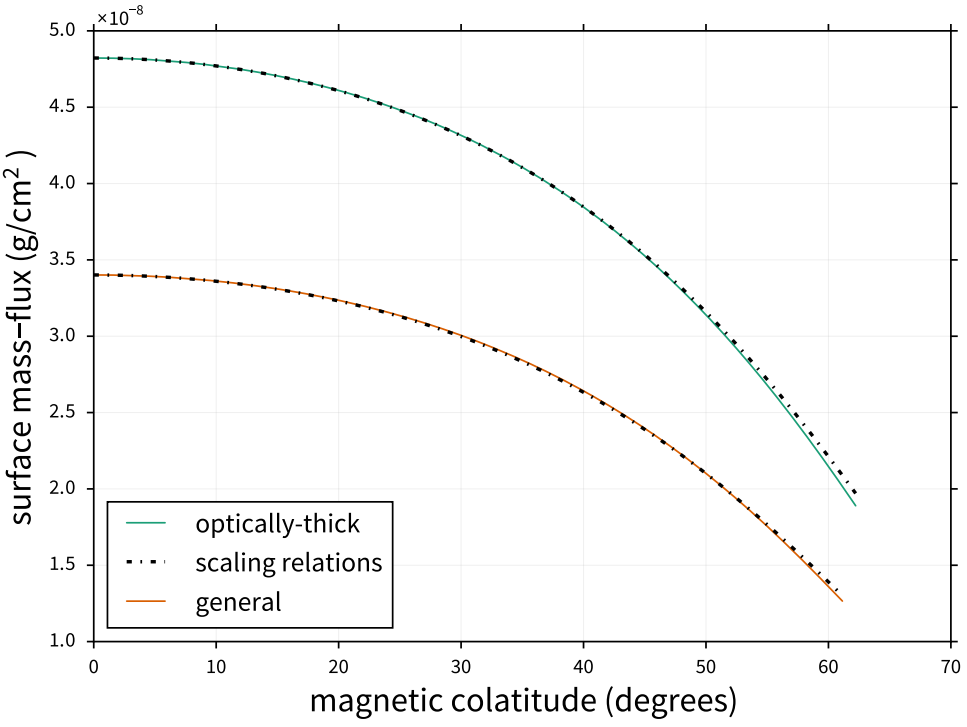


Figure 5.4 : Surface mass-flux ($\dot{m}$) as a function of surface colatitude calculated using both the general and optically-thick line acceleration for a B-type star (Table 5.1). $\dot{m}$, as calculated by the steady-state model (solid line), is compared to the derived scaling relations (dot-dashed) of Equation 5.51 and Equation 5.61. Similar results for the O star not shown.

5.5.4 General case

In the general case, we add a finite-disk "optically-thin correction" parameter, Σ_f, to our optically-thick result (Equation 5.52). First, we consider the effect of the fdcf on χ, evaluated at the critical point (derived in Appendix C):

$$\chi_f = \frac{\chi_c}{f} + \frac{y}{f}\frac{\partial f}{\partial y}\frac{|y|}{(1-\alpha)c\kappa_e\bar{Q}\dot{m}_*\frac{A_*}{A}}\left[\left(1+\frac{c\kappa_e\bar{Q}\dot{m}_*\frac{A_*}{A}}{|y|}\right)^{1-\alpha}-1\right], \tag{5.55}$$

with the non-fdcf critical value of χ for a magnetic dipole defined as

$$\chi_c = -g_{\mathrm{eff},s}r^2/[\bar{Q}\Gamma_{\mathrm{el}}GM_*\psi] = \chi_0\aleph. \tag{5.56}$$

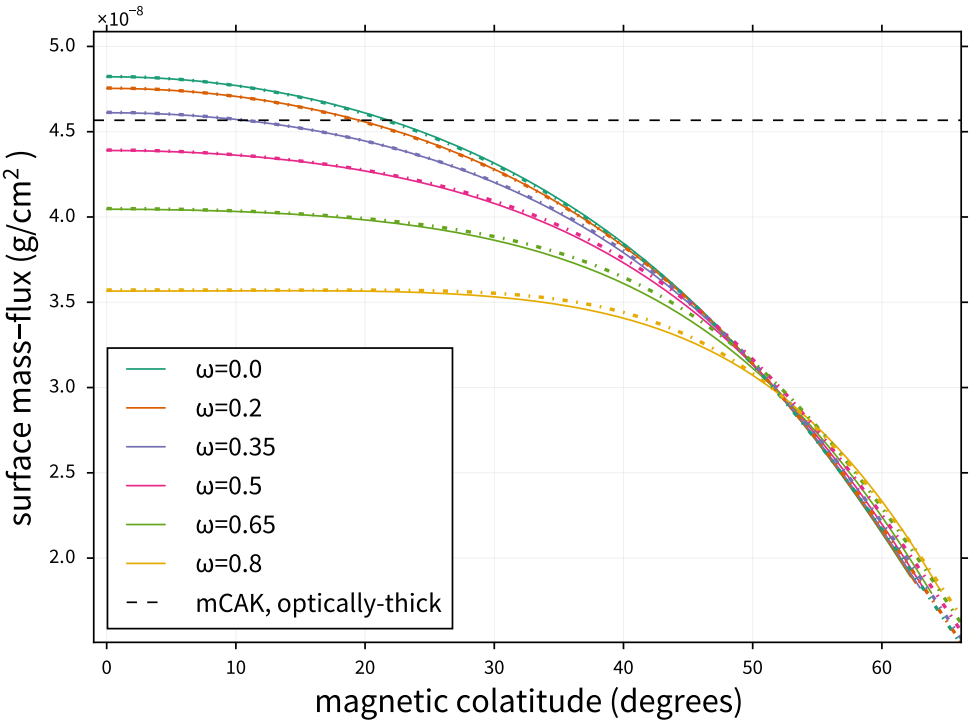


Figure 5.5 : Surface mass-flux ($\dot{m}$) as a function of surface colatitude calculated using the optically-thick line acceleration for a B-type star (Table 5.1). $\dot{m}$, as calculated by the steady-state model (solid line), is compared to the derived scaling relations (dot-dashed) of Equation 5.54 and the mCAK mass-flux (dashed). f_{ratio} in the scaling equations is given by the fits of Equation 5.46 and Table 5.2. Similar results for the O star not shown.

The right-hand term of Equation 5.55 is of order ten times smaller than the left-hand term, so we neglect it and take $\chi_f \approx \chi_c/f$. As shown in Appendix D, this results in an optically-thin correction parameter

$$\Sigma_f = \frac{|1-\alpha-\left[\frac{1-(\chi_0 \aleph/f_c)^{1/\alpha-1}}{1-(\chi_0 \aleph/f_c)^{1/\alpha}}\right]|}{\alpha} \tag{5.57}$$

such that our critical y is

$$y_c \approx -\frac{\alpha}{1-\alpha}\Sigma_f g_{\mathrm{eff},s} \tag{5.58}$$

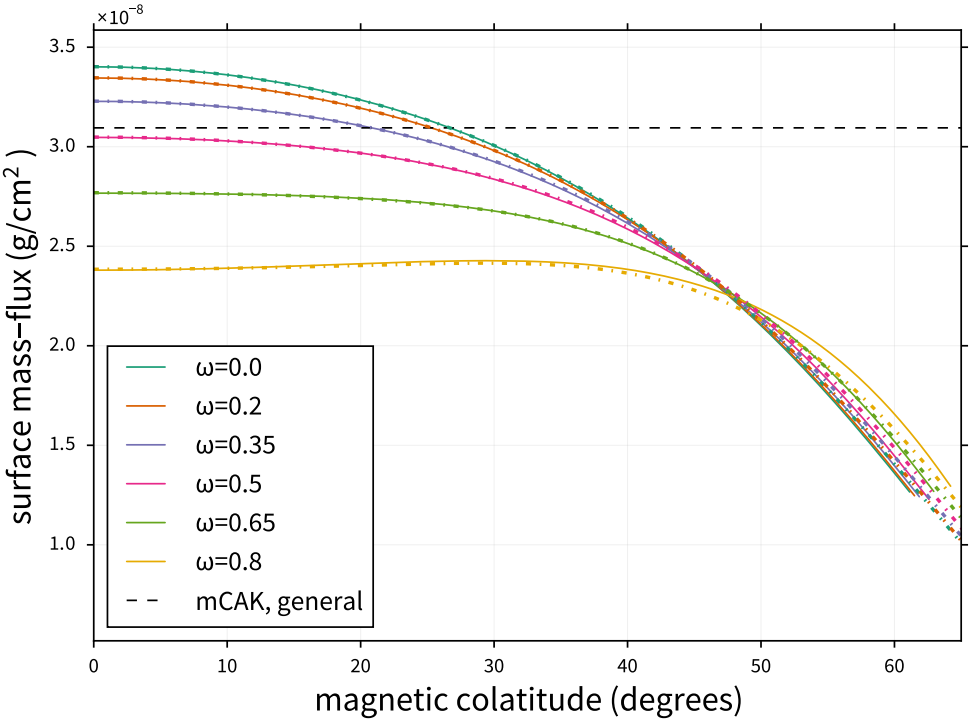


Figure 5.6 : Surface mass-flux ($\dot{m}$) as a function of surface colatitude calculated using the general line acceleration for a B-type star (Table 5.1). $\dot{m}$, as calculated by the steady-state model (solid line), is compared to the derived scaling relations (dot-dashed) of Equation 5.65 and the mCAK mass-flux (dashed). f_{ratio} in the scaling equations is given by the fits of Equation 5.46 and Table 5.2. Similar results for the O star not shown.

after making suitable approximations (details in Appendix D). Following some math, we get the mass-flux scaling

$$\dot{m}_* \approx f_c^{1/\alpha} \Sigma_f \aleph^{1-1/\alpha} \dot{m}_{\rm dip} \left(\frac{R_*}{R_p}\right)^3 \psi_c \sqrt{\frac{1+3\cos^2\theta_m}{1+3\cos^2\theta_c}}. \tag{5.59}$$

Without rotation ($\aleph = 1$, $R_* = R_p$), we can simplify this using the same approximations as the zero-rotation, optically-thick case above:

$$\dot{m}_* \approx \mu_B (\mu_B f_{\theta=0})^{1/\alpha} \Sigma_{f,0} \dot{m}_{\rm dip}. \tag{5.60}$$

We can correct for the approximations made in deriving Equation 5.58 by taking the model-calculated $\dot{m}_{{\rm gen},\theta=0}$ instead of $f_{\theta=0}^{1/\alpha} \Sigma_{f,0,\theta=0} \dot{m}_{\rm dip}$:

$$\dot{m}_* \sim \mu_B^{1+1/\alpha} \frac{\Sigma_{f,0}}{\Sigma_{f,0,\theta=0}} \dot{m}_{{\rm gen},\theta=0}, \tag{5.61}$$

with a non-rotating, fdcf optically-thin correction

$$\Sigma_{f,0} \equiv \frac{\left| 1 - \alpha - \left[\frac{1-\left(\frac{\chi_0}{\mu_B f_{\theta=0}}\right)^{1/\alpha-1}}{1-\left(\frac{\chi_0}{\mu_B f_{\theta=0}}\right)^{1/\alpha}} \right] \right|}{\alpha} \tag{5.62}$$

and $\Sigma_{f,0,\theta=0}$ the polar value of $\Sigma_{f,0}$ (with $\mu_B = 1$). Figure 5.4 shows the excellent agreement of this scaling relation with the model calculation.

With rotation, simplifying Equation 5.59 yields

$$\dot{m} \approx \mu_B (f_{\rm ratio} f_{\theta=0})^{1/\alpha} \Sigma_{f,m} \aleph_m^{1-1/\alpha} \dot{m}_{\rm dip}, \tag{5.63}$$

with the surface optically-thin-correction parameter

$$\Sigma_{f,m} \equiv \frac{\left| 1 - \alpha - \left[\frac{1-\left(\frac{\chi_0 \aleph_m}{f_{\rm ratio} f_{\theta=0}}\right)^{1/\alpha-1}}{1-\left(\frac{\chi_0 \aleph_m}{f_{\rm ratio} f_{\theta=0}}\right)^{1/\alpha}} \right] \right|}{\alpha}. \tag{5.64}$$

We correct for simplifications made in deriving the above scaling relation by taking $\dot{m}_{{\rm gen},\theta=0}$ as the polar mass-flux:

$$\dot{m} \sim \mu_B f_{\rm ratio}^{1/\alpha} \frac{\Sigma_{f,m}}{\Sigma_{f,m,\theta=0}} \aleph_m^{1-1/\alpha} \dot{m}_{{\rm gen},\theta=0}. \tag{5.65}$$

Figure 5.6 shows excellent agreement between this scaling relation and the model-calculated surface mass-flux.

5.6 Velocity Structure

In Paper I, we found that the faster-than-spherical divergence of the dipole magnetic field leads to higher terminal wind velocities (v_∞) for magnetic stars compared to their non-magnetic counterparts. Additionally, using the general form of the radiative acceleration rather than the optically-thick form reduces the terminal velocity, though the overall result for a magnetic dipole is a higher v_∞. Defining the escape velocity $v_{\rm esc}^2 = 2(1-\Gamma_{\rm el})GM_*/R_p$, we find the polar terminal velocity in the general case for a B star (O star) is $v_\infty \approx 2.30\ v_{\rm esc}$ ($v_\infty \approx 7.59\ v_{\rm esc}$). These values are larger than what we calculated for the point-star case: $v_\infty \approx 1.97\ v_{\rm esc}$ (5.90 $v_{\rm esc}$) for our B-type (O-type) star. This is expected since the finite-disk correction factor lowers the overall wind density, which leads to much larger accelerations

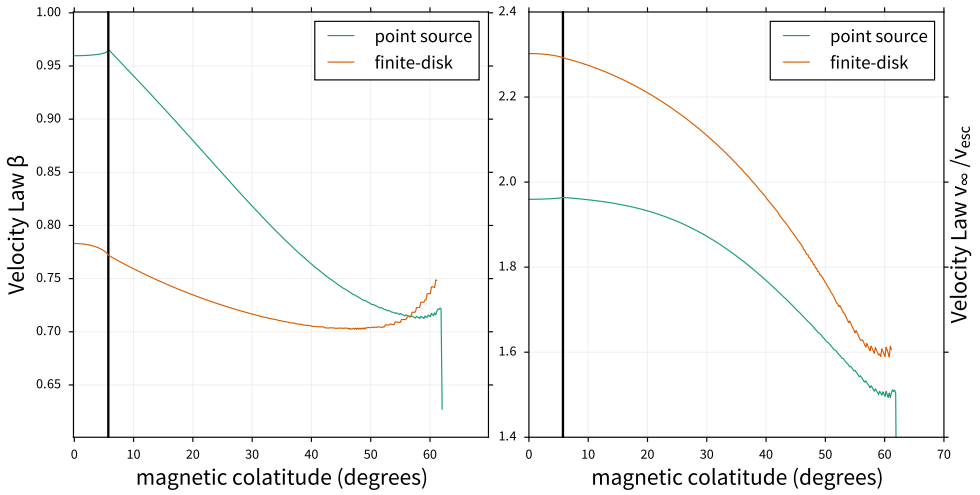


Figure 5.7 : Best-fit velocity-law betas (left) and corresponding terminal velocities (right) for our non-rotating B star model, with (dot-dashed) and without (solid) the finite-disk correction. The solid black line represents the surface colatitude for $L = 100\ R_p$, our truncation radius; results from more polar colatitudes should be ignored since those model lines were truncated well before reaching an apex.

beyond the critical point (Friend & Abbott 1986; Pauldrach et al. 1986). For comparison, the terminal velocities we calculate for the spherical, optically-thin, finite-disk winds are $v_\infty \approx 1.74\ v_{\rm esc}$ for the B star and $v_\infty \approx 2.94\ v_{\rm esc}$ for the O star. These fit well with observed terminal velocities of $v_\infty \approx 1.4 v_{\rm esc}$ for B stars and $v_\infty \approx 2-3\ v_{\rm esc}$ for O stars (Kudritzki & Puls 2000).

Another result of Paper I was the finding that the magnetospheric velocity structure cannot be fit with an universal beta-velocity law of the form $v = v_\infty(1 - R_*/r)^\beta$. The centrifugal force induced by co-rotation with the star may prevent the wind velocity from having an asymptotic limit (v_∞)! For high rotation, there is no well-fit beta-velocity law, though at zero rotation, each line can be fit with individual parameters $v_{\infty,i}$ and β_i (Figure 4.10). The unique magnetic tilt of each field line changes how the wind accelerates within each line; this leads to different values for the best-fit velocity law parameters across the magnetosphere.

As in Paper I, we fit beta-velocity laws to each field line's velocity structure using nonlinear least-squares. Figure 5.7 and Figure 5.8 show best-fit beta-velocity law

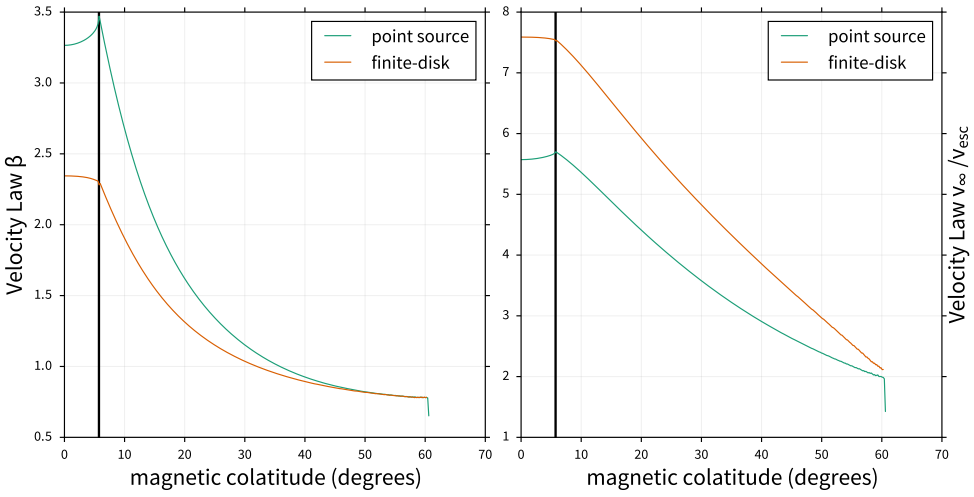


Figure 5.8 : Same as Figure 5.7, but for the non-rotating O star model.

parameters at zero rotation for the point-star and fdcf cases. In both cases, the magnetic tilt has a similar effect: the terminal velocity and how quickly the wind reaches this limit both decrease with increasing magnetic tilt. As one moves towards the pole, the line v_∞ approaches the polar v_∞. Adding the fdcf raises the overall terminal velocity as well as decreasing the β, though the results suggest that significant magnetic tilt ($\mu_B \lesssim 0.8$ at $\theta \gtrsim 55°$) will neuter the wind acceleration more than in the point star case.

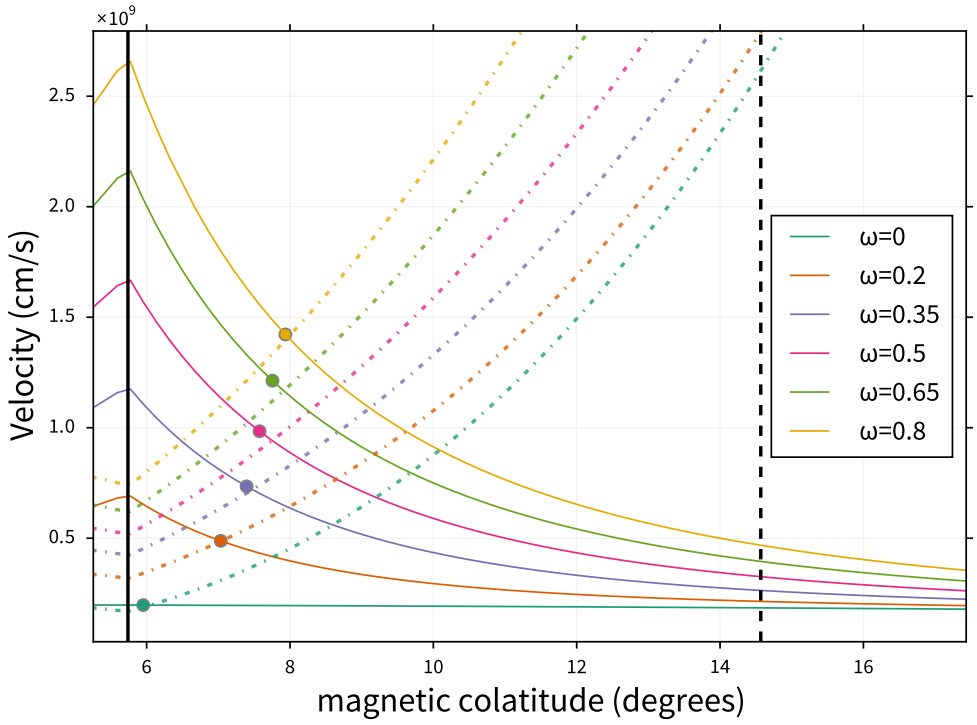


Figure 5.9 : Comparison of wind velocity (solid) vs. Alfvén velocity (dot-dashed) at apex of each magnetic field line for our optically-thin corrected B star model. Colored dots indicate approximation of closure colatitude for each rotation rate. The dashed line indicates the non-rotating MHD-predicted closure colatitude. The solid black line represents the surface colatitude for $L = 100\ R_p$, our truncation radius; results from more polar colatitudes should be ignored since those model lines were truncated well before reaching an apex. Similar results for the O star are not shown.

5.7 Closure Radius

Following Paper I, we estimate the extent of the magnetosphere with a closure radius scaling derived from MHD simulations (ud-Doula & Owocki 2002):

$$R_c \approx R_p + 0.7[R_p(0.3 + \eta_*^{1/4}) - R_p], \tag{5.66}$$

with the usual "wind magnetic confinement parameter" (ud-Doula & Owocki 2002)

$$\eta_* \equiv \frac{B_{\rm eq}^2 R_*^2}{\dot{M}_{B=0} v_{\infty,B=0}}, \tag{5.67}$$

taking $\dot{M}_{B=0}$ and $v_{\infty,B=0}$ as the non-magnetic values for the stellar mass-loss and wind terminal velocity, respectively. We use the same values of η_* as calculated in Paper I (Table 5.1).

By comparing the velocity at the apex of each field line with the local Alfvén velocity, $v_A = B_0/\sqrt{4\pi\rho}$, we can estimate whether or not the field line is able to keep the wind confined. A field line is considered "open" if the apex velocity exceeds the apex Alfvén velocity. Thus, we take R_c as the field line shell parameter, L, at which $v_A = v_{\rm apex}$. Our estimates will differ from MHD-derived closure radii scalings (ud-Doula & Owocki 2002) since we do not allow for the possibility of field lines being stretched out by the wind. If we allow for wind-stretched field lines, the effects of magnetic tilt will be reduced and the

Table 5.3 : Estimated closure radii (in units R_p) for the model B star centrifugal magnetosphere and O star dynamical magnetosphere at different rotation rates. The MHD scaling estimate (Equation 5.66) and the point-star model estimates ("point-star") are also included for comparison.

Type	$\omega = 0.0$	0.2	0.35	0.5	0.65	0.8	MHD
B	93.	66.7	60.4	57.6	55.0	52.6	15.8
B, point-star	74.1	54.9	48.0	44.2	42.4	39.2	18.4
O	10.6	10.6	10.5	10.5	10.3	10.3	4.7
O, point star	5.02	4.96	4.91	4.87	4.78	4.65	2.72

resulting faster wind will push the closure radius further in. Thus, our model estimates for the closure radii can be thought of as an upper bound, with the MHD estimates a lower bound.

Our results demonstrate that, although the finite-disk effect boosts the velocity in the wind, the lower density in the wind also boosts the Alfvén velocity. The overall effect is to push the estimated closure radii further out compared to the point-star model's closure radii (Table 5.3). These results suggest that the magnetic field may be better able to retain the wind in the finite-disk case than the point-star magnetosphere case. We note that ud-Doula & Owocki (2002) used the optically-thick form of the radiative driving in their simulations, so it is possible that their closure radii estimates underestimate the overall extent of the magnetosphere.

Table 5.4 : Mass-loss rates (in units of $10^{-9}\ M_\odot/\,\mathrm{yr}$) for a B-type star ($\eta_* = 4.28 \times 10^5$) as calculated from our model, assuming infinite magnetic confinement where applicable. "No B" indicates a mCAK-type mass-loss rate calculated from a non-rotating radial flow with spherical divergence. The other mass-loss rates are calculated from a dipole magnetosphere with the given rotation fraction ω. "Optically-Thick" indicates the mass-loss calculated from using the optically-thick $g_{\mathrm{rad},s}$; the rest use the general $g_{\mathrm{rad},s}$. "Open" is the mass-loss into open field lines ($L > R_c$), R_c given by the MHD-estimated value (Equation 5.66). "Disk" is the mass-loss into field lines with a centrifugally supported disk ($R_K < L < R_c$). Numbers in parentheses next to a mass-loss rate represent the ratio of that particular rate to the "General" mass-loss at its rotation fraction ω.

	No B	$\omega = 0.0$	0.2	0.35	0.5	0.65	0.8
Optically-Thick	0.89	0.373	0.373	0.372	0.371	0.365	0.349
General	0.605	0.253	0.253	0.252	0.252	0.248	0.236
Open	...	0.021(0.08)	0.020(0.08)	0.020(0.08)	0.019(0.075)	0.017(0.07)	0.015(0.06)
Disk	...	...	0.090(0.36)	0.138 (0.54)	0.178(0.7)	0.208(0.84)	0.221(0.93)
Effective	...	0.021(0.08)	0.11(0.44)	0.157(0.62)	0.196(0.77)	0.225(0.91)	0.236(0.999)

5.8 Global Mass-Loss

In our model, the global mass-loss rate $\dot{M}_{\rm global}$, is found by integrating the mass-flux over the stellar surface:

$$\dot{M}_{\rm global} = \int \dot{m}_r \; dA = 2\pi \int R_*^2 \; \mu_B \; \dot{m}_* d\mu_m \; , \tag{5.68}$$

with $\mu_m = \cos\theta_m$. For the non-magnetic case, we simply take $\dot{M}_{\rm global} = 4\pi R_p^2 \dot{m}_{B=0}$. Following Paper I, we also define "true", "disk", and "effective" mass-loss rates based on the behavior of the wind after it flows away from the stellar surface. The "effective" mass-loss is simply the mass lost to wind flowing along open field lines out of the magnetosphere ("true") or into a centrifugally-supported disk ("disk"). Plasma which does not escape the magnetosphere nor settles into a disk eventually falls back to the stellar surface (ud-Doula & Owocki 2002), so it is never "lost". We define open field lines as having $L > R_c$; we use the MHD-derived closure radii (Equation 5.66). Lines with a centrifugally-supported disk are defined with $L > R_K$, where the Kepler radius (e.g. ud-Doula et al. 2008)

$$R_K = \frac{GM_*}{v_\phi^2} = \omega^{-2/3} R_p, \tag{5.69}$$

demarcates the beginning of the centrifugally-supported disk. We assume that our various scaling relations derived in Section 5.5 hold for the entire stellar surface.

The results for our model stars are presented in Table 5.4 and Table 5.5. We get similar results as the point-star case, namely that a dipole magnetic field greatly reduces the overall mass-loss rate of a massive star. Additionally, we can roughly approximate the general mass-loss rates by multiplying the optically-thick mass-loss rates by the corresponding Σ_f parameter at a particular rotation rate, though there is a greater difference than in the point-star case. The increased error is due to the scaling of the finite-disk-correction with surface colatitude; the optically-thin-correction Σ_f is not constant over the surface as it was in the point-star case.

The relationship between mass-loss and rotation, however, changes when we correct for the finite disk of the star. As Paper I showed, rotation in the point-star model boosted the surface mass-flux and overall mass-loss rates. However, with the finite disk effect, the stellar mass-loss stays roughly the same! This is caused by the reduced magnitude of the finite-disk correction as rotation increases (Figure 5.3), which is itself caused by the reduction in stellar surface intensity for an oblate star (Equation 5.14). Though stellar rotation still wants to boost the mass-loss rate, the effect of a lower surface intensity cancels out this effect and even overtakes it at the highest rotation rates.

In the point-star case (Paper I), the effective mass-loss was found to be 20% (at $\omega = 0.2$) to 65% (at $\omega = 0.8$) of the non-rotating, non-magnetic CAK mass-loss rate. Here, with the finite-disk, we find effective mass-losses ranging from about 20% to 40% of the mCAK value. Since the general mass-loss roughly stays constant with rotation, the effective mass-loss rate is essentially set by the size of the centrifugal disk (parameterized here by R_K) for a given star.

5.9 Conclusion

In this paper, we added the finite-disk correction factor to our steady-state analysis of the steady-state Arbitrary Rigid-Field Hydrodynamics Equations. To simplify our analysis,

Table 5.5 : Same as Table 5.4, except for an O-type star with $\eta_* = 100$. All mass-loss rates are given in 10^{-6} $M_\odot$/yr. Numbers in parentheses next to a mass-loss rate represent the ratio of that particular rate to the "General" mass-loss with the same rotation.

	No B	$\omega = 0.0$	0.2	0.35	0.5	0.65	0.8
Optically-Thick	3.34	1.50	1.50	1.50	1.49	1.46	1.39
General	3.26	1.46	1.46	1.46	1.45	1.42	1.35
Open	...	0.40(0.27)	0.36(0.25)	0.37(0.25)	0.34(0.24)	0.34(0.24)	0.32(0.24)
Disk	...	...	0.25(0.17)	0.51 (0.35)	0.76 (0.52)	0.94 (0.66)	1.03 (0.76)
Effective	...	0.40(0.27)	0.62(0.42)	0.88(0.60)	1.10(0.76)	1.28(0.9)	1.35 (0.999)

we assumed that the line-of-sight velocity gradient for each ray tracing the stellar surface from any point in the wind could be approximated with only its radial component. After deriving new critical point values, we confirmed that they reduced to the point-star values in the proper limit. With these critical values in hand, we found critical point locations for an aligned dipole magnetic field in order to calculate steady-state mass-fluxes. We then derived analytic scaling relations for the surface mass-flux with stellar colatitude and used them to obtain global mass-loss rates. The velocity structure of the finite-disk-corrected magnetosphere was obtained by integrating from the critical point locations to the apex of each line.

The key results are summarized as follows:

1. For an optically-thick wind, the finite-disk correction (f) can simply be multiplied to the point-star mass-flux scaling relation. At zero rotation, we obtain a scaling $\dot{m} \sim \mu_B^{1+1/\alpha} \dot{m}_{\theta=0}$. With rotation, the relationship between f and surface colatitude (and the resulting scaling relation) becomes too complex for an analytic function.

2. The optically-thin-correction parameter Σ_f depends on the finite-disk correction. For O stars, their high mass-losses mean that f does not have a large effect on Σ_f compared to the point-star case. However, for B stars, f further decreases the wind density; as a result, Σ_f decreases by another 10% from ≈ 0.72 to 0.65. This results in an overall reduction of about 35% in B star mass-loss rates compared to their optically-thick mass-loss rate.

3. We reproduce the Paper I result that the magnetosphere does not have a global beta-velocity law; each line has an unique velocity structure. Additionally, with rotation, these velocity structures do not follow a beta-velocity law. Compared to the point-star case, the finite-disk factor is found to increase the polar terminal velocity and how

quickly the wind reaches this limit (represented by a decreasing β in the zero-rotation velocity law fits).

4. The closure radius is found to move further out, despite the higher wind terminal velocities for each line.

5. Unlike the point-star case, global mass-loss rates are found to stay roughly constant with rotation (and decrease at the highest rotation rates). This is caused by a decrease in the stellar surface intensity as rotation increases, keeping luminosity fixed. The "effective" mass-loss rate, in which plasma does not fall back to the star, is found to be about 20% (at $\omega = 0.2$) to 40% (at $\omega = 0.8$) of the non-rotating, non-magnetic mCAK mass-loss rate.

In sum, we have expanded our understanding of how a dipole magnetic field affects the line-driven wind of a massive star. These advances will further our understanding of magnetic massive star observations, especially their X-ray emission. This will also have implications for massive star evolution, as the amount of mass lost during a massive star's lifetime has a major effect on the end product of evolution (Maeder & Meynet 2012). Changing the mass-loss rate changes what will happen to the star.

References

Abbott, D. C. 1980, ApJ, 242, 1183

Bjorkman, J. E. 1995, ApJ, 453, 369

Castor, J. I., Abbott, D. C., & Klein, R. I. 1975, ApJ, 195, 157

Cranmer, S. R. 1996, PhD thesis, Bartol Research Institute, University of Delaware

Cranmer, S. R., & Owocki, S. P. 1995, ApJ, 440, 308

Dennis, J. E., & Schnabel, R. B. 1983, Numerical methods for unconstrained optimization and nonlinear equations

Drew, J. E. 1989, ApJS, 71, 267

Friend, D. B., & Abbott, D. C. 1986, ApJ, 311, 701

Gayley, K. G. 1995, ApJ, 454, 410

Kudritzki, R. P. 2002, ApJ, 577, 389

Kudritzki, R. P., Pauldrach, A., Puls, J., & Abbott, D. C. 1989, A&A, 219, 205

Kudritzki, R.-P., & Puls, J. 2000, ARA&A, 38, 613

Madura, T. I., Owocki, S. P., & Feldmeier, A. 2007, ApJ, 660, 687

Maeder, A., & Meynet, G. 2012, Reviews of Modern Physics, 84, 25

Müller, P. E., & Vink, J. S. 2008, A&A, 492, 493

Owocki, S. P., Castor, J. I., & Rybicki, G. B. 1988, ApJ, 335, 914

Pauldrach, A., Puls, J., & Kudritzki, R. P. 1986, A&A, 164, 86

Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. 2002, Numerical recipes in C++ : the art of scientific computing

Puls, J., Springmann, U., & Lennon, M. 2000, A&AS, 141, 23

Sundqvist, J. O., & Owocki, S. P. 2015, MNRAS, 453, 3428

Townsend, R. H. D., Owocki, S. P., & Ud-Doula, A. 2007, MNRAS, 382, 139

ud-Doula, A., Owocki, S., Townsend, R., Petit, V., & Cohen, D. 2014, MNRAS, 441, 3600

ud-Doula, A., & Owocki, S. P. 2002, ApJ, 576, 413

ud-Doula, A., Owocki, S. P., & Townsend, R. H. D. 2008, MNRAS, 385, 97

A Optically-Thick Critical Values

Before we begin, we mention that, unfortunately, we cannot disentangle the spatial components from the non-spatial components in the critical value derivation as well as we did in Paper I. The finite-disk correction factor and its derivatives make this practically impossible. In our derivation below, we intentionally parallel Paper I's analysis and definitions to present the clearest picture of how the fdcf affects our analysis.

We start with the equation of motion (Equation 5.24),

$$F[s, y, u] \equiv y(1 - 1/u^2) - g_{\text{eff},s} - f\Delta\Gamma|y|^\alpha, \tag{A1}$$

where, for mathematical simplicity, we consolidate the spatial terms into $\Gamma \equiv (A/A_*)^\alpha\, \psi/r^2$. We evaluate the CAK singularity and regularity conditions at the critical point, remembering that $f = f(s, y, u)$:

$$0 = \frac{\partial F}{\partial y} = (1 - 1/u_c^2) - \alpha f_c \Delta_c \Gamma \frac{|y_c|^\alpha}{y_c} - \frac{\partial f}{\partial y}\Delta_c\Gamma|y_c|^\alpha \tag{A2}$$

and

$$0 = \frac{\partial F}{\partial s} + \frac{y_c}{c_s^2 u_c}\frac{\partial F}{\partial u} = -\frac{\partial g_{\text{eff},s}}{\partial s} - \frac{\partial f}{\partial s}\Delta_c\Gamma|y_c|^\alpha - f_c\Delta_c\frac{\partial\Gamma}{\partial s}|y_c|^\alpha + \frac{y_c}{c_s^2 u_c}\left[\frac{2y_c}{u_c^3} - \Delta_c\Gamma|y_c|^\alpha\frac{\partial f}{\partial u}\right] \tag{A3}$$

Substituting Equation A2 into the equation of motion yields both

$$-g_{\text{eff},s} = f_c\Delta_c\Gamma|y_c|^\alpha\left(1 - \alpha - \frac{y_c}{f_c}\frac{\partial f}{\partial y}\right) \tag{A4}$$

and

$$y_c(1 - 1/u_c^2) = -g_{\text{eff},s}\,\frac{\alpha + \frac{y_c}{f_c}\frac{\partial f}{\partial y}}{1 - \alpha - \frac{y_c}{f_c}\frac{\partial f}{\partial y}}. \tag{A5}$$

Simplifying the regularity condition (Equation A3) with Equation A4 gives us

$$\frac{2y_c^2}{c_s^2 u_c^4} = \frac{\partial g_{\text{eff},s}}{\partial s} - \frac{g_{\text{eff},s}}{1 - \alpha - \frac{y_c}{f_c}\frac{\partial f}{\partial y}}\left[\frac{1}{\Gamma}\frac{\partial\Gamma}{\partial s} + \frac{1}{f_c}\frac{\partial f}{\partial s} + \frac{y_c}{c_s^2 u_c f_c}\frac{\partial f}{\partial s}\right]. \tag{A6}$$

Remembering our definition of Γ, we have

$$\frac{1}{\Gamma}\frac{\partial\Gamma}{\partial s} = \alpha\lambda + \frac{1}{\psi}\frac{\partial\psi}{\partial s} - \frac{2\psi}{r}, \tag{A7}$$

and we can define

$$\Phi_f \equiv \frac{2y_c^2}{c_s^2 u_c^4} = \frac{\partial g_{\text{eff},s}}{\partial s} - \frac{g_{\text{eff},s}}{1 - \alpha - \frac{y_c}{f_c}\frac{\partial f}{\partial y}}\left[\alpha\lambda + \frac{1}{\psi_c}\frac{\partial\psi}{\partial s} - \frac{2\psi_c}{r_c} + \frac{1}{f_c}\frac{\partial f}{\partial s} + \frac{y_c}{c_s^2 u_c f_c}\frac{\partial f}{\partial u}\right], \tag{A8}$$

which is Equation 5.31. $\sqrt{\Phi_f^2}$ has two possible roots: $y_c/u_c^2 = \pm c_s\Phi/\sqrt{2}$. In our sign convention, $y = vv'$ can be either positive or negative depending on the direction of flow. For an increasing arc length away from the stellar surface ($\psi > 0$), an accelerating outflow has $v > 0$, $dv > 0$, and $ds > 0$, resulting in $y_c > 0$. At the opposite line footprint ($\psi < 0$), if there is one, an accelerating outflow requires $v < 0$, $dv < 0$, and $ds < 0$, resulting in $y_c < 0$. Thus, we take the positive (negative) value of $y_c/u_c^2 = \sqrt{\Phi_f^2}$ for positive (negative) ψ. Using Φ_f^2 in Equation A5 gives us our critical y and u^2:

$$y_c = \pm\frac{c_s\Phi_f}{\sqrt{2}} - \frac{\alpha - \frac{y_c}{f_c}\frac{\partial f}{\partial y}}{1 - \alpha - \frac{y_c}{f_c}\frac{\partial f}{\partial y}}\, g_{\mathrm{eff},s} \tag{A9}$$

$$u_c^2 = 1 \mp \frac{\sqrt{2}g_{\mathrm{eff},s}}{c_s\Phi_f}\frac{\alpha - \frac{y_c}{f_c}\frac{\partial f}{\partial y}}{1 - \alpha - \frac{y_c}{f_c}\frac{\partial f}{\partial y}}. \tag{A10}$$

It is then easy to solve for our critical Δ

$$\Delta_c = \frac{-g_{\mathrm{eff},s}}{1 - \alpha - \frac{y_c}{f_c}\frac{\partial f}{\partial y}}\frac{r_c^2}{f_c\psi|y_c|^\alpha(A/A_*)^\alpha} \tag{A11}$$

and our resulting surface mass-flux

$$\dot{m}_* = \frac{L_*}{4\pi c^2}\left(\bar{Q}\Gamma_{\mathrm{el}}GM_*\right)^{\frac{1-\alpha}{\alpha}}\left(\frac{1}{(1-\alpha)\Delta_c}\right)^{1/\alpha}. \tag{A12}$$

In all cases, it is easily shown that the results of Paper I are obtained by taking $f = 1$ and its derivatives as zero.

B Equation of Motion, General Case

We follow the same procedure as the optically-thick case and eliminate the density from the general $g_{\mathrm{rad},s}$ (Equation 5.6) using the continuity equation (Equation 5.10):

$$g_{\mathrm{rad},s} = f\frac{\bar{Q}\Gamma_{\mathrm{el}}GM_*}{1-\alpha}\frac{\psi}{r^2}\frac{|vv'|}{\xi}\left[(1 + \xi/|vv'|)^{1-\alpha} - 1\right], \tag{B13}$$

where we define an eigenvalue

$$\xi \equiv c\kappa_e\bar{Q}\dot{m}_*\frac{A_*}{A}. \tag{B14}$$

Our equation of motion is then

$$F_{\rm sat} \equiv vv' \left(1 - c_{\rm s}^2/v^2\right) - g_{{\rm eff},s} \\ - f\frac{\bar{Q}\Gamma_{\rm el}GM_*}{1-\alpha}\frac{\psi}{r^2}\frac{|vv'|}{\xi}\left[(1+\xi/|vv'|)^{1-\alpha} - 1\right] = 0, \tag{B15}$$

which is identical to the non-finite-disk-corrected version except for the addition of f, the correction factor.

C General Critical Values

As in Appendix A, we will solve for $(y, u_c^2, \dot{m}_*)$ in a similar manner to Paper I. In the general equation of motion (Equation B15), we make the substitutions $y = vv'$ and $u = v/c_s$. Additionally, to slightly simplify the math, we define $\bar{\Gamma} = \psi/r^2$, resulting in

$$F_{\rm sat}[s, y, u] = y\left(1 - 1/u^2\right) - g_{{\rm eff},s} \\ - f\frac{\bar{Q}\Gamma_{\rm el}GM_*\bar{\Gamma}}{1-\alpha}\frac{|y|}{\xi}\left[(1+\xi/|y|)^{1-\alpha} - 1\right] = 0. \tag{C16}$$

After evaluating the CAK singularity condition and simplifying, we obtain

$$y_c(1 - 1/u_c^2) = f_c\frac{\bar{Q}\Gamma_{\rm el}GM_*}{1-\alpha}\frac{\bar{\Gamma}}{\xi_c}|y_c|\left[\left(1+\frac{\xi_c}{|y_c|}\right)^{1-\alpha} - 1\right]\left[1 + \frac{y_c}{f_c}\frac{\partial f}{\partial y}\right] \\ - f_c\bar{Q}\Gamma_{\rm el}GM_*\bar{\Gamma}(1+\xi_c/|y_c|)^{-\alpha} \tag{C17}$$

Substituting this into Equation C16 and simplifying yields

$$(1+\xi_c/|y_c|)^{-\alpha} = \frac{-g_{{\rm eff},s}}{f_c\bar{Q}\Gamma_{\rm el}GM_*\bar{\Gamma}} + \frac{y_c}{f_c}\frac{\partial f}{\partial y}\frac{|y_c|}{(1-\alpha)\xi_c}\left[(1+\xi_c/|y_c|)^{1-\alpha} - 1\right]. \tag{C18}$$

In Paper I, we defined a "wind transparency parameter" $\chi \equiv (1+\xi/|y|)^{-\alpha}$ representing the importance of using the general, optically-thin form of $g_{{\rm rad},s}$ (see Section 5.5.1 for discussion). Here, we keep this definition and define χ_f as the critical value of χ in the finite-disk case:

$$\chi_f \equiv \frac{\chi_c}{f} + \frac{y}{f}\frac{\partial f}{\partial y}\frac{|y|}{(1-\alpha)\xi}\left[(1+\xi/|y|)^{1-\alpha} - 1\right], \tag{C19}$$

where χ_c is the non-fdcf critical value for χ,

$$\chi_c = -g_{\mathrm{eff},s} r^2 / [\bar{Q}\Gamma_{\mathrm{el}} G M_* \psi]. \tag{C20}$$

Moving on to the CAK regularity condtion, we evaluate and simplify:

$$\begin{aligned}\frac{2y_c^2}{c_s^2 u_c^4} =& \frac{\partial g_{\mathrm{eff},s}}{\partial s} + f_c \bar{Q}\Gamma_{\mathrm{el}} G M_* \bar{\Gamma} \frac{1}{\xi_c}\frac{\partial \xi}{\partial s} \left(1 + \xi_c/|y_c|\right)^{-\alpha} \\ &+ \frac{f_c \bar{Q}\Gamma_{\mathrm{el}} G M_*}{1-\alpha}\frac{\bar{\Gamma}}{\xi_c}|y_c| \left[(1+\xi_c/|y_c|)^{1-\alpha} - 1\right]\left[\frac{1}{f_c}\frac{\partial f}{\partial s} + \frac{y_c}{c_s^2 u_c f_c}\frac{\partial f}{\partial u} + \frac{1}{\bar{\Gamma}}\frac{\partial \bar{\Gamma}}{\partial s} - \frac{1}{\xi_c}\frac{\partial \xi}{\partial s}\right].\end{aligned} \tag{C21}$$

Using $\partial\xi/\partial s/\xi = -\lambda$ and substituting in χ_f, we obtain

$$\begin{aligned}\frac{2y_c^2}{c_s^2 u_c^4} =& \frac{\partial g_{\mathrm{eff},s}}{\partial s} + \frac{f_c \bar{Q}\Gamma_{\mathrm{el}} G M_*}{1-\alpha}\bar{\Gamma}\chi_f \left[\frac{1-\chi_f^{1/\alpha-1}}{1-\chi_f^{1/\alpha}}\right]\left[\frac{1}{f_c}\frac{\partial f}{\partial s} + \frac{y_c}{c_s^2 u_c f_c}\frac{\partial f}{\partial u} + \frac{1}{\bar{\Gamma}}\frac{\partial \bar{\Gamma}}{\partial s} + \lambda\right] \\ &- f_c \bar{Q}\Gamma_{\mathrm{el}} G M_* \bar{\Gamma}\lambda\chi_f.\end{aligned} \tag{C22}$$

We can then define $\bar{\Phi}_f$ in a parallel manner to the optically-thick case:

$$\begin{aligned}\bar{\Phi}_f^2 \equiv \frac{2y_c^2}{c_s^2 u_c^4} =& \frac{\partial g_{\mathrm{eff},s}}{\partial s} + \frac{f_c \bar{Q}\Gamma_{\mathrm{el}} G M_*}{1-\alpha}\bar{\Gamma}\chi_f \left[\frac{1-\chi_f^{1/\alpha-1}}{1-\chi_f^{1/\alpha}}\right]\left[\frac{1}{f_c}\frac{\partial f}{\partial s} + \frac{y_c}{c_s^2 u_c f_c}\frac{\partial f}{\partial u} + \frac{1}{\bar{\Gamma}}\frac{\partial \bar{\Gamma}}{\partial s}\right] \\ &+ f_c \bar{Q}\Gamma_{\mathrm{el}} G M_* \bar{\Gamma}\lambda\chi_f \left[-1 + \frac{1}{1-\alpha}\frac{1-\chi_f^{1/\alpha-1}}{1-\chi_f^{1/\alpha}}\right].\end{aligned} \tag{C23}$$

Substituting Equation B15 and Equation C23 into Equation C21 gives us a way to simplify for our critical y and u^2:

$$y(1-1/u^2) - g_{\mathrm{eff},s} = \frac{f\bar{Q}\Gamma_{\mathrm{el}} G M_*}{1-\alpha}\bar{\Gamma}\chi_f \left[\frac{1-\chi_f^{1/\alpha-1}}{1-\chi_f^{1/\alpha}}\right], \tag{C24}$$

and we clean up by substituting in $\bar{\Phi}_f = \pm\sqrt{2}y_c/(c_s u_c^2)$:

$$y_c = \pm\frac{c_s\bar{\Phi}_f}{\sqrt{2}} + g_{\mathrm{eff},s} + \frac{f\bar{Q}\Gamma_{\mathrm{el}} G M_*\bar{\Gamma}}{1-\alpha}\chi_f \left[\frac{1-\chi_f^{1/\alpha-1}}{1-\chi_f^{1/\alpha}}\right], \tag{C25}$$

$$u_c^2 = 1 \pm \frac{\sqrt{2}}{c_s\bar{\Phi}_f}\left\{g_{\mathrm{eff},s} + \frac{f\bar{Q}\Gamma_{\mathrm{el}} G M_*\bar{\Gamma}}{1-\alpha}\chi_f \left[\frac{1-\chi_f^{1/\alpha-1}}{1-\chi_f^{1/\alpha}}\right]\right\}. \tag{C26}$$

Finally, we solve for our critical mass-flux using χ_f and ξ:

$$\dot{m}_* = \frac{\left(\chi_f^{-1/\alpha} - 1\right)|y_c|}{c\kappa_e \bar{Q}(A_*/A)}. \tag{C27}$$

It is easily shown that these general-fdcf critical values reduce to the non-fdcf critical values in the limit of $f = 1$ and vanishing derivatives.

As a reminder, χ_f and $\bar{\Phi}_f^2$ are not wholly spatial functions; they have dependencies on y and u through the fdcf and its derivatives. Simultaneous root finding in the same manner as the fdcf optically-thick case (Section 5.4) is required.

D Mass-flux scaling for general case

We reproduce the scaling of Section 5.5.3 for a general line acceleration. First, we note that our correction level parameter (Equation C19) can be simplified by neglecting the fdcf-derivative term (which is $\sim$ 10x smaller than the regular χ_c term), giving us $\chi_f \approx \chi_c/f$. As in the optically-thin case, we neglect $\bar{\Phi}_f$ in simplifying our critical y:

$$\begin{aligned} y_c &\approx g_{\text{eff},s} + \frac{f\bar{Q}\Gamma_{\text{el}} G M_* \bar{\Gamma}}{1-\alpha} \frac{\chi_c}{f} \left[\frac{1-(\chi_c/f)^{1/\alpha - 1}}{1-(\chi_c/f)^{1/\alpha}}\right] \\ &\approx -\frac{\alpha}{1-\alpha} \Sigma_f g_{\text{eff},s}, \end{aligned} \tag{D28}$$

where, remembering that $\chi_c = \chi_0 \aleph$ (Equation 5.56), we define a finite-disk "optically-thin correction" parameter

$$\Sigma_f \equiv \frac{|1 - \alpha - \left[\frac{1-(\chi_o \aleph/f)^{1/\alpha - 1}|}{1-(\chi_o \aleph/f)^{1/\alpha}}\right]}{\alpha}. \tag{D29}$$

The impact of the fdcf depends on the value of χ_0. O stars have a small value of χ_0, so the finite-disk-corrected Σ_f does not differ much from the point-star Σ. However, B stars, with larger values of χ_0, see a much larger reduction in the optically-thin correction parameter. As an example, our non-rotating model B star has $\chi_0 \approx 0.1$ and $f_{\theta=0} \approx 0.7$, leading to $\Sigma \approx 0.725$ and $\Sigma_f \approx 0.65$.

Finally, we simplify the critical mass-flux (Equation C27) for an aligned dipole (using Equation 5.48), noting that $(\chi_c/f_c)^{-1/\alpha} \gg 1$:

$$\begin{aligned}\dot{m}_* &\approx \frac{(\chi_c/f_c)^{-1/\alpha}|y_c|}{c\kappa_e \bar{Q}\frac{A_*}{A}} \\ &\approx f_c^{1/\alpha}\Sigma_f \aleph^{1-1/\alpha}\dot{m}_{\rm dip}\left(\frac{R_*}{R_p}\right)^3 \psi_c \sqrt{\frac{1+3\cos^2\theta_m}{1+3\cos^2\theta_c}}. \end{aligned} \tag{D30}$$

Chapter 6

Updating the XADM model with a magnetic wind

I've been ionized, but I'm ok now.

Buckaroo Banzai

Abstract

The most current model of magnetic massive star X-ray emission, the "X-ray Analytic Dynamical Magnetosphere" (XADM) model, uses non-magnetic values for the assumed stellar mass-loss rate and wind terminal velocity. These, however, do not take into account the effect of the stellar magnetic field on the wind. Using results from a critical point analysis of the steady-state Arbitrary Rigid-Field Hydrodynamic (ARFHD) equations, we update XADM with magnetic values for the mass-loss and terminal velocity. We find that, with its current assumptions, XADM underestimates the X-ray luminosity from early-type massive stars but overestimates it for late-type massive stars. Several implications of this result are discussed, most notably the validity of using the post-shock cooling length to set the size of the X-ray emission region. ARFHD simulations suggest that a centrifugally-supported disk will greatly reduce the cooling length and move the shock location further down the magnetic field line compared to XADM predictions. Ultimately, future X-ray emission models will need to incorporate rotation effects and better calculate the shock location.

6.1 Introduction

X-rays are commonly seen from massive stars, both magnetic and not. Non-magnetic massive stars have been observed to follow a general X-ray luminosity scaling $L_X/L_{\rm bol} \sim 1 \times 10^{-7}$ (Berghoefer et al. 1997; Nazé et al. 2011), though later-type B stars may have a scaling closer to 1×10^{-8} (Cassinelli & Cohen 1994). These scalings are thought to result from embedded shocks caused by flow instabilities in the wind (Lucy & Solomon 1970; Lucy 1982), an idea supported by hydrodynamical simulations (Owocki et al. 1988).

Magnetic massive stars, however, have X-ray luminosities about 10x stronger than their non-magnetic counterparts (Nazé et al. 2014, hereafter N14). To explain this, Babel & Montmerle (1997) developed a Magnetically Confined Wind Shock (MCWS) paradigm. For closed magnetic field loops, the stellar wind flows from both anchored footprints and collides with itself. The result from this collision of two supersonic flows is a strong shock capable of heating the plasma to millions of degrees; the plasma emits X-rays as it cools. ud-Doula et al. (2014) expanded this model with their "X-ray Analytic Dynamical Magnetosphere"(XADM) framework by introducing a "shock-retreat" paradigm in which the size of the X-ray emitting region within the magnetosphere depends on the post-shock cooling length from the shock to the apex of the field line. Eventually, at very high mass-loss rates ($\dot{M} \sim 1 \times 10^{-5}\ M_\odot/\,\mathrm{yr}$), the wind becomes strong enough that the overall size of the magnetosphere shrinks. This leads to a turnover in the X-ray luminosity; this trend is supported by observations (N14; also see Figure 6.1). Unfortunately, neither of these models incorporate the effect of the magnetic field on the wind itself; both use non-magnetic density and velocity structures to estimate the shock energy and resulting X-rays. This is a source of error we will address in this chapter.

In Paper I and Paper II, we took the first steps in understanding the effect of a large-scale magnetic dipole field on a massive-star's line-driven wind. Paper I focused on the simplest case of a point-star, while Paper II relaxed this assumption and correctly incorporated the finite size of the stellar disk. In both cases, the faster-than-spherical divergence of the magnetic field works to boost both the mass-flux and velocity. However, since the field lines force the wind to flow in non-radial directions, the magnetic tension works to reduce the flux and acceleration. The overall result is a faster, more dense magnetic-polar flow and a slower, less dense magnetic-equatorial flow. This change in velocity structure will, when implemented in XADM, produce a different estimate for the overall X-ray luminosity from the magnetosphere.

In this chapter, we discuss the validity of the assumptions made within the XADM

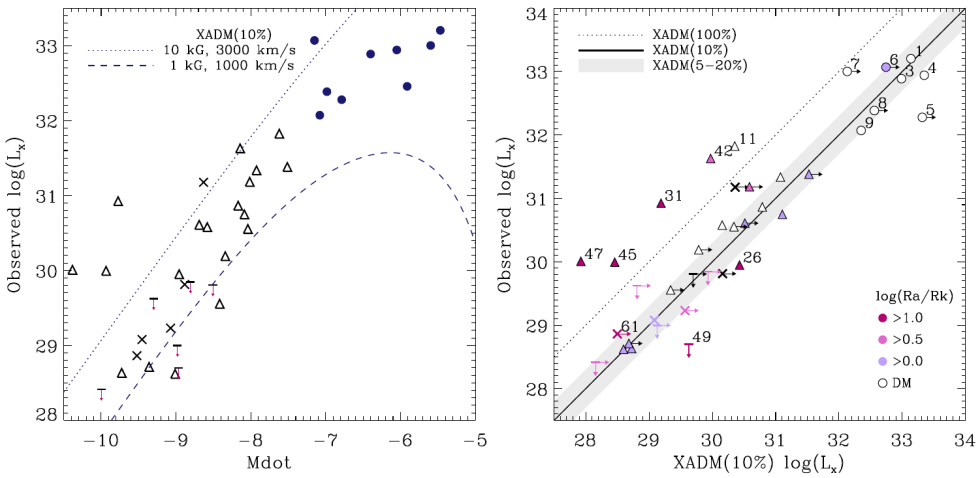


Figure 6.1 : Left: relation between the X-ray luminosity in the 0.5-10.0 keV band and logarithm of mass-loss rate. The dotted and dashed lines represent X-ray luminosities in the same energy band as predicted by XADM (ud-Doula et al. 2014), scaled by 10%, for the given magnetic field and wind terminal velocity. Dots are O stars, triangles are B stars, and crosses are faint detections. Right: Comparison between the observed X-ray luminosity and the predicted values from XADM. The dotted line is 100% efficiency, the solid line is 10%, and the gray area indicates an efficiency range from 5% to 20%. The symbols are color-coded according to the predicted size of their centrifugally-supported regions; darker is bigger, while empty symbols represent lack of centrifugal support. Figure taken from Figure 6 of Nazé et al. (2014); for further details see that paper.

model (Section 6.2) and update it with our Paper II results for a magnetospheric wind (Section 6.3). We talk about implications of our results in Section 6.4 and wrap up in Section 6.5.

6.2 Issues with XADM Assumptions

The velocity has an effect on the level of X-ray emission in the magnetosphere, due to colliding wind shocks (Babel & Montmerle 1997; ud-Doula et al. 2014). The wind speed sets the level of energy available for shock heating and, subsequently, the temperature of the plasma. In their XADM model, ud-Doula et al. (2014) use a simple velocity beta law $v = v_\infty(1 - R_p/r)^\beta \equiv v_\infty w(r)$ to estimate the velocity at the shock location, assuming that each line has the same values of v_∞ and β. This greatly simplifies the analytic calculations and allows the XADM model to depend on only three dimensionless parameters: an X-ray emission/kinetic energy ratio, the closure radius of the magnetosphere, and the XADM terminal speed cooling parameter

$$\chi_\infty \equiv \frac{15\pi}{128} \frac{v_{\infty,\mathrm{unmag}}^4 \, R_p}{\dot{M}_{\mathrm{unmag}} \, \Lambda_m}, \tag{6.1}$$

where $\dot{M}_{\mathrm{unmag}}$ and $v_{\infty,\mathrm{unmag}}$ are the non-magnetic values of the mass-loss rate and terminal velocity. $\Lambda_m \equiv \Lambda/\bar{\mu}_e\bar{\mu}_p$, the mass-weighed optically thin cooling function (c.f. Section 4.2.7), is approximated with a constant $\Lambda = 4.4 \times 10^{-23}$ ergs cm^3 s^{-1} over a shock temperature range $10^{6.5} < T_s < 10^{7.5}$.

There is a factor of 5 difference between XADM and the X-ray emission levels predicted by MHD results, indicating that an efficiency factor of 20% should be applied to the XADM ideal case. N14, in their study of X-ray emission from magnetic massive stars, found that this efficiency factor ranged from 5% - 20%, with a best fit around 10%. While some of this discrepancy most likely results from inefficient X-ray emission caused

by repeated infall of cooled, trapped material in the inner magnetosphere (ud-Doula et al. 2014), some of it may also come from simplifications within the XADM model. In order to study the effect of these simplifications, we consider the same two stars from previous chapters: a B star (parameters from Bjorkman 1995) and an O star (parameters from XADM; also see ud-Doula & Owocki 2002). These parameters are reproduced in Table 6.1 (for discussion see section 4.5).

First, the non-magnetic values for the terminal velocity and mass-loss rate are used, which will provide incorrect estimates for both the amount of mass present in the field line and the velocity at the shock. As we show in Section 4.6, the dipole divergence of magnetic field lines causes a larger-than-CAK polar flux, though there is a similar scaling in the radial mass-flux with μ_B^2 for increasing colatitude. Additionally, XADM uses the optically-thick form of the line-driven acceleration (Castor et al. 1975; hereafter CAK) rather than the general form (Owocki et al. 1988). While this distinction is not so important for early-type massive stars, $\dot{m}_{\rm CAK}$ does overestimate the amount of mass in the field line by about 25% for B-type stars (Section 4.6.2).

The faster-than-spherical divergence of the field will also lead to higher terminal velocities (Owocki & ud-Doula 2004). With the escape velocity defined as $v_{\rm esc}^2 = 2(1-\Gamma_{\rm el})GM_*/R_p$, we calculate the polar $v_\infty \approx 1.97\ v_{\rm esc}$ ($5.9\ v_{\rm esc}$) for our B-type (O-type) star in the point-star approximation. In the finite-disk case, we calculate $v_\infty \approx 2.30\ v_{\rm esc}$

Table 6.1 : Stellar and wind parameters used throughout this paper to represent a typical magnetic B-type star with a centrifugal magnetosphere and an O-type star with a dynamical magnetosphere. Values are taken from Table 2 in Bjorkman (1995) for the B star and ud-Doula et al. (2014) for the O star.

Type	M_*	R_p	$T_{\rm eff}$	α	$\Gamma_{\rm el}$	$\bar{Q}$	B_p	η_*
B	9.0 $M_\odot$	4.5 $R_\odot$	21000 K	0.56	9.27×10^{-3}	1025.14	11 kG	4.29×10^5
O	50 $M_\odot$	19 $R_\odot$	41860 K	0.6	0.5	500	3.715 kG	100

($v_\infty \approx 7.58\ v_{\rm esc}$) for the B star (O star). We note, however, that observed terminal velocities of magnetic stars are unlikely to ever reach these limits since the faster-than-spherical divergence of the wind will not persist indefinitely.

As given by XADM, the mCAK (Friend & Abbott 1986) terminal velocity for their ζ Pup analogue is 3000 km s^{-1} = 4.23 $v_{\rm esc}$. This value does not match our calculated non-magnetic, CAK $v_\infty = 3.11\ v_{\rm esc}$ for the same O star parameters as used in XADM. It is unknown why this discrepancy occurs, though we note that if we use the Friend & Abbott (1986) stellar parameters for a O star ($M_* = 60.5\ M_\odot$, $R_p = 13.8\ R_\odot$, $T = 49290$ K, $\log g = 3.94$), we calculate $v_\infty \approx 3070$ km s^{-1}.

Although closed field lines by definition cannot have terminal velocities, we find that the velocity along each field line can be represented with beta-velocity laws at zero-rotation with both v_∞ and β varying between field lines (Section 4.8, Section 5.6). This contradicts the XADM assumption of constant v_∞ and β across all field lines and suggests that there is no easily-defined global beta-velocity law for the magnetosphere. At critical rotation fractions $\omega > 0$, the acceleration from rigid-body rotation means that each line's velocity will not correspond to a beta-velocity law, seriously complicating the analysis. Here, we defer the impact of rotation to future studies and take $\omega = 0$.

6.3 Rederivation of XADM with magnetic values

We can now update the XADM model with an improved understanding of the wind velocity and density within the magnetosphere. In terms of the total kinetic energy of the unmagnetized wind $L_{\rm kin} = \dot{M}_{\rm unmag} v^2_{\infty,\rm unmag}/2$, the latitudinal distribution of shock-dissipated energy can be expressed as

$$\frac{dK_s}{d\mu_*} = \frac{\dot{m}_r}{\dot{m}_{\rm unmag}} \frac{v_s^2}{v^2_{\infty,\rm unmag}}, \tag{6.2}$$

where the subscript s indicates quantities at the shock, $\dot{m}_r$ is the radial surface mass-flux, and $\dot{m}_{\rm unmag} \equiv \dot{M}_{\rm unmag}/(4\pi R_p^2)$ is the surface mass-flux for an unmagnetized star. Throughout the rest of this chapter, we will follow XADM and assume that the unmagnetized mass-loss and terminal velocity are equal to their mCAK values: $\dot{M}_{\rm unmag} = \dot{M}_{\rm mCAK}$ and $v_{\infty,\rm unmag} = v_{\infty,\rm mCAK}$.

In XADM, this equation is further simplified by using the Owocki & ud-Doula (2004) mass-flux scaling relation with surface colatitude

$$\dot{m}_r = \mu_B \dot{m}_* = \mu_B^2 \dot{m}_{\rm mCAK}. \tag{6.3}$$

$\dot{m}_*$ is the surface mass-flux *into* the field line. $\mu_B = \hat{\mathbf{n}} \cdot \hat{\mathbf{B}}$ is the surface tilt of a dipole magnetic field line with $\hat{\mathbf{n}}$ the surface normal unit vector. For a non-rotating magnetic dipole, μ_B is simply given by

$$\mu_B = \frac{2\mu_*}{\sqrt{1+3\mu_*^2}}, \tag{6.4}$$

with $\mu_* = \cos\theta_m$ and θ_m the colatitude at which the dipole field line crosses the stellar surface. The relationship between the apex radius of each line (r_m) and the surface colatitude is given by

$$r_m = \frac{R_p}{1-\mu_*^2}. \tag{6.5}$$

Assuming a global beta velocity law $v = v_{\infty,\rm mCAK}(1-R_p/r)^\beta \equiv v_{\infty,\rm mCAK} w(r)$, XADM defines the scaled shock speeds $w_s \equiv (1-R_p/r_s)^\beta$ and $w_m \equiv (1-R_p/r_m)^\beta$ and obtains

$$\frac{dK_{s,\rm XADM}}{d\mu_*} = \mu_B^2 w_s^2 = \frac{4\mu_*^{2+4\beta}}{1+3\mu_*^2}\left(\frac{w_s}{w_m}\right)^2, \tag{6.6}$$

with the subscript m indicating apex quantities.

In a similar manner, we derive $dK/d\mu_*$ for our model magnetosphere using the zero-rotation mass-flux scaling derived in Paper II (Equation 5.60):

$$\dot{m}_* \approx \mu_B (\mu_B f_{\theta=0})^{1/\alpha} \Sigma_{f,0} \dot{m}_{\rm dip}, \tag{6.7}$$

with $f_{\theta=0}$ the finite-disk correction factor calculated at the pole and

$$\dot{m}_{\rm dip} = \dot{m}_{\rm CAK}\frac{r_c}{R_p} \tag{6.8}$$

is the dipole-corrected CAK surface mass-flux for a straight, polar line. r_c is the critical radius along the field line. For zero rotation, it is a good approximation to take the polar critical radius $r_{c,\theta=0} = r_c$ as constant along the stellar surface (Section 4.7). $\Sigma_{f,0}$, a parameter that corrects the error in assuming an optically-thick wind, is given by

$$\Sigma_{f,0} \equiv \frac{\left|1-\alpha-\left[\frac{1-\left(\frac{\chi_0}{\mu_B f_{\theta=0}}\right)^{1/\alpha-1}}{1-\left(\frac{\chi_0}{\mu_B f_{\theta=0}}\right)^{1/\alpha}}\right]\right|}{\alpha} \tag{6.9}$$

with

$$\chi_0 = \frac{1-\Gamma_{\rm el}}{\Gamma_{\rm el}\bar{Q}}. \tag{6.10}$$

$\Gamma_{\rm el} \equiv \kappa_e L_*/(4\pi c G M_*)$ is the Eddington parameter and $\bar{Q}$ is the dimensionless line-strength parameter introduced by Gayley (1995) to parameterize the CAK line-acceleration.

After noting that $\dot{m}_{\rm mCAK} = f_{\rm mCAK}^{1/\alpha}\dot{m}_{\rm CAK}$, we calculate the magnetospheric distribution of shock-dissipated kinetic energy as

$$\frac{dK_s}{d\mu_*} = \mu_B^{2+1/\alpha}\,\Sigma_{f,0}\left(\frac{f_{\theta=0}}{f_{\rm mCAK}}\right)^{1/\alpha}\frac{r_c}{R_p}\frac{v_{\infty,i}^2}{v_{\infty,{\rm mCAK}}^2}w_{s,i}^2, \tag{6.11}$$

where we use the best-fit velocity laws derived in Paper II for each line i: $v_s \approx v_{\infty,i}(1-R_p/r_s)^{\beta_i} = v_{\infty,i}w_{s,i}$. The ratio $f_{\theta=0}/f_{\rm mCAK}$ is approximately unity when the error in assuming an optically-thick wind is small ($\Sigma_{f,0} \to 1$), as for an O star. For B stars, however, this term will become slightly more important ($\approx 10\%$ boost to the calculated dissipated kinetic energy).

For a magnetic dipole geometry, we solve an implicit equation (akin to XADM

Equation B16) for the $w_{s,i}$ of each line by finding the root of

$$\mu_{s,i} - \mu_{s,i}^3 + \frac{3\mu_{s,i}^5}{5} + \frac{\mu_{s,i}^7}{7} = \\ = \frac{\chi_{\infty,i}}{3\dot{m}_{\rm wgh}} \frac{\sqrt{1+3\mu_{*,i}^2}}{1+3\mu_{s,i}^2} \left(\frac{w_{s,i}(1-w_{m,i}^{1/\beta})}{1-w_{s,i}^{1/\beta}} \right)^4 \left(\frac{1}{1-w_{s,i}^{1/\beta}} \right)^2, \tag{6.12}$$

where we express the μs in terms of the scaled velocities w_s and w_m:

$$\mu_{s,i}^2 = \frac{w_{m,i}^{1/\beta_i} - w_{s,i}^{1/\beta_i}}{1 - w_{s,i}^{1/\beta_i}} \tag{6.13}$$

$$\mu_{*,i}^2 = w_{m,i}^{1/\beta_i}. \tag{6.14}$$

Additionally, each line has an individual terminal speed cooling parameter

$$\chi_{\infty,i} = \frac{15\pi}{128} \frac{v_{\infty,i}^4 \; R_p}{\dot{M}_{\rm mCAK} \; \Lambda_m}. \tag{6.15}$$

In Equation 6.12, $\dot{m}_{\rm wgh} \equiv \dot{m}_*/\dot{m}_{\rm mCAK}$ is a weighing term to correct for the effect of the magnetic field redirecting the spherical mass-loss rate $\dot{M}_{\rm mCAK}$ into the field line. The XADM model follows Owocki & ud-Doula (2004) and takes $\dot{m}_{\rm wgh} = \mu_B$, but this fails to account for the boost in mass-loss due to dipole divergence (Equation 6.8) and the optically-thin correction (Equation 6.9). Thus, from Equation 6.7, we take

$$\dot{m}_{\rm wgh} = \mu_B^{1+1/\alpha} \left(\frac{f_{\theta=0}}{f_{\rm mCAK}} \right)^{1/\alpha} \Sigma_{f,0} \frac{r_{c,\theta=0}}{R_p}. \tag{6.16}$$

We use $r_{c,\theta=0} = 1.024 \; R_p$ (1.036 R_p) for the B star (O star) as calculated from the finite-disk-corrected general line-acceleration model.

In calculating $K_{\rm XADM}$, we assume $\beta = 1$ and use our model-calculated v_∞ for a optically-thick, spherically-diverging radial flow: $v_{\infty,\rm mCAK} \approx 2.66 \; v_{\rm esc}$ (3.11 $v_{\rm esc}$) for our B star (O star). For the XADM χ_∞, we will take $\dot{M}_{\rm unmag} = f_{\rm mCAK}^{1/\alpha} \dot{M}_{\rm CAK}$, with $f_{\rm mCAK} = 0.674$ (0.679) for our B star (O star).

Figure 6.2 shows our model results compared to the XADM model. Interestingly, though our model predicts an XADM overestimate of the shock-dissipated energy for a B star, it predicts an *underestimate* for an O star. This is almost entirely due to the difference in the terminal velocity ratios $v_\infty/v_{\infty,\mathrm{mCAK}}$ between our two stars. The model B star has $v_\infty/v_{\infty,\mathrm{mCAK}} = 0.87 < 1$, while the O star has $v_\infty/v_{\infty,\mathrm{mCAK}} = 2.44 > 1$. This implies that magnetized O stars have much stronger winds with higher kinetic luminosities than their unmagnetized counterparts, but the opposite is true for later-type massive stars (see discussion in Section 6.4).

Finally, we convert $dK_s/d\mu_*$ into an X-ray luminosity. Following XADM Equation 37 (and Kee et al. 2014), we write the fraction of kinetic energy emitted as X-rays above a

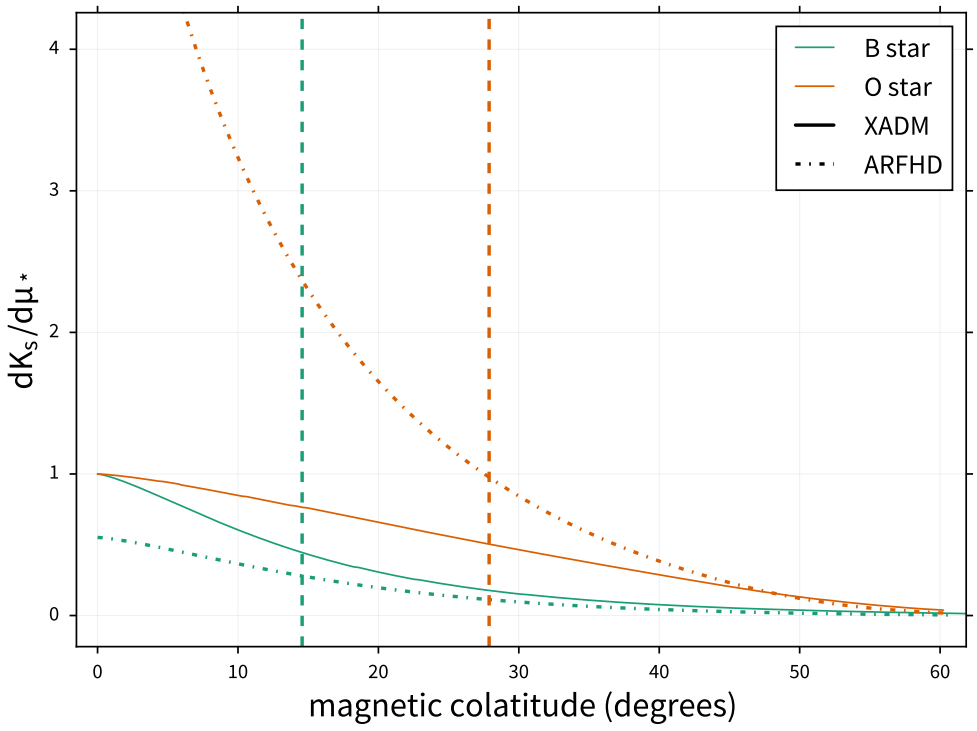


Figure 6.2 : Latitudinal distribution of shock-dissipated energy for our B star (green) and O star (orange) models relative to the kinetic energy of a non-magnetic mCAK wind, as predicted by XADM (Equation 6.6; dot-dashed) and our model (Equation 6.11; solid). Vertical dashed lines indicate surface colatitudes for lines with apex radius $r_m > R_c$, R_c the MHD-predicted closure radius (Equation 6.19)

threshold energy E_x as

$$f_x(T_s, E_x) = \int_0^{T_s} \frac{\bar{\Lambda}(T, E_x)}{\Lambda(T)} \frac{dT}{T_s}$$
$$\approx \mathrm{e}^{-E_x/k_b T_s} + \frac{E_x}{k_b T_s} \mathrm{E}_i(-E_x/k_b T_s) \tag{6.17}$$

with E_i being the exponential integral and $T_s = 3\bar{\mu}v_s^2/16k_b$ the post-shock temperature. The overall X-ray luminosity scaled to that of a unmagnetized wind is then

$$\frac{L_X}{L_{\rm kin}} = \int_0^{\mu_c} \frac{dK}{d\mu_*} f_x \, d\mu_*, \tag{6.18}$$

with $\mu_c^2 = 1 - R_p/R_c$. R_c is the MHD-estimated "closure radius"

$$R_c \approx R_p + 0.7[R_p(0.3 + \eta_*^{1/4}) - R_p], \tag{6.19}$$

characterizing the maximum radius at which the magnetic field still dominates the wind. Field lines with $r_m > R_c$ are considered "open" and will not have shock-produced X-rays.

In integrating Equation 6.18, we use the lower bound of N14's observed energy band to set the threshold energy $E_x = 0.5$ keV. We obtain $L_{\rm ARFHD}/L_{\rm XADM} = 0.58$ for our B star and 1.83 for our O star. This implies that the efficiency in converting wind kinetic luminosity into X-ray emission for B stars could be closer to 34%, if the MHD-derived luminosities (ud-Doula et al. 2014) are used, or 17%, if the best fit luminosities from N14 are used. For O stars, however, these efficiencies decrease to 11% (MHD) and 5.5% (N14).

6.4 Discussion

Here, we discuss some results from the rederivation of the XADM equations and some caveats. The first is that later-type magnetic B stars have weaker winds than non-magnetic B stars. Previous results (mCAK and Pauldrach et al. 1986) demonstrated that the finite-disk correction factor boosts the terminal velocity of the wind relative to the point star

case, which we see for our model O star. However, we do not see this behavior here for our finite-disk-corrected model B star. This contradiction is explained by the low optical depth of a finite-disk corrected B star wind. The mCAK results were derived using the optically-thick formulation of the radiative acceleration (CAK), which scales as $g_{\rm rad} \propto \rho^{-\alpha}$ for all densities. Our model incorporates the general form of the radiative driving (Owocki et al. 1988), which, at low optical depths, saturates at

$$g_{\rm rad} \approx \frac{\kappa_e \bar{Q} L_*}{4\pi r^2 c} \tag{6.20}$$

independent of density. In B stars, the lower density in the finite-disk-corrected wind causes the line-acceleration to saturate much more quickly; the overall result is a decreased terminal velocity relative to the mCAK case. O stars, on the other hand, have such large mass-loss rates that the line-acceleration stays in the optically thick limit over longer distances, resulting in a very large terminal velocities relative to mCAK. This implies that using the general form of the radiative driving, rather than the optically-thick CAK form, is important at low mass-loss rates.

From observations of X-ray luminosities, N14 found that the X-ray luminosity (L_X) scaled with mass-loss differently between magnetic O stars and later-type B stars. O stars (and some early B) follow $L_X \propto \dot{M}^{0.6}$, while the B stars follow $L_X \propto \dot{M}^{1.4}$. This difference in scaling could be a result of optically-thin winds in B stars. For B stars, increasing the mass-loss rate would reduce the effect of saturation on the radiative driving; this would boost each line's terminal velocity and lead to higher shock energies and resulting X-ray luminosities. Eventually, the mass-loss is high enough for the wind to be optically thick (as in O stars), and the terminal velocity no longer experiences this desaturation boost. The result is a shallower scaling of L_X at high $\dot{M}$.

Observational results indicate that B stars with large centrifugally-supported disks

(dark triangles in Figure 6.1) have higher X-ray luminosities than XADM would predict with 100% efficiency. The most likely explanation is that magnetospheric rotation boosts both the wind mass-loss and velocity, leading to a higher wind kinetic energy and resulting X-ray luminosity. Another contributing factor may be that the shock locations occur further down the line than would be predicted by the cooling length. This would lead to higher wind shock velocities and higher post-shock temperatures.

In the XADM model, the cooling length is the distance along the field line it takes the wind to cool from T_s to 0. However, we find that this assumption cannot be extended to cases with rotation ($\omega > 0$). Comparisons between ARFHD simulations of centrifugal

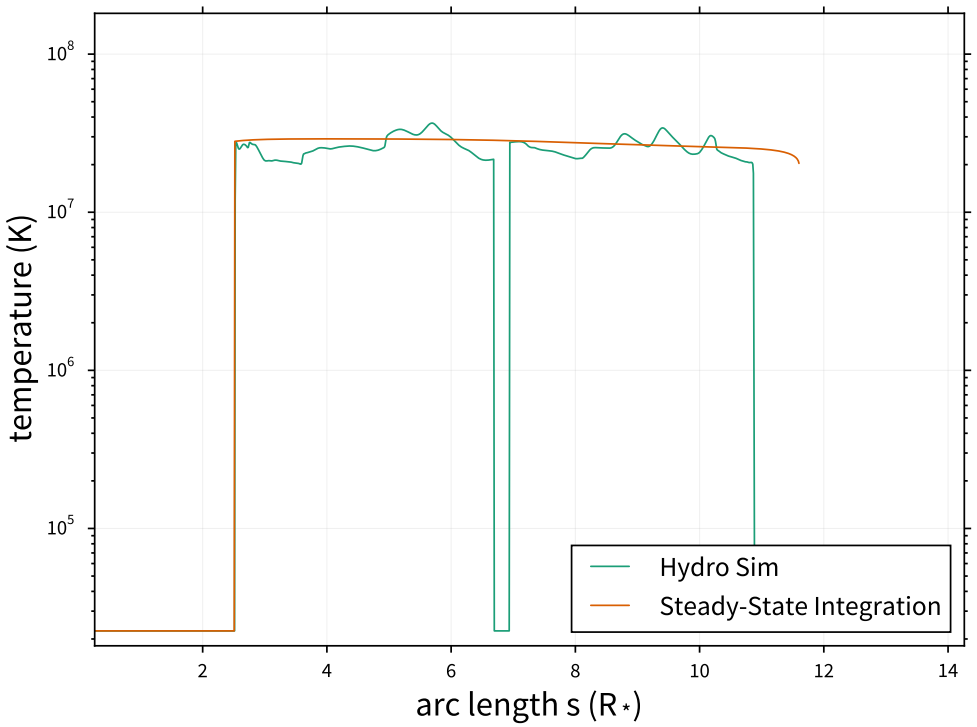


Figure 6.3 : Comparison between the temperature structure from ARFHD simulations (green) and integration with the steady-state ARFHD equations (orange) for a B star with $\omega = 0.2$. If the shock location is set by the cooling length, then the point at which the steady-state temperature begins to fall off (seen just before $s = 12\ R_*$) should be coincident with the temperature fall-off seen in the simulation. The difference is explained by the existence of a centrifugally-supported disk in the simulations.

magnetospheres and steady-state integrations find no evidence that the cooling length sets the shock location (Figure 6.3). The cooling length as calculated from integrating the post-shock temperature to zero is much longer than the distance from the shock to the disk. This occurs because the local wind conditions immediately following the shock are unaware that there is a large reservoir of centrifugally-supported plasma at the apex of the field line. When the wind reaches the disk, the local cooling length shortens abruptly and the temperature quickly falls. If the shock location is actually closer to the disk, using the cooling length would lead to an underestimate of the shock velocity and resulting X-ray luminosity for these magnetospheres.

It is uncertain whether this logic would also apply for zero rotation or for dynamical magnetospheres, which do not have these centrifugally-supported disks. However, the cooling length calculated in XADM still does not take into account the flow from the opposite footprint and any boost in density along the line that any interaction would provide. Further study is needed to ascertain what actually sets the shock length in magnetospheres and whether the cooling length is a good approximation for the shock location at zero rotation.

Another limitation of our analysis is the lack of rotation effects. Rotation will increase the velocity along each line, but its effect on the shock location and resulting shock velocity is yet unknown. One of the key assumptions in the XADM model in deriving the cooling length is the assumption of an isobaric post-shock flow ($\rho T \sim$ const), which greatly simplifies the calculation of the cooling length (and shock location). This assumption is supported by ARFHD simulations at zero rotation. However, at $\omega > 0$, corotation with the star induces a steady-state hydrostatic equilibrium along each field line in which the plasma pressure gradient balances the combined gravitocentrifugal force. In this case, the pressure (and temperature) actually *increases* from the shock to the apex of the field line!

Future models will need to take into account the effects of rotation on the X-ray luminosity and shock location.

6.5 Conclusion

In this chapter, we considered the assumptions made by the XADM model in calculating magnetic massive star X-ray luminosities. Using results from Paper II, we showed that the XADM assumption of a global beta-velocity law does not hold for all lines in a magnetosphere. Instead, each line has its own independent velocity structure. Additionally, XADM uses the non-magnetic values for the wind mass-loss and terminal velocity. These differ markedly from magnetic values, which leads to incorrect wind and shock energies. In Section 6.3, we use these magnetic values to rederive the XADM model.

The key results are summarized as follows:

1. Using non-magnetic values of the wind mass-flux and terminal velocity in the XADM model leads to an overestimate of the shock-dissipated energy (and resultant X-ray luminosity) for a B star, but an *underestimate* for an O star.

2. ud-Doula et al. (2014) found that the efficiency of turning shock energy to X-ray luminosity was about 20%, based on results from MHD simulations. Best-fits from observations suggest that this efficiency is about 10% (Nazé et al. 2014). Our results suggest that this efficiency factor is twice as low for O stars and twice as high for B stars, assuming that the XADM-calculated shock locations are correct.

3. The dipole divergence of magnetic field lines leads to faster winds than non-magnetic spherical divergence, provided that there is sufficient density in the wind such that the radiative driving does not reach the optically-thin limit. Magnetic O stars remain optically-thick even with the finite-disk correction reducing their mass-losses, and

thus have much higher terminal velocities than non-magnetic stars. B stars, on the other hand, have such low finite-disk-corrected mass-losses that the radiative driving saturates much more quickly; their eventual terminal velocities are weaker than that of their non-magnetic counterparts.

4. Nazé et al. (2014) found different scalings of X-ray luminosity with mass-loss rate for O stars ($L_X \propto \dot{M}^{0.6}$) and B stars ($L_X \propto \dot{M}^{1.4}$). This difference in scaling could be explained by optically-thin winds in B stars. Raising the mass-loss rate in B stars would both increase the wind density and provide a desaturation boost to the flow velocity, leading to much higher shock energies and resultant X-ray luminosities. Eventually, as in O stars, the mass-loss is sufficiently high that the line-acceleration no longer saturates and there is no desaturation boost. The result is a shallower scaling of L_X at high $\dot{M}$.

5. The XADM assumption that the shock location is set by the cooling length cannot be extended to field lines with centrifugally-supported disks. Hydrodynamic simulations suggest that the cooling length calculated from shock conditions is much longer than the actual distance from the shock to the disk at the apex of the field line. The large reservoir of plasma in the disk causes the local cooling length to abruptly decrease, reducing the overall size of the post-shock X-ray emitting region and increasing the shock velocity and post-shock temperature.

Further improvements are needed in magnetic massive star X-ray emission models, especially for magnetospheres with centrifugally-supported disks. We have taken the first steps by substituting magnetic values of wind mass-losses and velocities in place of non-magnetic values, but this alone is not sufficient to explain observed X-ray scalings. The next steps involve including rotation effects and deriving correct shock locations. This will

allow us to better estimate the size of the post-shock, X-ray emitting region and the overall X-ray luminosity.

References

Babel, J., & Montmerle, T. 1997, A&A, 323, 121

Berghoefer, T. W., Schmitt, J. H. M. M., Danner, R., & Cassinelli, J. P. 1997, A&A, 322, 167

Bjorkman, J. E. 1995, ApJ, 453, 369

Cassinelli, J. P., & Cohen, D. H. 1994, in IAU Symposium, Vol. 162, Pulsation; Rotation; and Mass Loss in Early-Type Stars, ed. L. A. Balona, H. F. Henrichs, & J. M. Le Contel, 189

Castor, J. I., Abbott, D. C., & Klein, R. I. 1975, ApJ, 195, 157

Friend, D. B., & Abbott, D. C. 1986, ApJ, 311, 701

Gayley, K. G. 1995, ApJ, 454, 410

Kee, N. D., Owocki, S., & ud-Doula, A. 2014, MNRAS, 438, 3557

Lucy, L. B. 1982, ApJ, 255, 286

Lucy, L. B., & Solomon, P. M. 1970, ApJ, 159, 879

Nazé, Y., Petit, V., Rinbrand, M., et al. 2014, ApJS, 215, 10

Nazé, Y., Broos, P. S., Oskinova, L., et al. 2011, ApJS, 194, 7

Owocki, S. P., Castor, J. I., & Rybicki, G. B. 1988, ApJ, 335, 914

Owocki, S. P., & ud-Doula, A. 2004, ApJ, 600, 1004

Pauldrach, A., Puls, J., & Kudritzki, R. P. 1986, A&A, 164, 86

ud-Doula, A., Owocki, S., Townsend, R., Petit, V., & Cohen, D. 2014, MNRAS, 441, 3600

ud-Doula, A., & Owocki, S. P. 2002, ApJ, 576, 413

Chapter 7

Summary and Conclusions

End? No, the journey doesn't end here. Death is just another path. One that we all must take.

J.R.R. Tolkien

In this thesis, I present magneto- and regular hydrodynamics equations relevant to physical conditions within both planetary and massive star magnetospheres. For planetary magnetospheres like Earth's, my goal is to understand how the stellar wind interacts with the magnetic field, eventually leading to reconnection. For massive star magnetospheres, my goal is to understand how the magnetic field changes the stellar wind in order to better model the resulting observational phenomena, like X-ray emission. In both cases, unique challenges make computational simulations with these equations difficult. In order to study reconnection in planetary magnetospheres, we need to use the Hall term; this, combined with the required resolution of tens of kilometers in a system tens of Earth radii wide, causes simulation timesteps to be minuscule. Similarly, the quite strong magnetic fields of massive stars with their high Alfvén speeds require tiny simulation timesteps. Here, I develop computational tools to either simplify the dynamic equations, in the case of massive stars, or simply solve them faster, in the case of planetary magnetospheres. These tools will further our understanding of how stellar winds affect large-scale magnetic fields, and vice versa.

In Chapter 2, I present my experience in porting a simple, two-dimensional ideal magnetohydrodynamic (MHD) code written in C to run on a graphics processing unit (GPU) using NVIDIA CUDA C. The key results of this chapter are:

1. Using a strategy of leaving data on the GPU, I was able to minimize data transfers between the CPU and the GPU. This led to a maximum of a 126x speedup on a NVIDIA GTX 480 compared to a single Intel i7 930 computational core. If each of the 8 cores on the CPU are used, this speedup is estimated to be approximately 15x (Section 2.4).

2. This initial result suggests that GPUs are a viable method for speeding up large and

complex simulations, like planetary magnetospheres.

In Chapter 3, I upgraded my two-dimensional ideal MHD code to three-dimensions and added the Hall term. The code was also updated from running on a single GPU to multiple GPUs using a Message Passing Interface scheme.

1. I find that the best method for optimizing GPU code is to focus on memory coalescing, which entails of making sure that adjacent threads in a warp access adjacent spaces of memory (Section 3.4).

2. Additional optimization strategies include minimizing the amount of data transfer and inter-node communication. For necessary data transfers, using pinned memory increased the overall speedup by about 3% (Section 3.4).

3. The code was validated with benchmarking tests: a whistler wave, designed to test the second-order Hall MHD algorithm; an Orszag-Tang vortex, designed to test the handling of turbulence and shocks; and a blast-wave, designed to test the handling of three-dimensional shocks (Section 3.5).

4. I assessed the capability of the code to scale to many GPUs with strong and weak scaling tests. Though our code does not scale well when the GPUs are underutilized, filling the memory capacity on each GPU results in excellent scaling at large problem sizes ($\approx 90\%$ weak scaling efficiency) (Section 3.6).

This code will, in the future, be used to investigate planetary magnetospheres in our solar system as well as extrasolar systems.

In Chapter 4, I present the Arbitrary Rigid-Field Hydrodynamics (ARFHD) Equations, a simplification of plasma behavior within massive star magnetospheres with the assumption that the stellar magnetic field totally dominates the wind flow. I undertake a

critical point analysis of the steady-state ARFHD equations in order to understand how a dipole magnetic field changes a massive-star line-driven wind. The key results are:

1. We are able to approximately confirm the Owocki & ud-Doula (2004) scaling for the influence of a magnetic dipole on the surface mass-flux, $\dot{m}_r \approx \mu_B^2 \dot{m}_{\rm CAK}$. While this scaling does not need much improvement, we provide a more accurate scaling equation (Equation 4.50) and detail which approximations are required to reproduce the OD04 scaling.

2. The effect of a optically-thin corrected line-force can be encapsulated in a OTC parameter, which we call Σ_0 (Equation 4.56). Including this does not have much of an effect for O-type stars, since their increased wind density means that there will be less difference in the corrected and uncorrected line-forces. B-type stars, on the other hand, have their surface mass-flux reduced by approximately 25-30% when the optically-thin correction is taken into account.

3. The effect of rotation can be similarly represented with a rotation-effect parameter, which we call $\aleph$ (Equation 4.64a). The amount of rotational boosting of the mass-flux is found to depend on both the rotational colatitude and the magnetic obliquity angle.

4. The effects of rotation and the optically-thin correction can not be decoupled, however. We find a different OTC parameter in the case of rotation, $\Sigma_{\rm rot}$ (Equation 4.72). Rotation is found to reduce the correction by driving a higher surface mass-flux.

5. The velocity structure within a magnetosphere cannot be described by a global beta-velocity law (Section 4.8). However, at least for zero rotation, we can well-fit each line with individual beta-velocity laws. The best-fit v_∞ and β do vary from line to

line, however. With rotation, the beta-velocity law assumption breaks down.

6. The global mass-loss rate for a optically-thin corrected line-force can be accurately estimated by multiplying the optically-thick mass-loss by the OTC parameter, Σ. We find "effective" magnetospheric mass-loss rates, in which the plasma does not fall back to the star, to be approximately 20-65% of the non-magnetic, non-rotating CAK mass-loss rate (Section 4.9).

In Chapter 5, I extend the analysis in Chapter 4 by correcting our prior point-star simplification with a finite-disk correction factor. The chief results are:

1. For an optically-thick wind, the finite-disk correction (f) can simply be multiplied to the point-star mass-flux scaling relation. At zero rotation, we obtain a scaling $\dot{m} \sim \mu_B^{1+1/\alpha} \dot{m}_{\theta=0}$. With rotation, the relationship between f and surface colatitude (and the resulting scaling relation) becomes too complex for an analytic function (Section 5.5.3).

2. The optically-thin-correction parameter Σ_f depends on the finite-disk correction. For O stars, their high mass-losses mean that f does not have a large effect on Σ_f compared to the point-star case. However, for B stars, f further decreases the wind density; as a result, Σ_f decreases by another 10% from ≈ 0.72 to 0.65. This results in an overall reduction of about 35% in B star mass-loss rates compared to their optically-thick mass-loss rate (Section 5.5.4).

3. We reproduce the Paper I result that the magnetosphere does not have a global beta-velocity law; each line has an unique velocity structure. Additionally, with rotation, these velocity structures do not follow a beta-velocity law. Compared to the point-star case, the finite-disk factor is found to increase the polar terminal velocity and how

quickly the wind reaches this limit (represented by a decreasing β in the zero-rotation velocity law fits) (Section 5.6).

4. The closure radius is found to move further out, despite the higher wind terminal velocities for each line (Section 5.7).

5. Unlike the point-star case, global mass-loss rates are found to stay roughly constant with rotation (and decrease at the highest rotation rates). This is caused by a decrease in the stellar surface intensity as rotation increases, keeping luminosity fixed. The "effective" mass-loss rate, in which plasma does not fall back to the star, is found to be about 20% (at $\omega = 0.2$) to 40% (at $\omega = 0.8$) of the non-rotating, non-magnetic mCAK mass-loss rate (Section 5.8).

Finally, in Chapter 6, I take my results from the steady-state, finite-disk-corrected ARFHD analysis and use them to update the most current magnetic massive star X-ray model (XADM; ud-Doula et al. 2014). The notable results are:

1. Using non-magnetic values of the wind mass-flux and terminal velocity in the XADM model leads to an overestimate of the shock-dissipated energy (and resultant X-ray luminosity) for a B star, but an *underestimate* for an O star (Section 6.3).

2. ud-Doula et al. (2014) found that the efficiency of turning shock energy to X-ray luminosity was about 20%, based on results from MHD simulations. Best-fits from observations suggest that this efficiency is about 10% (Nazé et al. 2014). Our results suggest that this efficiency factor is twice as low for O stars and twice as high for B stars, assuming that the XADM-calculated shock locations are correct (Section 6.3).

3. The dipole divergence of magnetic field lines leads to faster winds than non-magnetic spherical divergence, provided that there is sufficient density in the wind such that the

radiative driving does not reach the optically-thin limit. Magnetic O stars remain optically-thick even with the finite-disk correction reducing their mass-losses, and thus have much higher terminal velocities than non-magnetic stars. B stars, on the other hand, have such low finite-disk-corrected mass-losses that the radiative driving saturates much more quickly; their eventual terminal velocities are weaker than that of their non-magnetic counterparts (Section 6.4).

4. Nazé et al. (2014) found different scalings of X-ray luminosity with mass-loss rate for O stars ($L_X \propto \dot{M}^{0.6}$) and B stars ($L_X \propto \dot{M}^{1.4}$). This difference in scaling could be explained by optically-thin winds in B stars. Raising the mass-loss rate in B stars would both increase the wind density and provide a desaturation boost to the flow velocity, leading to much higher shock energies and resultant X-ray luminosities. Eventually, as in O stars, the mass-loss is sufficiently high that the line-acceleration no longer saturates and there is no desaturation boost. The result is a shallower scaling of L_X at high $\dot{M}$ (Section 6.4).

5. The XADM assumption that the shock location is set by the cooling length cannot be extended to field lines with centrifugally-supported disks. Hydrodynamic simulations suggest that the cooling length calculated from shock conditions is much longer than the actual distance from the shock to the disk at the apex of the field line. The large reservoir of plasma in the disk causes the local cooling length to abruptly decrease, reducing the overall size of the post-shock X-ray emitting region and increasing the shock velocity and post-shock temperature.

This thesis represents a significant advance in understanding massive-star magnetospheres by allowing us to understand how a large-scale, strong magnetic field affects line-driven winds. Using the tools I have developed, we can better calculate the mass-loss

rates and velocity structures within these magnetospheres. This has enormous implications for understanding the magnetospheric emission in X-rays, as well as studying the small-scale processes that contribute to their steady-states.

References

Nazé, Y., Petit, V., Rinbrand, M., et al. 2014, ApJS, 215, 10

Owocki, S. P., & ud-Doula, A. 2004, ApJ, 600, 1004

ud-Doula, A., Owocki, S., Townsend, R., Petit, V., & Cohen, D. 2014, MNRAS, 441, 3600

Appendix A

On Calculating the Finite-Disk Correction Factor

Just when I thought I was out, they pull *me back in.*

Michael Corleone

A.1 Introduction

Our algorithm for calculating the finite-disk correction factor (fdcf) is based upon Cranmer & Owocki (1995), who considered the effects of oblation, limb-darkening, and gravity-darkening. Although the work presented in this thesis does not include darkening effects, it does include the oblation of the finite-disk due to rotation. Regardless, our code has the capability to calculate all these effects. Here, in Section A.2, we present the overall calculation of the finite-disk correction factor following Cranmer & Owocki (1995), updating it with the general form of the line-driven acceleration (Owocki et al. 1988) rather than the usual Castor et al. (1975) form. In Section A.3, we introduce an algorithm to separate the fdcf spatial components from the velocity gradient components, which speeds up the fdcf calculation for a fixed spatial grid.

A.2 Finite-Disk Correction Factor

The coordinate system for the calculation is described in detail in Section 5.3; here we briefly review. We center the star at the Cartesian origin $\vec{\mathbf{r}} = [x, y, z] = [0, 0, 0]$ with its surface denoted by spherical coordinates $R_*(\theta)$, θ, and ϕ. A "wind-centered" coordinate system $\vec{\mathbf{r}'} = [x', y', z']$ is defined by translating the origin to a wind point $\vec{\mathbf{r_0}} = [x_0, y_0, z_0]$ and rotating the axes such that the z'-axis points along $\vec{\mathbf{r}}$ away from the star and the y'-axis is parallel to the y-axis. The x'-axis is thus tilted by an angle $\theta_0 = \arccos(z_0/r_0)$ relative to the x-axis. Figure 5.1 illustrates the coordinate geometry of the star- and wind-centered frames.

Evaluating the overall finite-disk factor requires evaluation of the stellar intensity and a projected velocity gradient along arbitrary rays $\hat{\mathbf{n}}$, directed from points on the

stellar surface to the wind point. The necessary angular integrations are most conveniently expressed in terms of the wind-centered spherical coordinates, (r', θ', ϕ').

From this perspective, the overall finite-disk correction is

$$\vec{\mathbf{f}} = \frac{\oint I_* \left|\hat{\mathbf{n}} \cdot \nabla(\hat{\mathbf{n}} \cdot \vec{\mathbf{v}})\right| \left[\left(1 + \frac{\bar{Q}\rho\kappa_e c}{|\hat{\mathbf{n}} \cdot \nabla(\hat{\mathbf{n}} \cdot \vec{\mathbf{v}})|}\right)^{1-\alpha} - 1\right] \hat{\mathbf{n}} \, d\mu' d\phi'}{\frac{L_*}{4\pi r_0^2} \left|\frac{\partial v_r}{\partial r}\right| \left[\left(1 + \frac{\bar{Q}\rho\kappa_e c}{|\partial v_r/\partial r|}\right)^{1-\alpha} - 1\right]}, \tag{A.1}$$

with $\mu' \equiv \cos\theta'$ and we define the stellar surface intensity

$$I_* = \sigma_B T_{\rm eff}^4 D, \tag{A.2}$$

with σ_B the Stefan-Boltzmann constant, $T_{\rm eff}$ the stellar effective temperature, and $D = D(r_0, \mu')$ a limb-darkening function.

For limb-darkening, we assume a gray atmosphere and use the Eddington approximation to obtain

$$D = \begin{cases} 0, & -1 \leq \mu' < \mu_* \\ (2 + 3\mu'')/(4\pi), & \mu_* \leq \mu' < 1 \end{cases} \tag{A.3}$$

with μ_* marking the edge of the star. At zero rotation, $\mu_*^2 = 1 - (R_*/r_0)^2$. μ'', the cosine of the angle between the local normal surface and the ray angle $\hat{\mathbf{n}}$ is

$$\mu'' = -\frac{\vec{\mathbf{g}} \cdot \hat{\mathbf{n}}}{|\vec{\mathbf{g}}|}, \tag{A.4}$$

with the surface gravity vector $\vec{\mathbf{g}}(\theta_s) = [g_r, g_\theta, g_\phi]$:

$$\begin{aligned} g_r &= (1 - \Gamma_{\rm el}) G M_* \left(\frac{-1}{R_*^2} + \frac{8\omega^2}{27 R_p^3} R_* \sin^2\theta_s\right), \\ g_\theta &= (1 - \Gamma_{\rm el}) G M_* \left(\frac{8\omega^2}{27 R_p^3} R_* \sin\theta_s \cos\theta_s\right), \\ g_\phi &= 0. \end{aligned} \tag{A.5}$$

In order to calculate $\vec{\mathbf{g}}$ for a given ray $\hat{\mathbf{n}}$, we need to calculate the surface colatitude θ_s at which the ray intersects the stellar surface (we do this numerically). For zero rotation, the surface normal vector is purely radial, and μ'' simplifies to

$$\mu'' = \sqrt{\frac{\mu'^2 - \mu_*^2}{1 - \mu_*^2}}. \tag{A.6}$$

If we wish to ignore the effects of limb-darkening, we simply take $D = 1/\pi$ for the region $\mu_* < \mu' < 1$ in Equation A.3.

The influence of von Zeipel gravity darkening (von Zeipel 1924; Slettebak 1949) is parameterized through the variation of effective temperature over the stellar surface

$$\sigma_B T_{\rm eff}^4(\theta_s) = \frac{L_*}{\Sigma_g} |\vec{\mathbf{g}}(\theta_s)|, \tag{A.7}$$

where the surface gravity vector $\vec{\mathbf{g}}$ is given above and the constant Σ_g is the surface-weighed gravity

$$\Sigma_g = \oint |\vec{\mathbf{g}}| dS = 2\pi \int_0^\pi |g| \frac{R_*^2 \sin\theta d\theta}{-g_r/|\vec{\mathbf{g}}|}. \tag{A.8}$$

Cranmer & Owocki (1995) computed values of Σ_g numerically and fit them using least squares as a power series in ω:

$$\begin{aligned} \Sigma_g \approx 4\pi G M_* [1 &- 0.19696\omega^2 - 0.094292\omega^4 \\ &+ 0.33812\omega^6 - 1.3066\omega^8 + 1.8286\omega^{10} - 0.92714\omega^{12}]. \end{aligned} \tag{A.9}$$

The relative error in this fit is no more than 0.6% for all values of ω. Again, we need to know where the ray intersects the stellar surface (θ_s) in order to calculate the surface gravity vector.

Without gravity darkening, the effective temperature is simply taken as the stellar luminosity over the surface area of the star, $\sigma_B T_{\rm eff}^4 = L_*/\Sigma_{\rm surf}$. For an oblate star, the

stellar surface area can be fit as a power series in ω (Cranmer 1996):

$$\Sigma_{\rm surf} = 4\pi R_p^2 \, (1 + 0.91444\omega^2 + 0.28053\omega^4 \\ - 1.9014\omega^6 + 6.8298\omega^8 - 9.5002\omega^{10} + 4.6631\omega^{12}), \tag{A.10}$$

which reduces to the usual $4\pi R_p^2$ for non-rotating stars.

The radiative driving term is given by

$$|\hat{\mathbf{n}} \cdot \nabla(\hat{\mathbf{n}} \cdot \vec{\mathbf{v}})| \left[\left(1 + \frac{\bar{Q}\rho\kappa_e c}{|\hat{\mathbf{n}} \cdot \nabla(\hat{\mathbf{n}} \cdot \vec{\mathbf{v}})|} \right)^{1-\alpha} - 1 \right], \tag{A.11}$$

where α is the CAK power-law index, $\bar{Q}$ is the dimensionless line strength parameter introduced by Gayley (1995), κ_e is the electron scattering opacity, ρ is the wind density, and c the usual speed of light. The $\hat{\mathbf{n}}\cdot\nabla(\hat{\mathbf{n}}\cdot\vec{\mathbf{v}})$ term is the projected gradient of the projected velocity along the ray. This vector has been derived for a general three-dimensional geometry by Batchelor (2000):

$$\begin{aligned} \hat{\mathbf{n}} \cdot \nabla(\hat{\mathbf{n}} \cdot \vec{\mathbf{v}}) = {} & \frac{\partial v_r}{\partial r}\hat{n}_r^2 + \left(\frac{1}{r_0}\frac{\partial v_\theta}{\partial \theta} + \frac{v_r}{r_0} \right) \hat{n}_\theta^2 + \left(\frac{1}{r_0 \sin\theta_0}\frac{\partial v_\phi}{\partial \phi} + \frac{v_r}{r_0} + \frac{v_\theta \cot\theta_0}{r_0} \right) \hat{n}_\phi^2 \\ & + \left(\frac{\partial v_\theta}{\partial r} - \frac{v_\theta}{r_0} + \frac{1}{r_0}\frac{\partial v_r}{\partial \theta} \right) \hat{n}_r\hat{n}_\theta + \left(\frac{\partial v_\phi}{\partial r} + \frac{1}{r_0 \sin\theta_0}\frac{\partial v_r}{\partial \phi} \right) \hat{n}_r\hat{n}_\phi \\ & + \left(\frac{1}{r_0}\frac{\partial v_\phi}{\partial \theta} + \frac{1}{r_0 \sin\theta_0}\frac{\partial v_\theta}{\partial \phi} \right) \hat{n}_\theta\hat{n}_\phi + \frac{v_\phi}{r_0 \sin\theta_0}\hat{n}_\phi \left(-\hat{n}_r \sin\theta_0 - \hat{n}_\theta \cos\theta_0 \right). \end{aligned} \tag{A.12}$$

The ray spherical unit vector components are given by

$$\begin{aligned} \hat{n}_r &= \cos\theta' = \mu', \\ \hat{n}_\theta &= \sin\theta' \cos\phi' = \sqrt{1-\mu'^2}\cos\phi', \\ \hat{n}_\phi &= \sin\theta' \sin\phi' = \sqrt{1-\mu'^2}\sin\phi'. \end{aligned} \tag{A.13}$$

For a radial wind, all angular components and derivatives are ignored, resulting in a line-of-sight gradient of

$$\hat{\mathbf{n}} \cdot \nabla(\hat{\mathbf{n}} \cdot \vec{\mathbf{v}}) = \frac{\partial v}{\partial r}\mu'^2 + (1-\mu'^2)\frac{v}{r_0}. \tag{A.14}$$

For a round stellar disk with no darkening and a optically-thick line-force, the integral reduces to

$$\vec{\mathbf{f}} = \frac{2}{(R_*/r_0)^2(\partial v_r/\partial r)^\alpha} \int_{\mu_*}^{1} \left[\mu'^2 \frac{\partial v_r}{\partial r} + (1-\mu'^2)\frac{v_r}{r_0} \right]^\alpha \hat{\mathbf{n}}\ d\mu' \tag{A.15}$$

after taking into account symmetry in ϕ'. The radial component of Equation A.15 is easily integrated (Castor et al. 1975), and often used in numerical models (Friend & Abbott 1986; Pauldrach et al. 1986):

$$f_r = \frac{(1+\sigma)^{1+\alpha} - (1+\sigma\mu_*^2)^{1+\alpha}}{\sigma(1+\alpha)(1+\sigma)^\alpha(1-\mu_*^2)}, \tag{A.16}$$

with $\sigma = (d\ \ln v_r/d\ \ln r) - 1$ and μ_* marking the non-oblate stellar limb (as defined above).

In our finite-disk corrected steady-state analysis of a magnetospheric wind (Chapter 5), our knowledge of the velocity structure is limited to individual lines, with $\vec{\mathbf{v}} = v\hat{\mathbf{s}}$, $\hat{\mathbf{s}}$ being the unit vector indicating the direction the line is going in Cartesian space. We follow Townsend et al. (2007) and approximate the line velocity as the radial velocity and the along-line velocity derivative ($\partial v/\partial s$) as the radial velocity derivative:

$$\hat{\mathbf{n}} \cdot \nabla(\hat{\mathbf{n}} \cdot \vec{\mathbf{v}}) = \frac{\partial v}{\partial s}\mu'^2 + (1-\mu'^2)\frac{v}{r_0}. \tag{A.17}$$

This gives us a magnetospheric fdcf vector

$$\vec{\mathbf{f}} = \frac{\oint \sigma_B T_{\rm eff}^4 D \left| \frac{\partial v}{\partial s}\mu'^2 + (1-\mu'^2)\frac{v}{r_0} \right| \left[\left(1 + \frac{\bar{Q}\rho\kappa_e c}{|\frac{\partial v}{\partial s}\mu'^2 + (1-\mu'^2)\frac{v}{r_0}|} \right)^{1-\alpha} - 1 \right] \hat{\mathbf{n}}\ d\mu' d\phi'}{\frac{L_*}{4\pi r_0^2} \left| \frac{\partial v}{\partial s} \right| \left[\left(1 + \frac{\bar{Q}\rho\kappa_e c}{|\partial v/\partial s|} \right)^{1-\alpha} - 1 \right]}, \tag{A.18}$$

and we take the final fdcf as $f = \vec{\mathbf{f}} \cdot \hat{s}$.

A.3 Spatially-Fixed Algorithm for Magnetospheric Finite-Disk Factor

The calculations described above are quite computationally intensive, especially when determining surface θ_s for each ray $\hat{\mathbf{n}}$ in calculating limb/gravity-darkening effects. Since

our hydrodynamic and steady-state codes calculate the fdcf many times, we need a better way to calculate this factor so that our codes do not become computationally clogged. Fortunately, the codes have fixed grids; for each grid point, the rays tracing the finite-disk of the star will not change with time and velocity structure! Thus, we can split the fdcf calculation of the previous section into "spatially-dependent" and "velocity-dependent" components. At the beginning of the program, we calculate the spatial components once, for the fixed grid (or point). At each timestep, we calculate the radiative driving term Equation A.11, which depends on the line-of-sight velocity gradient, and combine it with the spatial components to get the overall fdcf.

Since there is an integral at the heart of the fdcf evaluation, we cannot simply calculate one term encapsulating the entire spatial dependency. Instead, we must create an array containing the spatial integrand components of each quadrature point (θ', ϕ') in the overall integral; there will be $N_{\theta'}$ $(N_{\phi'})$ quadrature nodes in θ' (ϕ') such that the array will have size $(N_{\theta'}, N_{\phi'})$. Using Gauss-Legendre (G-L) quadrature, the array is then calculated at one single location (x', y', z') with a magnetic field line direction $\hat{\mathbf{s}}$:

$$S(i,j) = W_{\theta'}(i) W_{\phi'}(j) \sigma_B T_{\rm eff}^4 D \ \hat{\mathbf{n}}(\theta'(i), \phi'(j)) \cdot \hat{\mathbf{s}}, \tag{A.19}$$

where $W_{\theta'}$ and $W_{\phi'}$ are the G-L quadrature weights associated with the G-L nodes $\theta'(i)$ and $\phi'(j)$. The limb-darkening (D) and gravity-darkening $(\sigma_B T_{\rm eff}^4)$ terms are calculated as described above for each individual $(\theta'(i), \phi'(j))$ pair. During this calculation, we save the $\hat{\mathbf{n}}$ components in an array

$$\begin{aligned} H(i,j,0) &= \hat{n}_r(i,j) \\ H(i,j,1) &= \hat{n}_\theta(i,j) \\ H(i,j,2) &= \hat{n}_\phi(i,j), \end{aligned} \tag{A.20}$$

for use in calculating the radiative driving term.

The radiative driving term depends on the line-of-sight velocity gradient ($\hat{\mathbf{n}}\cdot\nabla(\hat{\mathbf{n}}\cdot\vec{\mathbf{v}})$), which is itself dependent on velocity, the velocity gradient, and the orientation of the tracing rays at the spatial location (x', y', z'). Generally, the line-of-sight gradient can simply be calculated using numerical derivatives for each tracing ray and placed in an array of size $(N_{\theta'},\ N_{\phi'})$ such that $L(i,j) = \hat{\mathbf{n}}(i,j)\cdot\nabla(\hat{\mathbf{n}}(i,j)\cdot\vec{\mathbf{v}}(i,j))$.

For our magnetosphere, we simplify the velocity gradient term according to Equation A.17. This term can be simplified into an array with three components

$$\begin{aligned} G(0) &= 1 \\ G(1) &= 1/r_0 \\ G(2) &= 1/r_0, \end{aligned} \tag{A.21}$$

with r_0 the distance from the origin to the wind point in the star-centered coordinate frame. This allows us to simply calculate the line-of-sight velocity gradient given only the line velocity v and gradient $\partial v/\partial s$ at our location:

$$L(i,j) = H(i,j,0)^2\ G(0)\ \frac{\partial v}{\partial s} + H(i,j,1)^2\ G(1)\ v + H(i,j,2)^2\ G(2)\ v. \tag{A.22}$$

The overall velocity-dependent term is then given by

$$V(i,j) = \begin{cases} |L(i,j)|^{\alpha}, & \text{optically} - \text{thick} \\ |L(i,j)|\left[\left(1 + \frac{\rho\bar{Q}\kappa_e c}{|L(i,j)|}\right)^{1-\alpha} - 1\right], & \text{general} \end{cases}, \tag{A.23}$$

and the final fdcf numerator is

$$F_N = \sum_{i}^{N_{\theta'}}\sum_{j}^{N_{\phi'}} S(i,j)V(i,j). \tag{A.24}$$

The spatial component of the fdcf denominator is simply $F_s = L_*/(4\pi r_0^2)$, such that

the overall fdcf denominator is calculated through

$$F_D = \begin{cases} F_s \ \left|\frac{\partial v}{\partial s}\right|^{\alpha}, & \text{optically} - \text{thick} \\ F_s \ \left|\frac{\partial v}{\partial s}\right| \left[\left(1 + \frac{\rho \bar{Q} \kappa_e c}{|\partial v/\partial s|}\right)^{1-\alpha} - 1\right], & \text{general} \end{cases}, \tag{A.25}$$

and we get the final factor $F = F_N/F_D$.

We test this algorithm by setting up a grid of 125,000 points in $(r, v, \partial v/\partial s)$ phase space and calculate the fdcf at each point, once with the full integration and once with the decomposed algorithm. The results are checked for differences and the averaged calculation times compared. We repeat this process for different combinations of limb-/gravity-darkening at two rotation rates, $\omega = 0, 0.5$ (Table A.1). We see about a 14x speedup without rotational darkening and a 93x speedup with rotational darkening, which significantly improves the feasibility of running simulations which incorporate the finite-disk factor. Since we only have to calculate the spatial information at the beginning of the program, this overhead (timing on the order of the "full" calculation in Table A.1), is invoked just once.

	Run 1	2	3	4	5	6
ω	0.0	0.5	0.0	0.5	0.5	0.5
darkening?	none	none	limb	limb	grav	limb+grav
decomp	7.25×10^{-5}	7.23×10^{-5}	7.23×10^{-5}	7.25×10^{-5}	7.28×10^{-5}	7.23×10^{-5}
full	1.00×10^{-3}	1.36×10^{-3}	1.01×10^{-3}	6.94×10^{-3}	6.71×10^{-3}	6.844×10^{-3}
speedup	13.8	18.8	14.0	95.7	92.3	94.7
fdcf calc error	5.49×10^{-15}	5.73×10^{-15}	5.73×10^{-15}	5.07×10^{-15}	5.50×10^{-15}	5.09×10^{-15}

Table A.1 : Run time comparison between full finite-disk integration ("full") and decomposed algorithm ("decomp") for different combinations of rotation rate, limb darkening, and gravity darkening. "fdcf calc error" is the maximum relative error between the spatially-decomposed algorithm and the full numerical integration. Here, the error is on the order of machine epsilon.

References

Batchelor, G. K. 2000, An Introduction to Fluid Dynamics, 635

Castor, J. I., Abbott, D. C., & Klein, R. I. 1975, ApJ, 195, 157

Cranmer, S. R. 1996, PhD thesis, Bartol Research Institute, University of Delaware

Cranmer, S. R., & Owocki, S. P. 1995, ApJ, 440, 308

Friend, D. B., & Abbott, D. C. 1986, ApJ, 311, 701

Gayley, K. G. 1995, ApJ, 454, 410

Owocki, S. P., Castor, J. I., & Rybicki, G. B. 1988, ApJ, 335, 914

Pauldrach, A., Puls, J., & Kudritzki, R. P. 1986, A&A, 164, 86

Slettebak, A. 1949, ApJ, 110, 498

Townsend, R. H. D., Owocki, S. P., & Ud-Doula, A. 2007, MNRAS, 382, 139

von Zeipel, H. 1924, MNRAS, 84, 665