DISSERTATION

submitted to the

Combined Faculties for the Natural Sciences and for Mathematics

of the Ruperto-Carola University of Heidelberg, Germany

for the degree of

Doctor of Natural Sciences

Put forward by

Diplom-Physiker Bernhard W. Keil

born in Besigheim, Germany

Oral examination: 9th of December 2009

Astro-GRIPS,

the

General Relativistic Implicit Parallel Solver

for

Astrophysical Fluid Flows

Referees: **Prof. Dr. Max Camenzind**

Priv.-Doz. Dr. Christian Fendt

Zusammenfassung

In dieser Arbeit wurde das Simulationsprogramm Astro-GRIPS, der General Relativistic Implicit Parallel Solver, entwickelt, der die dreidimensional axialsymmetrischen allgemein-relativistischen hydrodynamischen Euler und Navier-Stokes Gleichungen mit fester Hintergrunds-Metrik eines Schwarzschild oder Kerr Schwarzen Loches mit impliziten Methoden löst. Es ist eine fast vollständige Neuauflage eines alten 'Spaghetti-Code' artigen seriellen Fortran 77 Simulationsprogrammes. Durch Modernisierung und Optimierung ist ein modernes, gut strukturiertes, benutzerfreundliches, flexibles und erweiterbares Simulationsprogramm in Fortran 90/95 entstanden. Die Diskretisierung nach dem Finite Volumen Verfahren gewährleistet die Erhaltungseigenschaften der Gleichungen und die Methode der iterativen Defekt-Korrektur wird benutzt um die Nichtlinearitäten aufzulösen. Es enthält verschiedene Lösungsverfahren von rein explizit zu voll implizit, die bis zur dritten Ordnung im Raum und zweiten Ordnung in der Zeit genau sind. Die großen dünn besetzten linearen Gleichungssysteme, die bei den impliziten Methoden aufgestellt werden, können mit der Black-White Line-Gauß-Seidel Relaxationsmethode (BW-LGS), der Approximate Factorization Methode (AFM) oder den Krylov Unterraum-Methoden wie GMRES gelöst werden. Die beste Lösungsmethode und der Grad der Gleichungskopplung hängen vom Problem ab. Die Optimierung der Gleichungssystem-Aufstellung, die MPI-Parallelisierung für Computersysteme mit verteiltem Arbeitsspeicher und einige Newtonsche und relativistische Testrechnungen wurden erfolgreich durchgeführt.

Abstract

In this work the development of the simulation code Astro-GRIPS, the General Relativistic Implicit Parallel Solver, is performed, which solves the three-dimensional axi-symmetric general relativistic hydrodynamic Euler or Navier-Stokes equations under the assumption of a fixed background metric of a Schwarzschild or Kerr black hole using time-implicit methods. It is an almost total re-write of an old spaghetti-code like serial Fortran 77 simulation program. By modernization and optimization it is now a modern, well structured, user-friendly, flexible and extensible simulation program written in Fortran 90/95. The finite volume discretization ensures conservation and the defect-correction iteration strategy is used to resolve the non-linearities of the equations. One can use a variety of solution procedures that range from purely explicit up to fully implicit schemes with up to third order spatial and second order temporal accuracy. The large sparse linear equation systems used for the implicit methods can be solved by the Black-White Line-Gauß-Seidel relaxation method (BW-LGS), the Approximate Factorization Method (AFM) or by Krylov Subspace Iterative methods like GMRES. The optimal solution method and the coupling of equations is problem-dependent. Optimizations in the matrix construction, the MPI-Parallelization for distributed memory machines and several Newtonian and relativistic tests were conducted successfully.

The numerical results presented here have been obtained using the here described simulation program Astro-GRIPS, the General Relativistic Implicit Parallel Solver, developed at the Landessternwarte as part of the GR-I-RMHD (General Relativistic - Implicit - Radiative Magneto-HydroDynamics) project which is supervised by Priv.-Doz. Dr. Ahmad A. Hujeirat and financially supported by the Klaus-Tschira-Stiftung: project number: 00.099.2006.

Contents

List of Figures

List of Tables

1 Introduction

In 1609, 400 years ago, Galileo was one of the first to look into the sky with the newly invented telescope and to present the results to the public: On the 25th of August 1609 he conducted the first astronomical outreach activity when he met with policy and law-makers from the Venetian Republic. This remarkable event is now celebrated by many public events of the International Year of Astronomy 2009. But 2009 is also the 400th anniversary of Kepler's Astronomia Nova, the cornerstone of modern astronomy and the year of the launch of NASA's Kepler mission to seek for habitable planets: an optical photometer mounted on a spacecraft, to detect a dip in the lightcurve via the transit method for the search of exoplanets, this for example shows that there was a huge development in Astronomy since then. Before the invention of the telescope it was only possible to do Astronomical observations with the bare eyes. So, for example, Tycho Brahe did a great job in astrometry by taking a precise bearing of the stars and planets and compiling a catalogue. Using this data Kepler calculated the shape of the planet orbits by hand or by a slide rule without the use of a computer, which resulted in the famous Kepler laws.

In the days of Galileo, Kepler and Tycho Brahe, drawings were the only possibility to document observations. Only at the end of the 18th century and the beginning of the 19th century, Max Wolf, the first director of the Landessternwarte Heidelberg, did pioneering work in the field of Astrophotography and detected many asteroids with the Bruce telescope. Today photo-multipliers and CCD cameras are used instead due to their larger quantum efficiency and the easier handling of electronic data processing.

Since the first telescope used for Astronomy a major development in technology has happened. Nowadays one does not only observe in the optical wavelength regime, observations are done in the whole spectral range: radio, microwave, infrared, visible, ultraviolet, x-rays, up to the high energy gamma rays and also particles like neutrinos. Multi-wavelength observations give us more information and hints on the nature of the observed objects. Since the Earth's atmosphere blocks some spectral regimes, it is only possible to observe in these wavelengths from space. So many scientific space missions were conducted in the last half century. In the last few years the Hubble space telescope, after an error in the optical system was corrected by an additionally mounted correction lens, presented beautiful images of galaxies in the optical. But nowadays with the new instrumental techniques of adaptive optics (AO) and interferometry used at the VLT, the Very Large Telescope of the European Southern Observatory (ESO), one can even overcome the results of the Hubble space telescope from the ground of the Earth, e.g. by using the FORS instruments (Appenzeller et al. 2000, 2004).

These new instrumental and data analysis techniques also benefit from the development in instrumental and computational science. But also everyday life profits from space and Astrophysical instrument development, e.g. the Teflon-coated pan or the Ceran cooking field.

In Astrophysics one wants to learn something about a single Astrophysical object, a special class of Astrophysical objects or about the history and fate of the whole Universe (cosmology).

Today many features in the cosmic microwave background (CMB), which was detected in 1963 by Pencias and Wilson and is a relic of the Big Bang, the origin of the Universe, was or is observed with satellites like COBE, WMAP and the Planck satellite, which was launched recently on the 14th May 2009, to determine and improve the precision of the basic parameters of cosmology and also to test the standard model of particle physics.

Many philosophical questions are asked: Where are we? From where are we coming from? Are we alone? So the SETI project, the search for extraterrestrial intelligence, was started, where many, mostly amateur, radio telescopes collect data, which is then analysed for unusual signals that might come from extraterrestrial civilisations. In the SETI at home project many computers all distributed over the world help to analyse the data taken with the large Arecibo radio telescope. Up to now over 350 exoplanets, planets around other stars, were discovered using various methods: e.g. radial velocity or transit method.

Many phenomena in fundamental research, Astrophysics and particle physics, cannot be explained with Newtonian gravity anymore, one needs to use general relativity (GR), which was presented by Albert Einstein in 1915 (Einstein 1915). General relativity was tested many times. It explained an extra term in the perihelion precession of Mercury, predicted the gravitational light deflection, which was first measured in 1919 by Sir Arthur Eddington and his collaborators during a total solar eclipse.

Gravitational lensing, an effect of GR also based on light deflection, is used today e.g. to determine the mass of galaxies or to observe very distant objects inspecting the Einstein arcs and rings.

With the satellite Gravity Probe B general relativistic effects on rotating gyroscopes in the gravitational field of the Earth (see for example: Keil and Schartmann 2005) were measured 2005. The geodetic effect could be confirmed with an accuracy of less than 0.5%. But due to problems with noise in measuring the frame-dragging effect, also called the Lense–Thirring effect, the final results are still not published (in 2009).

In the recent years gravitational wave detectors, like the Laser Interferometer Gravitational-wave Observatory (LIGO), have been built which try to detect gravitational waves resulting from e.g. mergers of two black holes or neutron stars.

Compact objects are very exotic objects in the universe, they are so dense that they can only be (partially) explained by general relativity and quantum mechanics. To this class of objects belong white dwarfs, neutron stars and black holes.

White dwarfs are the final evolutionary states of normal stars with initial masses below about eight solar masses. These very hot objects consist mainly of carbon and oxygen and have a typical mass of 0.6 solar masses and a size of about the Earth (6371 km in radius), which gives a density of approximately one metric ton per cubic centimetre, after the helium burning stopped and the star collapsed until it is stabilized by the electron degeneracy pressure of the electron gas in which the nuclei swim.

Neutron stars, which are formed starting from normal stars with initial masses higher than about eight solar masses, are very dense objects which consist of exotic matter, which might be quark matter or Bose-Einstein condensate of K-Mesons, and have a mass between about 1.2 to 2.2 solar masses, a radius of only about 10 to 12 km and a surface temperature of about 10^6 K. This gives roughly the same density as in an atomic nucleus or one Earth mass per cube of edge length 400 metres. The

possible formation processes are: After the fusion burning of all elements down to iron, the stellar core collapses and the density and pressure of the inner core raise so high that the electrons are pressed into the protons to form neutrons by a process called inverse beta-decay. The resulting degeneracy pressure can stabilize the neutron star if it has a mass of about 1.2 to 2.2 solar masses, if the mass is higher the star collapses further and a black hole is formed. In the case of the neutron star formation, the still from outside in-falling matter is repelled from the newly formed hard surface of the neutron star and additionally heated by neutrinos which are a further output of the inverse beta decay. The out-flowing matter can be observed as a supernova explosion (of type Ib, Ic or II). The other possible formation process is as the end product of a supernova type Ia explosion, where a white dwarf, which accretes matter from a companion star, reaches the Chandrasekhar mass limit of 1.44 solar masses, and therefore is not stabilized anymore by the electron degeneracy pressure and collapses to a neutron star. Up to now only a few neutron stars are observed directly using X-ray satellites, see Weisskopf et al. (2007). If the neutron star is rapidly spinning with a period as short as a few milliseconds and it has a large magnetic field aligned at an angle relative to the spin axis, electrons can be excited by the periodic arrival of the magnetic field. This process leads to periodic pulses of synchrotron radiation in the radio and optical band of the spectrum. In this case one speaks of a pulsar. The first such object, the Hulse-Taylor binary pulsar, was detected in 1974 and gives a good testbed for general relativity.

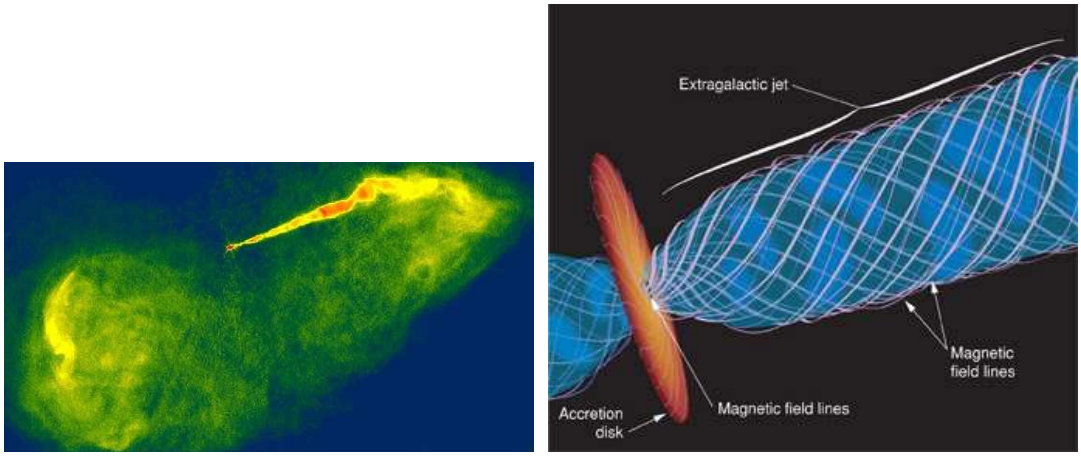
Black holes exist in a huge mass range: they can have masses starting from a few solar masses (stellar black holes) or maybe even less (mini black holes) up to several billion solar masses, the so-called super-massive black holes (SMBH), which usually are found in the centre of galaxies and might have formed by black hole mergers and accretion over billion of years.
Black holes are so massive and dense that the spacetime is extremely bended in the vicinity of such objects, that means that gravity is so strong that even light cannot escape from them. So one cannot look inside a black hole from outside, but one can study the influence on the dynamics of the surrounding matter and radiation.

Black holes can be characterized according to the so-called no-hair theorem by John Archibald Wheeler (see e.g. Taylor and Wheeler 1992): "Black holes have no hair!" by only a few parameters, which are the mass of the black hole, its angular momentum and charge. Actually these are three hairs.
In practice there will be no charged black hole because a non-negligible amount of charge would be immediately compensated by the surrounding charged matter.
Black holes are mathematically described by the vacuum solutions of Einstein's field equation: The Schwarzschild(Schwarzschild 1916) solution for stationary spherical symmetric black holes and the Kerr (Kerr 1963) solution for rotating black holes.

Cyg X-1, a very strong X-ray source, is a binary system, where a blue super-giant variable star orbits at about 0.2 AU around a black hole, which is also called a microquasar. Matter from the star forms an accretion disc around the black hole, which is then heated up to millions of Kelvin radiating X-rays. A pair of large jets, perpendicular to the accretion disc, is observed, too.

M 87, a giant elliptical galaxy, shows large X-ray and radio waves and also large gamma-ray emission, which is for example observed by the H.E.S.S. Cherenkov telescopes (Aharonian et al. 2006). It has a active galactic nucleus (AGN) with a super-massive black hole of about 6.4 billion (6.4×10^9) solar masses in its centre and also shows large jets.

Stellar dynamics around the centre of our galaxy, the Milky Way, give a hint that there is a giant super-massive black hole in the vicinity of Sgr A* (Ghez et al. 1998).



(a) *Large scale jets of M87 in radio (VLA image)*

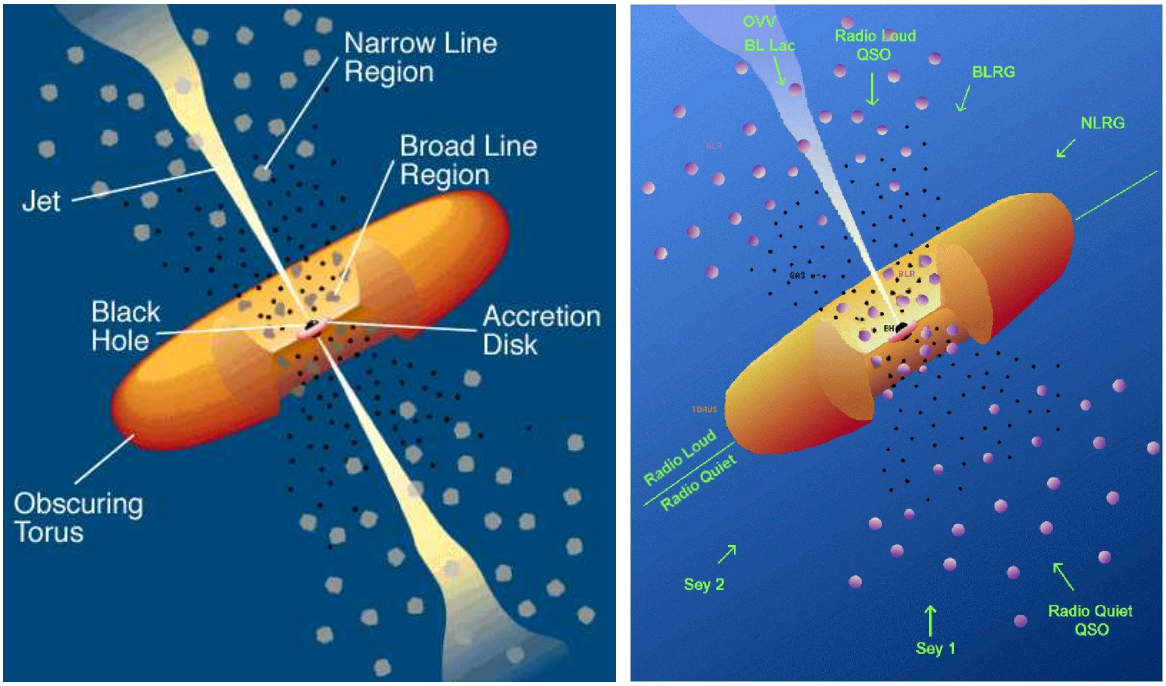
(b) *Sketch of the inner part of an AGN*

Figure 1.1: *Large scale structure of accretion phenomena from pc to Mpc scale (plots taken from Hilscher 2009)*

Active galactic nuclei (AGN), the central part of galaxies with a super-massive black hole in the centre surrounded by a large accretion torus, and their ultra-relativistic jets are observed up to a redshift factors of about 6, which makes them one of the most brightest phenomena in the Universe. According to the AGN unification theory, an AGN is called differently by observers depending on the angle of view and on the radio activity, see figure 1.2. AGN-jets are collimated outflows, which can extend up to a distance of several Mpc. These outflows are thought to be powered by energy extraction of a rotating black hole by the Blandford and Znajek (1977) process and then collimated by magnetic fields. But the detailed process is still unclear, especially the formation process of such jets from accreting matter in the ergosphere and the vicinity of the black hole.

Also the role of turbulence in accretion discs (Frank et al. 2002) is not fully understood: Turbulence is a mechanism to transport angular momentum outwards so that matter can be accreted inwards. Since the microscopic viscosity is too small to explain the observed accretion rate, Shakura and Sunyaev (1973) parameterized the turbulence in their standard α-disc model. The Magneto-Rotational Instability (MRI) by Balbus and Hawley (1991) (see also Balbus et al. 1996) provides a possible mechanism for the generation of turbulence. But since turbulence occurs instantaneous on various space and timescales it is very difficult to construct a proper numerical model or analytic description for it.

A runaway reaction of thermonuclear burning on the thin atmosphere of a neutron star can be observed as X-ray bursts, which show two prominent oscillations frequencies at about 10 Hz and about 1 kHz, the quasi-periodic oscillations, see Fig. (1.4). There are at least two different models that explain these oscillations: Stella and Vietri (1998) propose a precession effect of the innermost disc region due to the frame-dragging effect of the fast-rotating neutron star. Whereas Lamb and Miller (2003) explain these oscillations by an interaction between the sonic-point and the radiation pattern of clumps orbiting at the spin-resonance radius. From both models one can determine an estimate of the neutron star's mass and radius and they might give a hint on the interior structure and the equation of state of the neutron star.



(a) *AGN model showing its components* **(b)** *AGN unification model*

Figure 1.2: *Unification model of the active galactic nucleus (AGN) (see Urry and Padovani 1995). On the left a schematic diagram of an AGN with its components is shown: a central black hole surrounded by an accretion disc and an obscuring dust torus. Perpendicular to the accretion disc there is a pair of ultra-relativistic jets. The Broad Line Region (BLR) located near the black hole and the Narrow Line Region (NLR) located further outwards consist of clouds, which show line broadening in the spectra due to their fast movements. On the right figure the unification model of the AGN is depicted: Various different types of observed objects (radio-loud and radio-quiet, narrow-line and broad-line) are interpreted simply due to the angle at which the AGN is observed. If the torus blocks the light from the BLR, only narrow lines are seen in the optical spectrum one can observe either Seyfert 2 (Sey 2) galaxies or Narrow Line Radio Galaxies (NLRG). Seyfert 1 and Broad Line Radio Galaxies (BLRGs) show both narrow and broad components to their optical emission lines. Some objects, the radio loud objects (shown in the top half), emit strong radio signals, whereas others, the radio quiet objects (shown in the bottom half), show only low or no radio emission. BL Lacs, which show no, and OVVs (Optically Violent Variables), which show very weak emission lines, belong to a subgroup known as blazars, that are thought to be viewed along the ultra-relativistic radio jet.*

$\gamma-$ray bursts (GRBs), which are extragalactic and isotropically distributed explosion events, are one of the most luminous phenomena in the universe. Accidently such a huge explosion (GRB 971214) was observed together with a Supernova Ib explosion, suggesting that both phenomena might have the same origin, which leads to the collapsar model from Bodenheimer and Woosley (1983), where the core collapse of a rotating neutron star or a giant star to a black hole results in a supernova explosion. Due to the distribution of angular momentum a large torus is formed in the equatorial region and matter can escape along the axis and focused to form ultra-relativistic jets, where one of them might hit the Earth in form of a *gamma*-ray burst if the polar axis is directed towards the Earth. An explanation for short γ-ray bursts Brown et al. (2000) is the merging of two neutron stars or black holes, which additionally emits gravitational waves, that the LIGO consortium wants to detect (Cutler and Thorne 2002).

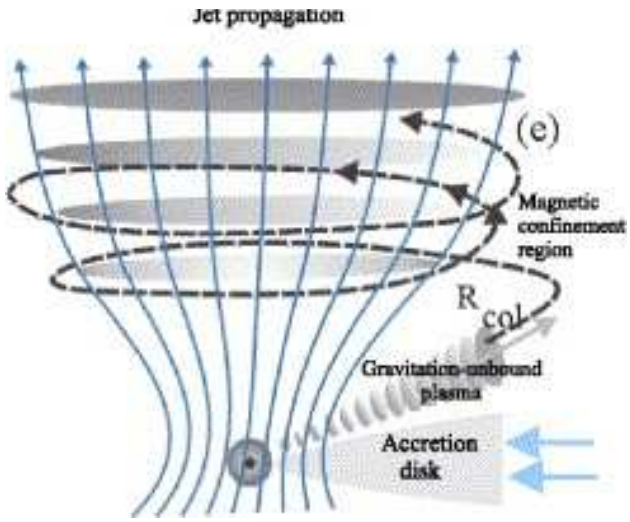


Figure 1.3: *Sketch of the Jet Launching Mechanism (plot from Hujeirat 2005a)*

In the field of Astrophysical Fluid Dynamics (AFD) one models Astrophysical flows of plasma and matter using a variety of possible methods to explain what is observed and predict what might be observed.

Due to the complexity of such phenomena one is severely limited if one would rely only on analytic solutions, so numerical methods play a more and more important role nowadays. Especially non-linear systems cannot really be studied using only analytical methods.

Different physical descriptions and numerical approaches are used: N-body codes, where one calculates the gravitational forces between the single objects, Boltzmann lattice codes, where one looks at the distribution function of the particles in phase space, smoothed particle hydrodynamics (SPH), where one has pseudo particles representing a fluid, spectral methods, particle in cell methods and grid based methods: finite element method (FEM), finite difference method (FDM) and finite volume method (FVM). All have their advantages and disadvantages and limits on applicability to a certain problem. There are a wide range of physical phenomena and processes to take into account in a realistic simulation of an Astrophysical flow, e.g. Hydrodynamics (HD), ideal Magneto-Hydrodynamics (MHD), effects of non-ideal MHD: finite conductivity, ohmic heating and Hall term, radiative processes and atomic and chemical networks and multi-component plasmas, all in Newtonian, special relativistic or general relativistic formulation. In Astrophysics various processes play a role with very different sizes and time scales and often initial and boundary conditions are not clearly given, therefore it is very complicated to do a really realistic simulation taking into account all physical processes. So one does simple approximations as a first step and then adds more complexity step by step.

To try to tackle these enormous problems the exponential development of computer systems and programming languages was and is very important:

The first very large computer systems at the end or the 1960s filled large halls and had the power that now easily fits in a small pocket calculator. In these days one used punchcards (in German: Lochkarten) to program the computer and for the output simple plotters, before the first terminals and then graphical output on a monitor was possible. Since then many things changed.

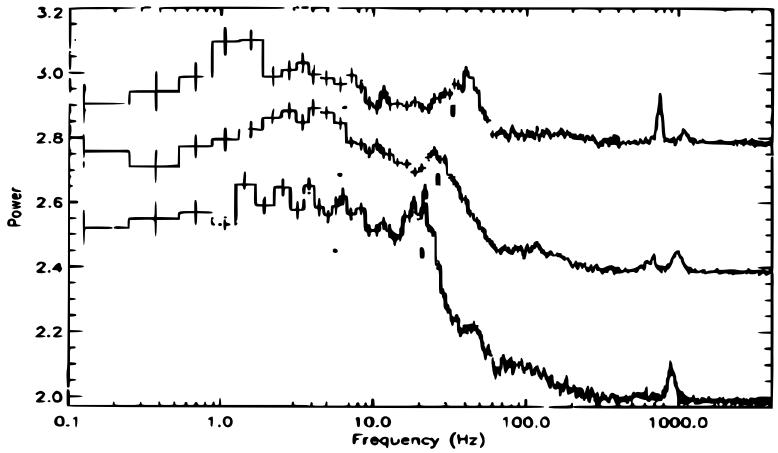


Figure 1.4: *Power spectrum of the light curves of 4U 1728-34. The spectra includes two peaks at about* 10 *Hz and about* 1 *kHz, the quasi periodic oscillations (plot from Stella and Vietri 1998).*

In 1982 the internet arised from the ARPANET, a project to connect research institutes and universities, which was started in 1969 by the Advanced Research Project Agency (ARPA) in the USA, by introducing the TCP/IP network protocol for communication between distant computers. In 1990 the internet was opened to the public and so it could be used by everyone. The World Wide Web (WWW) was developed in 1989 at CERN (Genf) by Tim Berners-Lee and 1993 the first graphic webbrowser could be freely downloaded. This development made it possible that basically everyone and not only specialists can use the Internet. Since then it developed very quickly. The development of the internet was tightly coupled to the Unix operating system, whose first version was constructed in 1969; it was developed at AT&T and freely distributed to universities, where it was also further developed as the Berkeley Software Distribution (BSD). But later in the early 1980s AT&T decided to commercialize Unix (system-V) and the source code was not published anymore. Many Unix derivates were appearing, for example AIX, HP-UX and Solaris. Since it is much better for research and development (and also for the development of the economy as a total) to have an open source code, the GNU-Projekt („GNU's Not Unix“) was founded in 1983 by Richard Stallman and in 1985 the Free Software Foundation (FSF), with the aim to have a free unix-compatible operating system. Until 1990 all basic parts of this system except of the kernel were developed. In 1987 Minix, a small unix clone which runs on normal PC hardware was developed by Andrew S. Tanenbaum at the University of Amsterdam. In 1991 Linus Torvalds published its first version of his newly created unix-like operating system called Linux. Other free unix-like operating systems are free BSD variants, Mac OS X and OpenSolaris. Now many Linux derivates are available and Linux can run on almost any computer platform, it is even used in embedded devices, e.g. multimedia players.

On the hardware side the development is also extremely: it changed the whole life, not only in science, industry and business (computer simulations, robotics, logistics, controlling, ...), but also in office and at home and during travel: Today home and office computers and notebooks already contain multi-core processors, mobile phones, media players, game controllers were developed and the Global Positioning System (GPS), and other satellite positioning systems, e.g. the European Galileo project, are applications of general relativity in 'everyday' life. In High Performance Computing (HPC) specialized hardware was developed, like the GRAPE board, to calculate the gravity force for

Smoothed Particle Hydrodynamics (SPH) simulations, scalar machines and vector machines, like the NEC SX-8 at the HLRS, the High Performance Computing Center in Stuttgart, were constructed and optimized. The vector machine power of graphics cards, which were vastly developed for use by computer games, can now also be used for computation: e.g. Nvidia TESLA graphics chips, which are specially designed for computation and can be programmed using Cuda, or similar chips by AMD. To have a common graphics card computation language now the OpenCL standard was published. Another way to speed up often used program parts is to use programmable chips, e.g. FPGEs. Now multi-core systems in clusters with fast interconnects, like Gigabit ethernet or Myrinet, are used. Parallel file systems were developed to have a fast I/O system for data storage.

But not only the hardware has been developed also the programming languages: One possibility to program a computer is to use assembler, a machine near computer language, which is very fast, but not easy to program and not portable, because it is machine specific. Therefore high level programming languages were developed, where the programs have to be compiled to binary code, before they can be executed. The first high level programming language was Plankalkül, developed by Konrad Zuse in 1946. In 1952 Grace Hopper developed the first compiler called A-0, and already 1957 the first FORTRAN-Compiler appeared. In 1972 the programming language C was presented by Dennis Ritchie, whereas C++ was invented in 1983 by Bjarne Stroustrup. Scripting languages, like Perl (1987), Tcl (1988) and Python (1991), were developed, which consist of an interactive interpreter, which analyses the script at runtime and possibly uses C and Fortran code underneath. So it is possible to program very quickly and test the program without compilation. Relatively modern is the concept of Object Oriented (OO) Programming, which is in contrast to the conventional procedural programming an other way to construct a program. OO-programming can for example be done with C++, which is an extension to C. Many concepts of it are also introduced into other already existing programming languages like Perl and even FORTRAN (in the standards Fortran 90/95 and Fortran 2003). One basic concept of Object Oriented Programming is to have objects which belong to classes with specified properties and methods. But a object can belong to a certain subclass, which inherits all properties and methods of the higher class. With a well-thought-out program design it is then relatively easy to extend a program. But for problems with time-critical issues there might be — depending on the programming — too much overhead by the Object Oriented Programming. Up to now Object Oriented Programming with classes plays a minor role in writing scientific simulation codes, since the routines which do the actual computational calculations, which are called methods in the OO language, are the same, only the structure and organisation of the code is different. But that might change in the future, since one might use and already uses OO-programming or some concepts of it to make it easier to extend simulation codes.

In the future, where the solution of larger problems is increasingly more significant, the parallelization of a numerical code plays a crucial role: For shared memory machines one can use for example OpenMP and other threading directives, with which one very easily can parallelize an already existing code in total, but also only a part of it. So it is possible to do the parallelization step by step. But for distributed memory machines, like most High Performance Supercomputer Systems and Linux Clusters, for parallelization the Message Passing Interface (MPI), which is the de-facto standard, has to be used. The MPI-library defines special functions which are called from C or FORTRAN in between the usual other code segments. Here compared to OpenMP the parallelization of an existing code is not so easy, since one might have to restructure the whole code. One needs to invest a large amount of programming time, to get out the right results, since this cannot be done in small steps and therefore is very error-prone. The goal here is to distribute the workload evenly to the processes (this is also important on shared memory systems) and to minimize the communication time between the

processes, especially between processes on different nodes. The MPI parallelization gives the most flexibility, since the code can either run on distributed memory machines but also on shared memory machines. A code parallelized with OpenMP can only be run on shared memory machines. There is also the possibility to combine the two parallelization methods MPI and OpenMP to do a hybrid parallelization.

Also the progress made in the development of numerical methods and simulation codes in the last decades is extreme: The development of modern computational fluid dynamic codes started in 1959, where S. K. Godunov (Godunov 1959) presented a conservative numerical scheme for solving partial differential equations, the so-called Godunov method. The Godunov method is a three step grid-based method: First the variable values at the grid interfaces between the cells are reconstructed from the piecewise constant cell values, then on each cell interface a local Riemann problem is solved and finally one takes the averages of the solution to get a single averaged cell value for the next time step. This first order scheme was extended to higher orders using linear and piecewise parabolic reconstruction (Colella and Woodward 1984). To avoid numerical artificial oscillations which result in a numerical breakdown of the higher order method, one has to apply slope or flux limiters in the process of reconstruction to guaranty that the scheme is total variation diminishing (TVD). The first such higher-order scheme, the Monotone Upstream-centred Schemes for Conservation Laws (MUSCL), which has second order spatial accuracy, was presented in 1979 by Bram van Leer (van Leer 1979). In the field of Newtonian (magneto-) hydrodynamics up to now several quite sophisticated High Resolution Shock Capturing (HRSC) schemes, e.g. the PLUTO code (Mignone et al. 2007), have been developed using different exact and approximate Riemann solvers, like Lax-Friedrichs, Roe and hllc, and slope or flux limiters like minmod and suberbee and also different time advancement methods, for example Runge Kutta integration, are used to improve the accuracy of the solution.

General relativistic numerical hydrodynamic codes (a good overview can be found in Font 2008) are mostly based on the 3+1 (ADM Arnowitt et al. 1962) formulation, where spacetime is foliated into non-intersecting spacelike hypersurfaces.

Already in 1966 May and White (May and White 1966) developed a time-dependent one-dimensional general relativistic code describing an adiabatic spherical collapse using Lagrangian coordinates (these are coordinates which are moving with the fluid element).

Multi-dimensional numerical general-relativistic hydrodynamics in Eulerian coordinates started to develop with the pioneering work of Wilson in 1972 (Wilson 1972). He introduced basic dynamic variables representing the relativistic density, momenta and the generalized internal energy. Even in the case of Cartesian coordinates this system of equations is not in a strictly conservative form since pressure gradients are treated as source terms. The conservative formulation of nonlinear hyperbolic systems is very important to guarantee the correct jump conditions and shock speeds. Therefore using Wilson's scheme one has to introduce artificial viscosity terms, numerical dissipation, to damp the oscillations and stabilize the solution near discontinuities. Many general-relativistic codes, both solving Einstein's Field equations and using a fixed background metric of a Schwarzschild or Kerr black hole, use the Wilson scheme and with them one can study many Astrophysical problems containing for example also the head-on collision of two black holes. But there are also some severe limitations in simulating ultra-relativistic flows with Eulerian Lorentz factors larger than about two: Norman and Winkler (Norman and Winkler 1986) found out by studying special relativistic problems, like the relativistic shock reflection problem, that the amount of errors occurring for high Lorentz factors mainly

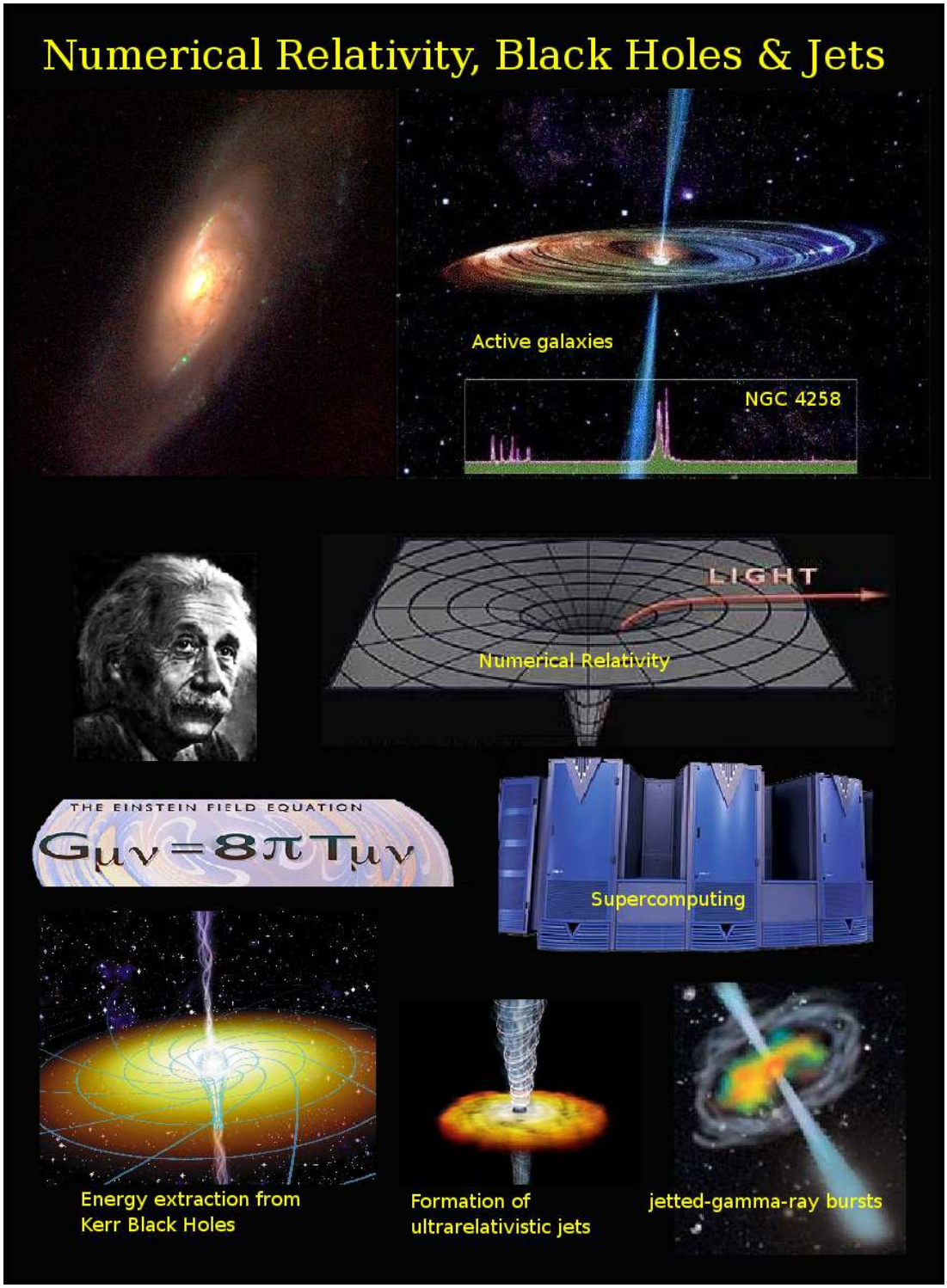


Figure 1.5: *Numerical Relativity, Black Holes and Jets (from the GR-I-RMHD (General Relativistic - Implicit - Radiative Magneto-HydroDynamics) project proposal to the Klaus-Tschira-Foundation, November 2006 (collage by B. W. Keil)*

depend on the way in which the artificial viscosity terms are included in the Wilson scheme. The artificial viscosity terms should be implemented in a consistent way into the equations, in order to consider the artificial viscosity as a real viscosity, which occurs in the equations like additional pressure terms and should not be omitted in the calculation of the relativistic enthalpy as in the original Wilson formulation. The resulting equations are highly nonlinear due to the nonlinear function of artificial viscosity and the direct occurrence of the Lorentz factor in the convective terms and are therefore much more coupled than their Newtonian counterparts. So Norman and Winkler proposed the use of implicit schemes to describe this coupling more accurately and developed a one-dimensional code (Winkler et al. 1984) in flat spacetime using an adaptive grid which reproduced very accurate results up to Lorentz factors of about ten.

In 2003 Anninos and Fragile (Anninos and Fragile 2003) used their explicit three-dimensional Cartesian code cosmos to compare state-of-the-art artificial viscosity schemes with high-order non-oscillatory central schemes. Simulating shock tube tests and shock reflection tests, they confirmed earlier results for artificial viscosity schemes: the numerical solution becomes increasingly unstable for shock velocities greater than about 0.95 the speed of light. But instead using the conservative formulation of the high-order non-oscillatory central schemes they were able to handle ultra-relativistic flows up to very large Lorentz numbers.

In 1991, Martí, Ibáñez and Miralles (Martí et al. 1991) presented a new formulation of the general relativistic hydrodynamics equations, also called the Valencia formulation, which is a conservative Eulerian formulation using the total energy instead of the internal energy equation. But the strict conservation form is only possible in flat spacetime and the recovery of the primitive variables (density, velocities and internal energy) from the conservative ones (relativistic density, momenta and total energy) might be problematic in some cases, e.g. in the calculation of the internal energy (or the pressure) one might have to subtract some very large and almost similar numbers which numerically can lead to a very non-accurate result, which in the extreme case might be negative and causes the simulation code to crash. Apart from such problems this formulation is in general very accurate using state-of-the-art high-resolution shock capturing (HRSC) schemes with exact or approximate Riemann solvers and avoids to fine-tune an artificial viscosity parameter, since artificial viscosity is not necessary in this formulation.

There exist also some more general formulations, covariant approaches, that are not restricted to the spacelike foliation of the 3+1 split of spacetime (see e.g. Font 2008).

But up to now almost only time-explicit methods solving the hydrodynamic Euler equations have been used in Astrophysical Fluid Dynamics (AFD). The reason for this might be, that almost all Astrophysical phenomena are studied up to now by (simple) approximations which only use the hydrodynamic Euler equations without diffusion and without sophisticated radiative effects and only time-dependent compressible flows were be looked at. For such strongly time-dependent and compressible flows, explicit methods are very well suited. But to do a better approximation to nature one wants to simulate a very complex flow with magnetic fields, viscosity, radiative effects and with atomic and chemical reactions. Also diffusive and viscous effects may play an important role, so one should solve the Navier-Stokes equations instead of the Euler equations. The different physical processes posses different time scales and the smallest timescale dictates the Courant-Friedrichs-Levy (CFL) time step size, the outermost limiting time step up to which explicit methods are numerically stable. Larger time

steps lead to a numerical breakdown. For example in the case of the important Astrophysical problem of the gravitational collapse in the process of star formation, the explicit time-step size decreases enormously, and therefore the calculations stagnate in time.

But in contrast implicit methods are unconditionally stable, and can therefore use a larger time step. So only implicit methods have the potential to overcome the so-called time-step crisis which occurs in the explicit calculations of the gravitational collapse. The time step should for physical reasons not made too large, so that all physical relevant properties of the flow are still resolved in time. Another advantage of implicit methods is that there is a tighter coupling of the equations, which is very important for an accurate description of non-linear effects.

Unfortunately one has to pay a price for the use of implicit methods: Since the solution does not depend only on neighbouring cell values from the present time step, but also on the values of the future time step, the equations at each grid point are coupled together and form a large linear system of equations. Fortunately this system possesses a highly sparse and well structured matrix. In the recent decades also huge progress was made in development of fast methods to solve large sparse systems of equations, especially in the field of Krylov subspace iterative methods, for example: GMRES (1986), BiCGSTAB (1992) and TFQMR (1993).

Ideally one would have an ultimate black-box algorithm, a Swiss Army knife algorithm (in German: "Eierlegende Wollmilchsau"), which contains numerical solvers that are unconditionally stable, robust, efficient, Newtonian, special and general relativistic and capable of treating flows that range from strongly compressible to almost incompressible, are self-gravitating, radiating, magnetized multi-component-plasmas taking into account atomic and chemical reactions and solving the equations with high spatial and temporal accuracy on unstructured meshes. A further ingredient of this ultimate black-box algorithm would be that depending on the physical properties of the flow the optimal method is selected automatically. But unfortunately that is not the case: doing numerical simulations is not a 'push button' technology. But maybe that is good so, because elsewise no research and no improvement would be done anymore.

To have such an ultimate goal, which probably can never be reached entirely, is similar to the concept of physics in total: one makes a model of nature which describes nature up to a specific accuracy. Then one measures or observes a contradiction or a deviation to the model. Thereafter one refines the model to describe the measurements or observations. And with this method one gets iteratively closer to the final aim, the right description of nature, but one will never be able to exactly describe nature in total. Even if one could describe it exactly, e.g. if one would have the 'world formula', one would not be able to calculate everything with it.

To have a better description of the Astrophysical flows one needs to include more physical processes into the simulations, which in general possess very different time scales and so implicit methods are very important in the solution of these phenomena, because time-explicit methods are time-step limited.

To have an easy to use implicit simulation program well suited for general relativistic Astrophysical problems, **Astro-GRIPS**, the **G**eneral **R**elativistic **I**mplicit **P**arallel **S**olver, the numerical simulation code described here, was developed. This simulation code solves the general relativistic hydrodynamic Euler- or Navier-Stokes equations under the assumption of a fixed background metric of a static Schwarzschild or rotating Kerr black hole. It can be used very flexible and can be run on various computer platforms, compute clusters and high performance computers due to its MPI-Parallelization,

which was a major task of this work. Due to the implicit method the simulation code is unconditionally stable and also takes into account the non-linearity of the fluid flows by coupling of the equations. The Hierarchical Solution Scenario (HSS: Hujeirat 2005a) can be used to find a stationary or quasi-stationary solution quickly by gradual enhancement of the equation coupling.

The development of such an implicit simulation code gives the basis to include and study the influence of interesting physical aspects like magnetic fields and radiative transfer, to better understand the relativistic flows around compact objects, especially the formation and acceleration of ultra-relativistic multi-component plasma MHD-jets and compare the numerical results with Astrophysical observations.

2 General Relativity and Fluid Dynamics

2.1 General Relativity

2.1.1 Basic Ideas and Equations

Since one intents to study the plasma flows in the vicinity of black holes and other compact object where flow velocities near the speed of light can occur and where gravity is very strong, it is not possible anymore to use the classical Newtonian formulation of the hydrodynamic equations. Due to the strong gravitational field it is also not sufficient to use special relativity, since many physical effects can only be explained by general relativity.

In general relativity gravity is not described as a force like in Newtonian mechanics, instead it is the result of spacetime curvature.

The Riemann curvature tensor is defined as:

$$R^{\lambda}{}_{\mu\nu\rho} = \Gamma^{\lambda}{}_{\mu\rho,\nu} - \Gamma^{\lambda}{}_{\mu\nu,\rho} + \Gamma^{\zeta}{}_{\mu\rho}\Gamma^{\lambda}{}_{\zeta\nu} - \Gamma^{\zeta}{}_{\mu\nu}\Gamma^{\lambda}{}_{\zeta\rho}, \tag{2.1}$$

where $\Gamma^{\rho}{}_{\mu\nu}$ are the connection coefficients, also called the Christoffel symbols of the second kind, which have to be taken into account if one transports a vector from one point to another point in curved space on different paths and result from the parallel transport of a vector in a curved space on a closed path, a loop, and are defined as:

$$\Gamma^{\rho}{}_{\mu\nu} = \frac{1}{2} g^{\rho\eta} \left(g_{\eta\nu,\mu} + g_{\mu\eta,\nu} - g_{\mu\nu,\eta} \right), \tag{2.2}$$

where $g^{\mu\nu}$ is the metric of the spacetime, which is symmetric $g^{\mu\nu} = g^{\nu\mu}$ (and torsion free). Distances are measured according to the square of the line element $ds^2 = g_{\mu\nu} dx^{\mu} dx^{\nu}$, where x^{μ} is the 4-vector of the location in spacetime which is given by $x^{\mu} = \left(ct, \vec{x}\right)$.

By contraction of the the Riemann curvature tensor one gets the Ricci tensor $R_{\mu\nu} = R^{\lambda}{}_{\mu\nu,\lambda}$ and its (the Ricci tensors) trace is the Ricci scalar R which is given by $R = R^{\mu}_{\mu} = g_{\mu\nu} R^{\mu\nu}$.

To get a metric theory of gravitation Albert Einstein constructed a tensor, the Einstein tensor $G^{\mu\nu}$, that is divergence free: $G^{\mu}_{\nu;\mu} = 0$ and is defined by: $G^{\mu\nu} = R^{\mu\nu} - \frac{1}{2} g^{\mu\nu} R$ and represents the curvature of spacetime.

Another important ingredient of relativity is the stress-energy tensor $T^{\mu\nu}$, also called energy-momentum tensor, which represents the density and flux of energy, including the relativistic rest mass energy, and momentum. The equations of motion in a local inertial frame, in which gravity is absent, follow from the condition that its covariant derivative is zero $T^{\mu\nu}_{;\nu} = 0$.

Combining all ingredients results in the Einstein's field equations (Einstein 1915), which written in the compact tensorial form are:

$$G^{\mu\nu} = -\frac{8\pi G}{c^4} T^{\mu\nu}, \tag{2.3}$$

where G denotes the gravitational constant.
John Archibald Wheeler noted a nice and simple explanation for this formula, which in compact form describes the whole concept of general relativity:

> "Spacetime tells mass how to move and mass tells spacetime how to curve."

Due to the complicated structure of the Einstein Equations, they form a system of coupled non-linear differential equations, only a few analytic solutions are known.
Already in 1916, the year after Einstein had published his theory of general relativity, Karl Schwarzschild (Schwarzschild 1916) found the spherical symmetric vacuum solution for a non-charged and non-rotating black hole, called the Schwarzschild solution. The other important solution for black hole astrophysics was found in 1963 by the New Zealand mathematician Roy P. Kerr (Kerr 1963) and is the stationary and axisymmetrical vacuum solution for a rotating, non-charged black hole.

2.1.2 Divergence of Vector and Tensor Fields

The covariant divergence of a vector field V^ν can always be expressed without referring to Christoffel symbols:

$$V^\nu_{;\nu} = \frac{1}{\sqrt{-g}} \partial_\nu \left(\sqrt{-g} V^\nu \right) \tag{2.4}$$

For the divergence of a (2,0)-tensor

$$T^{\mu\nu}_{;\nu} = \partial_\nu T^{\mu\nu} + \Gamma^\mu_{\mu\alpha} T^{\alpha\nu} + \Gamma^\nu_{\mu\alpha} T^{\mu\alpha} \tag{2.5}$$

follows if T is antisymmetric $T^{\mu\nu} = A^{\mu\nu} = -A^{\nu\mu}$:

$$A^{\mu\nu}_{;\nu} = \frac{1}{\sqrt{-g}} \partial_\mu \left(\sqrt{-g} A^{\mu\nu} \right) \tag{2.6}$$

and if T is symmetric (e.g. the energy-momentum tensor) $T^{\mu\nu} = T^{\nu\mu}$:

$$T^{\mu\nu}_{;\nu} = \frac{1}{\sqrt{-g}} \partial_\mu \left(\sqrt{-g} T^{\mu\nu} \right) + \Gamma^\mu_{\nu\alpha} T^{\nu\alpha} \tag{2.7}$$

2.2 Rotating Black Holes: the Kerr solution of Einstein's Field Equation

Black holes can be characterized by the so-called 'no-hair theorem' by John Archibald Wheeler "Black holes have no hair!" by only a few parameters, which are the mass of the black hole, its angular momentum and charge. Actually these are three hairs. In practice there will be no charged

black hole because a non-negligible amount of charge would be immediately compensated by the surrounding charged matter.
The Kerr solution, which was found 1963 by the New Zealand mathematician Roy Patrick Kerr (Kerr 1963), is a vacuum solution of Einstein's field equation, which is stationary and rotationally symmetric and has two free parameters, the mass of the central object $M_{\rm BH}$ and the Kerr rotation parameter a. Due to the Robinson theorem (Robinson 1975) the Kerr solution is the unique solution of stationary axisymmetric solutions of the general relativistic vacuum field equations which are asymptotically flat, that means that for large radii the Kerr metric goes to the flat Minkowski metric[1], have a smooth convex horizon and is regular outside the horizon and is uniquely defined by the two aforementioned parameters.

Adopting the 3+1 split of spacetime, a line element ds (dimension: L (length)) with the metric signature $(-,+,+,+)$ can be written as follows:

$$ds^2 = c^2 d\tau^2 = g_{\mu\nu} dx^\mu dx^\nu = -\alpha^2 (cdt)^2 + h_{\rm ik}(dx^i + \beta^i cdt)(dx^k + \beta^k cdt) \tag{2.9}$$

where $dx^0 = cdt$ and Greek indices run from 0 to 3 and Latin indices from 1 to 3.

For the Kerr metric in Boyer-Lindquist coordinates (Boyer and Lindquist 1967) $(ct,\ r,\ \theta,\ \phi)$ with dimensions $[L, L, 1(radians), 1(radians)]$, the most famous used coordinates, the line element reads:

$$ds^2 = -(\alpha^2 - \beta_\phi \beta^\phi)(cdt)^2 + 2\beta_\phi d\phi cdt + h_{\rm ik} dx^i dx^k, \tag{2.10}$$

which corresponds to a spacetime metric represented in the following matrix notation:

$$g_{\mu\nu} = \begin{pmatrix} g_{tt} & 0 & 0 & g_{t\phi} \\ 0 & g_{rr} & 0 & 0 \\ 0 & 0 & g_{\theta\theta} & 0 \\ g_{\phi t} & 0 & 0 & g_{\phi\phi} \end{pmatrix} = \begin{pmatrix} \beta_\phi \beta^\phi - \alpha^2 & 0 & 0 & \beta_\phi \\ 0 & h_{rr} & 0 & 0 \\ 0 & 0 & h_{\theta\theta} & 0 \\ \beta_\phi & 0 & 0 & h_{\phi\phi} \end{pmatrix}, \tag{2.11}$$

The coefficients $g_{\mu\nu}$ in the Boyer-Lindquist coordinates [with dimensions: $g_{tt} : 1$, $g_{rr} : 1$, $g_{\theta\theta} : L^2$, $g_{\phi\phi} : L^2$, and $g_{t\phi} = g_{\phi t} : L$] and their related functions, **not setting the speed of light c and the gravitational constant G to unity**[2], are defined as follows:

$$\begin{array}{lcll} \alpha & = & \frac{\bar{\rho}}{\Sigma}\sqrt{\Delta}, & [\alpha : 1] \\ h_{rr} & = & \frac{\bar{\rho}^2}{\Delta},\ h_{\theta\theta} = \bar{\rho}^2,\ h_{\phi\phi} = \bar{\omega}^2 \ \text{ with } \bar{\omega} = \frac{\Sigma}{\bar{\rho}}\cos\theta, & [h_{rr} : 1, h_{\theta\theta} : L^2, h_{\phi\phi} : L^2, \bar{\omega} : L] \\ \beta^r & = & \beta^\theta = 0,\ \beta^\phi = -\frac{\omega_{FDE}}{c} & [\beta^\phi : L^{-1}, \omega_{FDE} : T^{-1}] \\ \beta_\phi & = & g_{\phi i}\beta^i = g_{\phi\phi}\beta^\phi = \bar{\omega}^2\beta^\phi & [\beta_\phi : L] \\ \Upsilon & = & \frac{\bar{\rho}^2\Sigma^2}{\Delta}\cos^2\theta, & [L^4] \\ \sqrt{-g} & = & \bar{\rho}^2\cos\theta = \alpha\sqrt{\Upsilon} & [L^2]. \end{array} \tag{2.12}$$

[1] The Minkowski tensor in Cartesian coordinates has the form

$$\eta = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{2.8}$$

[2] many authors set $c = 1$ and $G = 1$, but to better understand the units of the variables and the scaling in the code (see later) here c and G are not set to zero and in most cases the dimensions/units are given in square brackets.

and the auxiliary functions:

$$\begin{array}{lcll} \Delta & = & r^2 - 2r_g r + (ar_g)^2 & [L^2] \\ \bar{\rho}^2 & = & r^2 + (ar_g)^2 \sin^2\theta & [L^2] \\ \Sigma^2 & = & (r^2 + (ar_g)^2)^2 - (ar_g)^2 \Delta \cos^2\theta & [L^4] \end{array} \tag{2.13}$$

with the gravitational radius $r_g = \frac{GM_{BH}}{c^2}$ $[L]$ where c, M_{BH}, G are the speed of light, mass of the black hole and the gravitational constant, respectively. α is the redshift or lapse function, which describes the gravitational redshift or relativistic time dilatation of local clocks as compared to infinity, and β is the shift vector function.
The cylindrical radius $\bar{\omega}$ is called so because $2\pi\bar{\omega} = 2\pi\sqrt{g_{\phi\phi}}$ is the circumference of cylinders at the radial position r that are concentric to the axis of symmetry.
Δ is called the horizon function and in the equatorial plane $\bar{\rho}$ equals the radial coordinate r.
In writing these expressions, the coordinate transformation $\bar{\theta} = \pi/2 - \theta$ was used, where the latitude θ is used instead of the polar distance angle $\bar{\theta}$; hence the appearance of "cos" instead of "sin" in the metric terms.
g is the determinant of the 4-metric whereas Υ is the determinant of the 3-metric. The contravariant components of the metric, which can be derived by the property that the metric satisfies $g^{\mu\nu}g_{\nu\alpha} = \delta^{\mu}_{\alpha}$ [1], are:

$$\begin{cases} g^{tt} & = & -\frac{1}{\alpha^2} & [1] \\ g^{t\phi} & = & \frac{\beta^{\phi}}{\alpha^2} = -\frac{\omega_{FDE}}{(c\alpha^2)} & [L^{-1}] \\ g^{rr} & = & \frac{1}{g_{rr}} = \frac{\Delta}{\bar{\rho}^2} & [1] \\ g^{\theta\theta} & = & \frac{1}{g_{\theta\theta}} = \frac{1}{\bar{\rho}^2} & [L^{-2}] \\ g^{\phi\phi} & = & \frac{1}{\bar{\omega}^2} - \frac{\beta^{\phi}\beta^{\phi}}{\alpha^2} & [L^{-2}]. \end{cases} \tag{2.14}$$

The radial dependence of the Boyer-Lindquist functions can be seen in figure 2.1 for several Kerr parameters a.

Horizons and Ergosphere

Δ is the so-called horizon function and is zero at the horizon, where the redshift factor α vanishes, that means the redshift z is infinity for an observer located at infinity, where the redshift is defined as follows:

$$z + 1 = \frac{\lambda_{\text{obs}}}{\lambda_{\text{em}}} = \sqrt{\frac{g_{tt}(x_{\text{obs}})}{g_{tt}(x_{\text{em}})}} \tag{2.15}$$

(for Schwarzschild: $z + 1 = \frac{1}{\alpha(r)}$)
with emitted wavelength λ_{em} of a signal and at infinity observed wavelength λ_{obs}. This behaviour explains the name "black hole", since no emitted light from inside the horizon is observed at infinity.

In the Kerr case ($a \neq 0$) there exist two horizons: the **Cauchy horizon** $r_- = r_g(1 - \sqrt{1 - a^2})$, the inner horizon, and the **event horizon** $r_+ = r_g(1 + \sqrt{1 - a^2})$, which is the outer horizon. Physically meaningful solutions have to posses a horizon, so the parameter a is limited by $|a| \leq 1$. An interesting

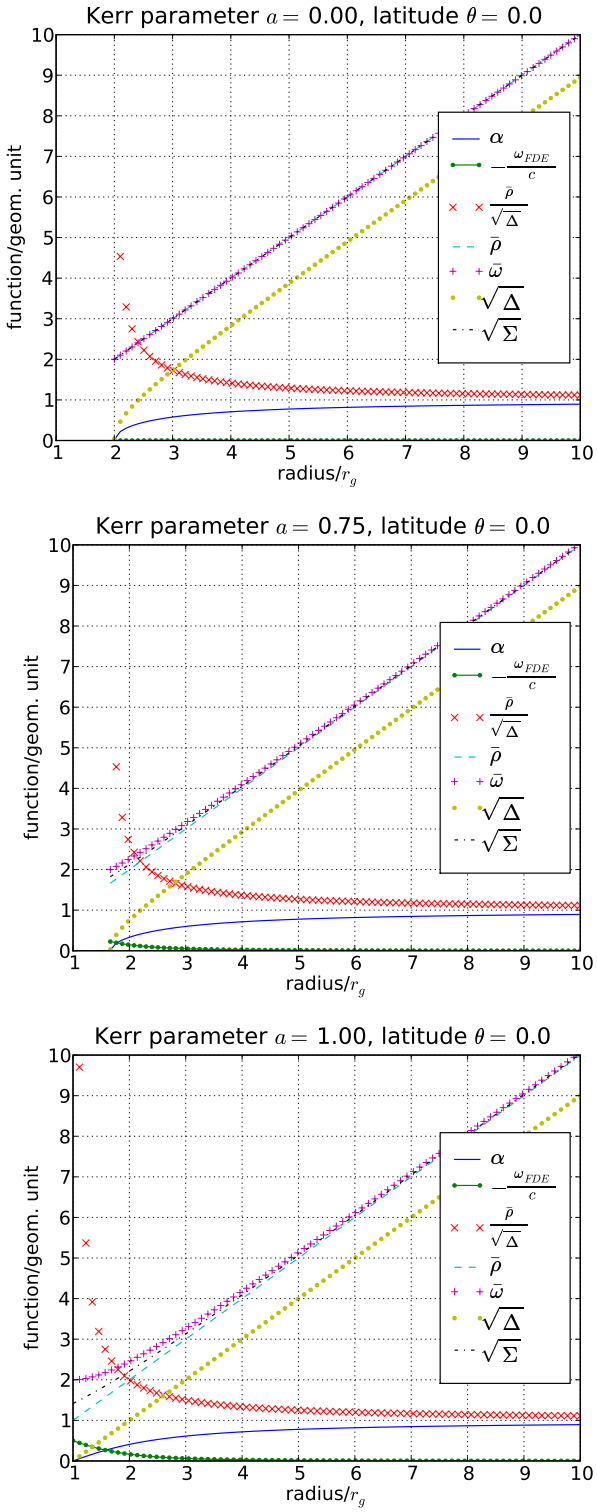


Figure 2.1: *Radial dependence of the Boyer-Lindquist functions for the Schwarzschild black hole $a = 0$, an intermediate Kerr black hole with $a = 0.75$ and for the extreme Kerr hole $a = 1$. The lapse or redshift function α is in units of 1, the frame dragging frequency divided by the speed of light $-\frac{\omega_{FDE}}{c}$ in units of r_g and the function $\sqrt{g_{rr}} = \frac{\bar{\rho}}{\sqrt{\Delta}}$ in units of 1; $\sqrt{g_{\theta\theta}} = \bar{\rho}$, $\sqrt{g_{\phi\phi}} = \bar{\omega}$, $\sqrt{\Delta}$ and $\sqrt{\Sigma}$ have all units of r_g. In the case of a Schwarzschild black hole $a = 0$, there is no frame dragging frequency $-\frac{\omega_{FDE}}{c} = 0$, $\bar{\rho} = \sqrt{\Sigma} = r_g$ and $\bar{\omega} = r_g \cos(\theta)$. With increasing Kerr parameter a from 0 to 1 the horizon, the radius at which the horizon function Δ is zero, moves from $2r_g$ further inwards up to one r_g.*

phenomenon occurs at the Cauchy horizon when it is reached from the outside: an observer witnesses the entire history of the external world in a single flash and due to the infinite blueshift the observer is hit by an infinite flash of radiation. These strange effects are a consequence of pure general relativity and may give a hint that in reality quantum effects may play a role in the interior of the black hole. For the extreme Kerr solution $a = 1$, almost all radii are equal to the gravitational radius r_g, whereas for the Schwarzschild solution $a = 0$ there is only one event horizon which is at the Schwarzschild radius $r_S = 2\,r_g$.

The ergosphere is the surface where the metric component g_{tt} vanishes:

$$g_{tt} = \beta_\phi \beta^\phi - \alpha^2 = \bar{\omega}^2 \left(\frac{\omega_{FDE}}{c}\right)^2 - \alpha^2 = 0 \quad \Leftrightarrow \quad r_E = r_g(1 + \sqrt{1 - a^2 sin^2\theta}) \tag{2.16}$$

It is dependent on the latitude θ and has an oblate shape which touches the event horizon. The zone between the ergosphere and the event horizon is called the ergoregion, where the rotation of the spacetime becomes extreme. It is not possible to sustain static observers inside the ergosphere, hence the ergosphere is also called the static limit, where $\omega = \frac{d\phi}{dt} = 0$. The ergoregion is also of importance for the energetics of the processes around a black hole, since Penrose (1969) showed that energy can be extracted from every spacetime which possesses an ergosphere.

Frame-Dragging Effect

The parameter a with dimension [1] is the Kerr or spin parameter which specifies the specific angular momentum (angular momentum per mass) with respect to the maximum specific angular momentum $r_g c = \frac{GM_{BH}}{c}$ and can have values between -1 and $+1$, negative values give retrograde and positive prograde rotation. For $a = 0$ one gets the Schwarzschild solution of a non-rotating black hole, which is due to Birkhoff's theorem the unique spherical symmetric vacuum solution.
The frame-dragging effect, also called Lense-Thirring effect, is the property that spacetime is dragged along with a rotating object around the rotation axis and results from the non-diagonal elements $g_{\phi t}$ and $g_{t\phi}$ of the Kerr metric in Boyer-Lindquist coordinates. The frame-dragging angular frequency is defined as $\omega_{FDE} = 2ar_g c\frac{r_g r}{\Sigma^2} = -\frac{g_{t\phi}}{g_{\phi\phi}}c\,[T^{-1}]$. Note that in the expression of β^ϕ there is a minus sign for the metric signature (-,+,+,+) whereas for the metric signature (+,-,-,-) there occurs a plus sign.

In the ergoregion, inside the ergosphere, everything is rotating due to the frame-dragging effect: observers, photons, magnetic field lines and even spacetime itself. Since the velocity field must be globally space-like (see also the normalisation condition of the 4-velocity and setting V^r and V^θ to zero):

$$g_{tt} + 2g_{t\phi}\frac{\omega}{c} + g_{\phi\phi}(\frac{\omega}{c})^2 < 0, \tag{2.17}$$

there is a minimum and maximum limit of the angular velocity ω:

$$\omega_- \leq \omega \leq \omega_+ \quad \text{with} \quad \omega_- = \omega_{FDE} - \frac{c\,\alpha}{\bar{\omega}} \quad \text{and} \quad \omega_+ = \omega_{FDE} + \frac{c\,\alpha}{\bar{\omega}} \tag{2.18}$$

The dependence of ω_{FDE}, ω_- and ω_+ on r can be seen in figure 2.2

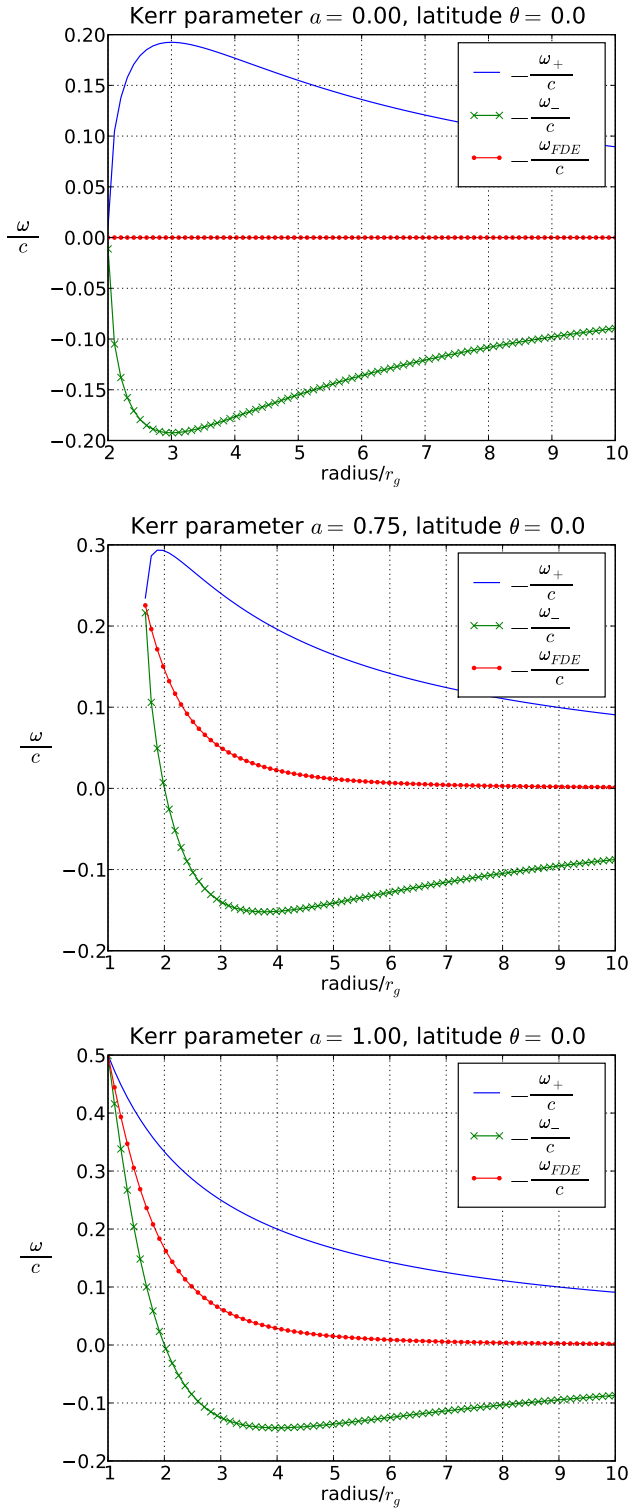


Figure 2.2: *Radial dependence of the frame-dragging angular frequency ω_{FDE} and the minimal and maximal allowed angular frequencies ω_- and ω_+ for the Schwarzschild black hole $a = 0$, an intermediate Kerr black hole with $a = 0.75$ and for the extreme Kerr hole $a = 1$. At the horizon all three functions meet at one point, so all matter, photons and even spacetime has to rotate with the frame-dragging frequency there.*

In the Kerr space there exist special non-static observers, called ZAMOs (zero angular momentum observers), with angular velocity $\omega = \omega_{FDE}$ (angular velocity with respect to the fixed stars, the usual static Bardeen observer at infinity) and vanishing specific angular momentum $M_\phi = 0$:

$$\begin{aligned} M_\phi &= \overline{\mathrm{D}} u_\phi \\ &= \overline{\mathrm{D}}\, u^t \left[g_{\phi\phi}\, \tfrac{V^\phi}{c} + g_{t\phi}\right] = \overline{\mathrm{D}}\, u^t\, g_{\phi\phi}\, \left[\tfrac{V^\phi}{c} + \tfrac{g_{t\phi}}{g_{\phi\phi}}\right] = \overline{\mathrm{D}}\, u^t\, g_{\phi\phi}\, \left[\tfrac{V^\phi}{c} - \tfrac{\omega_{FDE}}{c}\right] \\ &= \overline{\mathrm{D}}\, \tfrac{u^t}{c}\, g_{\phi\phi}\, [\omega - \omega_{FDE}] = \overline{\mathrm{D}}\, \Gamma\, g_{\phi\phi}\, [\omega - \omega_{FDE}] = \overline{\overline{\mathrm{D}}}\, g_{\phi\phi}\, [\omega - \omega_{FDE}] \end{aligned} \tag{2.19}$$

with $\overline{\mathrm{D}} = D\frac{h}{c^2} = \rho\Gamma\frac{h}{c^2}$ and angular velocity $\omega = V^\phi = \frac{u^\phi}{u^t}\, c = \frac{u^\phi}{\Gamma} = \frac{\frac{d\phi}{d\tau}}{\Gamma} = \frac{d\phi}{dt}$.

Singularities

The Kerr solution in Boyer-Lindquist coordinates shows a coordinate singularity at the event horizon. This can be transformed away, for example, by changing to the so-called Kerr-Schild coordinates . But unfortunately the resulting metric has more off-diagonal elements which results in much more complicated formulas.
For a Schwarzschild black hole, a non-rotating black hole, one has a point singularity at the origin, where a point mass is located and the Riemann tensor diverges, e.g. the spacetime has infinite curvature.
Whereas in the case of a rotating (Kerr) black hole one has a ring singularity which is located in the equatorial plane and has a radius of $r = ar_g$, which in Boyer-Lindquist coordinates is given by $\overline{\rho}(r,\theta) = \sqrt{r^2 + (ar_g)^2 \sin^2\theta} = 0$.

Innermost Stable Circular Orbit (ISCO)

The innermost stable circular orbit (ISCO), also called marginally stable orbit, is the innermost circular orbit of a test particle in the equatorial plane, where a stable rotation around a black hole is still possible. This radius follows from the discussion of effective potentials in the Kerr spacetime and the limiting cases are: for a extreme Kerr hole with retrograde rotation $r_{ISCO}(a = -1) = 9r_g$, with prograde rotation $r_{ISCO}(a = +1) = r_g$ and for a Schwarzschild black hole $r_{ISCO}(a = 0) = 6r_g$. Following Müller (2004, p. 14) or Camenzind (2007, Chap. 8), the ISCO radii in dependence of the Kerr rotation parameter a can be calculated by

$$\begin{aligned} r_{\mathrm{ISCO}} &= r_g \left(3 + Z_2 \mp \sqrt{(3 - Z_1)\,(3 + Z_1 + 2Z_2)}\right), \\ Z_1 &= 1 + \left(1 - a^2\right)^{1/3} \left((1+a)^{1/3} + (1-a)^{1/3}\right), \\ Z_2 &= \sqrt{3a^2 + Z_1^2}, \end{aligned} \tag{2.20}$$

The upper sign holds for prograde whereas the lower sign is for retrograde orbits.
Another characteristic radius in black hole physics is the marginally bound orbit r_{mb}, which is the smallest possible radius of a circular orbit at which an outward perturbation lets escape a test particle to infinity or in other words, where a test particle starts to be gravitationally bound by the black hole. This radius is given by:

$$r_{mb} = r_g \left(2 - a + 2\sqrt{1-a}\right) = r_g \left(1 + \sqrt{1-a}\right)^2. \tag{2.21}$$

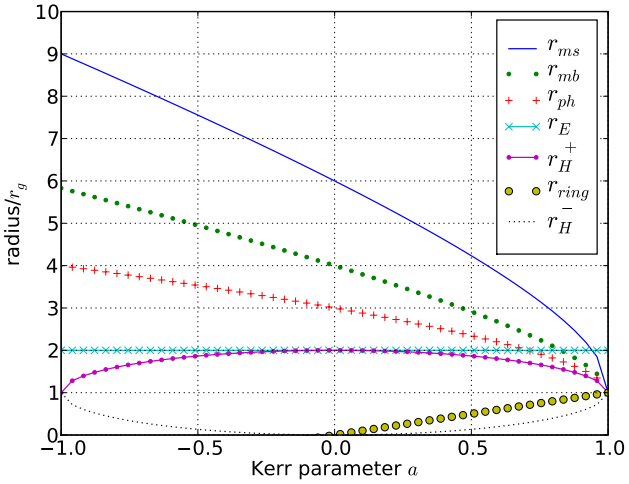


Figure 2.3: *Dependence of the characteristic radii on the Kerr parameter: r_{ms} is the marginally stable orbit, also called innermost stable circular orbit (ISCO), r_{mb} is the marginally bound orbit, r_{ph} is the radius of the photon sphere, r_E the radius of the ergosphere, r_H^+ the radius of the event horizon, r_{ring} the radius of the ring singularity and r_H^- the radius of the Cauchy horizon. For $a = 0$ the Schwarzschild black hole is reproduced.*

A further radius is the radius of the photon sphere, inside which photons cannot have stable trajectories anymore.

$$r_{ph} = 2\, r_g \left[1 + \cos\left(\frac{2}{3}\,\arccos(-a)\right)\right] \tag{2.22}$$

The dependence of the characteristic radii on a can be seen in Fig. (2.3).

2.3 The hydrodynamical equations in Kerr spacetime: General Relativistic Euler and Navier-Stokes equations

Hydrodynamics is a macroscopic description of a fluid and assumes that the mean free path of the particles is much smaller than the length scales of relevant physical processes and that the particles in a small volume have a specific energy distribution, which can be described by the thermodynamical concept of temperature.
The hydrodynamic equations describe the conservation of mass, momentum and energy. This system of equations is closed by an equation of state (EoS).
The Euler equations are the fundamental equations of hydrodynamics for an ideal gas whereas the fluid motions of a viscous gas are described by the Navier-Stokes equations.

The general relativistic hydrodynamical equations are derived (following the internal energy formulation of Wilson (1972) and Hawley et al. (1984a,b) from the four-velocity normalisation $u^\mu u_\mu = -c^2$, the conservation of baryonic number $\nabla_\mu(\rho u^\mu) = 0$, the parallel component of the stress-energy conservation equation $u_\nu \nabla_\mu T^{\mu\nu} = 0$ (to derive the internal energy equation) and from the transverse components $(g_{\xi\nu} + u_\xi u_\nu)\nabla_\mu T^{\mu\nu} = 0$ (to derive the momentum equations).

2.3.1 Velocities and Momenta

The **four-velocity** is defined as

$$u^\mu \doteq \frac{dx^\mu}{d\tau} \tag{2.23}$$

and satisfies the **normalisation**

$$u^\mu u_\mu = -c^2. \tag{2.24}$$

The units/dimensions of the four-velocity components are: $u^t : LT^{-1}$, $u^r : LT^{-1}$, $u^\theta : T^{-1}$ and $u^\phi : T^{-1}$.

Defining the **general relativistic pseudo Lorentz factor**, the non-dimensional time-component of the 4-velocity u^t:

$$\Gamma = \frac{dt}{d\tau} \tag{2.25}$$

one gets for u^t:

$$u^t = \frac{dct}{d\tau} = c\frac{dt}{d\tau} = c\Gamma. \tag{2.26}$$

The **transport velocity** is defined as

$$V^\mu \doteq \frac{u^\mu}{u^t}c = \frac{dx^\mu}{dt}, \tag{2.27}$$

from which follows for the four-velocity: $u^\mu = u^t(V^\mu/c) = \Gamma\, V^\mu$.
The units are the same as for the four-velocities.

The **physical velocities** in the Boyer-Lindquist coordinate system for an observer at infinity are defined as the absolute values of the transport velocities and have all units of $[LT^{-1}]$:
The radial velocity U is:

$$U = \sqrt{U^2} = \sqrt{V_r V^r} = \sqrt{g_{rr} V^r V^r} = \sqrt{g_{rr}}\, V^r \tag{2.28}$$

the azimuthal/latitudinal velocity V:

$$V = \sqrt{V^2} = \sqrt{V_\theta V^\theta} = \sqrt{g_{\theta\theta} V^\theta V^\theta} = \sqrt{g_{\theta\theta}}\, V^\theta \tag{2.29}$$

and the toroidal velocity $\bar{v}_\phi$:

$$\bar{v}_\phi = \sqrt{\bar{v}_\phi^2} = \sqrt{V_\phi V^\phi} = \sqrt{g_{\phi\alpha} V^\alpha V^\phi} = \sqrt{g_{\phi t} V^t V^\phi + g_{\phi\phi} V^\phi V^\phi} = \sqrt{g_{\phi t} c V^\phi + g_{\phi\phi} V^\phi V^\phi}, \tag{2.30}$$

which gives in case of a non-rotating (Schwarzschild) black hole:

$$\bar{v}_{\phi;\,\text{Schwarzschild}} = \sqrt{g_{\phi\phi} V^\phi V^\phi} = \sqrt{g_{\phi\phi}} V^\phi. \tag{2.31}$$

Instead of the toroidal velocity $\bar{v}_\phi$ the angular velocity ω is used:

$$\omega := V^\phi \tag{2.32}$$

The frame dragging angular velocity is:

$$\omega_{FDE} = -\frac{g_{t\phi}}{g_{\phi\phi}} c \tag{2.33}$$

The absolute value of the physical velocity vector (in 3-space) (for an observer at infinity in the Boyer-Lindquist coordinates) is

$$\bar{v} = |\vec{v}| = \sqrt{\vec{v}\vec{v}} = \sqrt{U^2 + V^2 + \vec{v}_\phi^2} \tag{2.34}$$

In the 3+1 split of spacetime, also called the ADM formalism (Arnowitt, Deser, and Misner 1962) (see also: Gourgoulhon 2007), the 4 dimensional spacetime is foliated into non-intersecting space-like hypersurfaces of constant coordinate time t = const . Each of such 3 dimensional hypersurfaces have a space like metric h_{ij} and can therefore seen as 'absolute space' at the specified time t. The advance of such hypersurfaces in time is parameterized by the redshift or lapse function $\alpha = -\frac{1}{\sqrt{g^{tt}}}$, which describes the distance to the next parallel hypersurface along a timelike unit vector n^μ normal to the surface. The space-like tangential shift vector β^i describes the motion of coordinates within a hypersurface. One can define a local fiducial observer, the FIDO, who is at rest in absolute space and has a 4-velocity u_μ with $u_t = u_0 = -\alpha c$ and $u_i = 0$. So the shift vector β^i describes the shift in time of the grid of the hypersurface relative to the local FIDO.

The three-velocity components of the plasma for a **local observer in the Euler frame** (sitting on the local grid point) are:

$${v_E}^i = -\frac{\hat{h}^i_\mu u^\mu}{n_\mu u^\mu} c, \qquad i = 1,2,3 \tag{2.35}$$

where $\hat{h}_{\mu\nu} = g_{\mu\nu} + n_\mu n_\nu$ is the projection operator onto the hypersurface orthogonal to n^μ.
These velocity components can also be written as:

$${v_E}^i = \left(\frac{u^i}{\alpha u^t} + \frac{\beta^i}{\alpha}\right) c. \tag{2.36}$$

Using this velocity one can define and calculate the local **relativistic Lorentz factor** W in the Eulerian frame:

$$W = -n_\mu \frac{u^t}{c} = \alpha \frac{u^t}{c} = \alpha\Gamma = \frac{1}{\sqrt{1 - \frac{{v_E}^2}{c^2}}}, \tag{2.37}$$

where ${v_E}^2 = v_{Ei}{v_E}^i = g_{ij}{v_E}^i{v_E}^j$ and the normalisation condition $u_\mu u^\mu = -c^2$ was used to transform the first term into the second, which shows that in the limit of a locally flat spacetime (with a Minkowski metric) the general relativistic equations give the usual special relativistic equations. In the Euler frame one gets for the 4-velocity:

$$u^\mu = W \begin{pmatrix} 1 \\ {v_E}^i \end{pmatrix}. \tag{2.38}$$

The relations between the local Eulerian velocities and the Boyer Lindquist velocities (used in the simulation code) are:

$$\begin{aligned} {v_E}^r &= \frac{U}{\alpha\sqrt{g_{rr}}} & U_E &= |{v_E}^r| = \frac{U}{\alpha} \\ {v_E}^\theta &= \frac{V}{\alpha\sqrt{g_{\theta\theta}}} & V_E &= |{v_E}^\theta| = \frac{V}{\alpha} \\ {v_E}^\phi &= \frac{d\omega}{\alpha} & V_E^\phi &= |{v_E}^\phi| = \sqrt{g_{\phi\phi}}\,\frac{d\omega}{\alpha} \end{aligned} \tag{2.39}$$

The velocity components ${v_E}^i$ of a local observer in the Euler frame are physically more intuitive than the Boyer-Lindquist coordinate velocities.

Using the transformation $\mathrm{q}_\mu = \mathrm{g}_{\mu\nu}\mathrm{q}^\nu$ one can determine the covariant vector q_μ from the contravariant vector q^ν using the metric, whereas $\mathrm{q}^\mu = \mathrm{g}^{\mu\nu}\mathrm{q}_\nu$ is used to do the transformation in the other direction.

This is used for example to transform the contravariant components of the 4-velocity for a Kerr black hole in Boyer-Lindquist coordinates to the covariant components:

$$\begin{cases} \mathrm{u_t} = g_{tt}\,u^t + g_{t\phi}\,u^\phi = u^t\,[g_{tt} + g_{t\phi}\,(V^\phi/c)] & [LT^{-1}] \\ u_r = g_{rr}\,u^r = g_{rr}\,u^t\,(V^r/c) = \sqrt{g_{rr}}\,\frac{u^t}{c}\,U = \sqrt{g_{rr}}\,\Gamma\,U & [LT^{-1}] \\ u_\theta = g_{\theta\theta}\,u^\theta = g_{\theta\theta}\,u^t\,(V^\theta/c) = \sqrt{g_{\theta\theta}}\,\frac{u^t}{c}\,V = \sqrt{g_{\theta\theta}}\,\Gamma\,V & [L^2T^{-1}] \\ u_\phi = g_{t\phi}\,u^t + g_{\phi\phi}\,u^\phi = u^t\,[g_{t\phi} + g_{\phi\phi}\,(V^\phi/c)] = g_{\phi\phi}\,\Gamma\,[\omega - \omega_{FDE}] & [L^2T^{-1}]. \end{cases} \tag{2.40}$$

The relativistic mass density is:

$$D \doteq \rho\frac{u^t}{c} = \rho\Gamma \quad [ML^{-3}]. \tag{2.41}$$

The specific relativistic enthalpy (enthalpy per mass density) is:

$$h = c^2 + \epsilon + \frac{p}{\rho} \quad [L^2T^{-2}], \tag{2.42}$$

where c^2 comes from the rest-mass energy density and ϵ is the specific internal energy (internal energy per mass density).
With this one can define the variables

$$\overline{\mathrm{D}} \doteq D\frac{h}{c^2} \quad [ML^{-3}] \tag{2.43}$$

and

$$\overline{\overline{\mathrm{D}}} \doteq \overline{\mathrm{D}}\frac{u^t}{c} = \overline{\mathrm{D}}\Gamma = D\frac{h}{c^2}\Gamma = \rho\frac{h}{c^2}\Gamma^2 = \rho\tilde{h}\Gamma^2 \quad [ML^{-3}]. \tag{2.44}$$

where

$$\tilde{h} := \frac{h}{c^2} \quad [1] \tag{2.45}$$

From the four-velocities and the relativistic density one can define the corresponding relativistic covariant 4-momenta which are defined by $M_\mu \doteq \overline{\mathrm{D}} u_\mu$:

$$\left\{\begin{array}{lcll} M_t & = & \overline{\mathrm{D}} u_t = \overline{\mathrm{D}}\, u^t\, [g_{tt} + g_{t\phi}\, (V^\phi/c)] & [ML^{-3}LT^{-1}] \\ M_r & = & \overline{\mathrm{D}} u_r = \overline{\mathrm{D}}\, g_{rr}\, u^t\, (V^r/c) = \overline{\mathrm{D}}\, \sqrt{g_{rr}}\, \frac{u^t}{c}\, U = \sqrt{g_{rr}}\, \overline{\overline{\mathrm{D}}}\, U & [ML^{-3}LT^{-1}] \\ M_\theta & = & \overline{\mathrm{D}} u_\theta = \overline{\mathrm{D}}\, g_{\theta\theta}\, u^t\, (V^\theta/c) = \overline{\mathrm{D}}\, \sqrt{g_{\theta\theta}}\, \frac{u^t}{c}\, V = \sqrt{g_{\theta\theta}}\, \overline{\overline{\mathrm{D}}}\, V & [ML^{-3}L^2T^{-1}] \\ M_\phi & = & \overline{\mathrm{D}} u_\phi = \overline{\mathrm{D}}\, u^t\, [g_{t\phi} + g_{\phi\phi}\, (V^\phi/c)] & \\ & = & \overline{\mathrm{D}}\, \frac{u^t}{c}\, g_{\phi\phi}\, [\omega - \omega_{FDE}] = \overline{\overline{\mathrm{D}}}\, g_{\phi\phi}\, [\omega - \omega_{FDE}] & [ML^{-3}L^2T^{-1}]. \end{array}\right. \tag{2.46}$$

from which the contravariant 4-momenta may be obtained (using the orthogonalisation and normalisation relation $g^{\mu\nu} g_{\nu\alpha} = \delta^\mu_\alpha$ and the symmetry relation of the metric: $g_{\mu\nu} = g_{\nu\mu}$):

$$\left\{\begin{array}{lcll} M^t & = & g^{tt}\, M_t + g^{t\phi}\, M_\phi & = \overline{\mathrm{D}}\, u^t\, (g^{tt}\, [g_{tt} + g_{t\phi}\, (V^\phi/c)] + g^{t\phi}\, [g_{t\phi} + g_{\phi\phi}\, (V^\phi/c)]) \\ & & & = \overline{\mathrm{D}}\, u^t\, (g^{tt}\, g_{tt} + g^{t\phi}\, g_{t\phi} + [g^{tt}\, g_{t\phi} + g^{t\phi}\, g_{\phi\phi}]\, (V^\phi/c)) \\ & & & = \overline{\mathrm{D}}\, u^t = \overline{\overline{\mathrm{D}}}\, c \\ & & & \quad [ML^{-3}LT^{-1}] \\ M^r & = & g^{rr}\, M_r & = \overline{\mathrm{D}}\, g^{rr} g_{rr}\, u^t\, (V^r/c) = \overline{\mathrm{D}}\, \frac{u^t}{c}\, V^r = \overline{\mathrm{D}}\, u^r = \overline{\overline{\mathrm{D}}}\, V^r = \overline{\overline{\mathrm{D}}}\, \frac{U}{\sqrt{g_{rr}}} \\ & & & \quad [ML^{-3}LT^{-1}] \\ M^\theta & = & g^{\theta\theta}\, M_\theta & = \overline{\mathrm{D}}\, g^{\theta\theta} g_{\theta\theta}\, u^t\, (V^\theta/c) = \overline{\mathrm{D}}\, \frac{u^t}{c}\, V^\theta = \overline{\mathrm{D}}\, u^\theta = \overline{\overline{\mathrm{D}}}\, V^\theta = \overline{\overline{\mathrm{D}}}\, \frac{V}{\sqrt{g_{\theta\theta}}} \\ & & & \quad [ML^{-3}T^{-1}] \\ M^\phi & = & g^{\phi\phi}\, M_\phi + g^{\phi t}\, M_t & = \overline{\mathrm{D}}\, u^t\, (g^{\phi\phi}\, [g_{t\phi} + g_{\phi\phi}\, (V^\phi/c)] + g^{\phi t}\, [g_{tt} + g_{t\phi}\, (V^\phi/c)]) \\ & & & = \overline{\mathrm{D}}\, u^t\, (g^{\phi\phi}\, g_{t\phi} + g^{\phi t}\, g_{tt} + [g^{\phi\phi}\, g_{\phi\phi} + g^{\phi t}\, g_{t\phi}]\, (V^\phi/c)) \\ & & & = \overline{\mathrm{D}}\, u^t\, (V^\phi/c) = \overline{\mathrm{D}}\, u^\phi = \overline{\overline{\mathrm{D}}}\, V^\phi \\ & & & \quad [ML^{-3}T^{-1}]. \end{array}\right. \tag{2.47}$$

The corresponding physical momenta (used in the simulation code) are defined as follows:

$$\begin{aligned} \bar{m} &= M_r = \sqrt{g_{rr}}\, \overline{\overline{\mathrm{D}}}\, U \\ \bar{n} &= M_\theta = \sqrt{g_{\theta\theta}}\, \overline{\overline{\mathrm{D}}}\, V \\ \bar{l} &= M_\phi = \overline{\mathrm{D}}\, u^t\, [g_{t\phi} + g_{\phi\phi}\, (V^\phi/c)] = \overline{\mathrm{D}}\, \Gamma\, g_{\phi\phi}\, [\omega - \omega_{FDE}] = \overline{\overline{\mathrm{D}}}\, g_{\phi\phi}\, [\omega - \omega_{FDE}] \end{aligned} \tag{2.48}$$

The normalisation of the 4-velocity $u^\mu u_\mu = -c^2$ yields for the Kerr metric in Boyer-Lindquist coordinates:

$$\begin{aligned} &(u^t)^2\, \left[g_{tt} + 2\, g_{t\phi}\, (V^\phi/c) + g_{rr}\, (V^r/c)^2 + g_{\theta\theta}\, (V^\theta/c)^2 + g_{\phi\phi}\, (V^\phi/c)^2\right] = -c^2 \\ &(u^t)^2\, \left[g_{tt} + 2\, g_{t\phi}\, (V^\phi/c) + (U/c)^2 + (V/c)^2 + g_{\phi\phi}\, (V^\phi/c)^2\right] = -c^2 \end{aligned} \tag{2.49}$$

For the 4-momenta the normalisation is then:

$$\begin{aligned} &M^\mu M_\mu = \overline{\mathrm{D}} u^\mu\, \overline{\mathrm{D}} u_\mu = \overline{\mathrm{D}}^2\, u^\mu u_\mu = -c^2\, \overline{\mathrm{D}}^2 \\ &M^\mu M_\mu = g^{\mu\nu} M_\nu M_\mu = g^{tt} M_t^2 + 2\, g^{t\phi} M_t M_\phi + g^{rr} M_r^2 + g^{\theta\theta} M_\theta^2 + g^{\phi\phi} M_\phi^2 = -c^2\, \overline{\mathrm{D}}^2 \end{aligned} \tag{2.50}$$

From the normalisation one can determine the general relativistic pseudo Lorentz factor $\Gamma = \frac{u^t}{c}$.

2.3.2 The Euler Equations

The Euler equations describe the motion for a one-component ideal fluid (non-viscous, no heat conduction, ...).
They consist of conservation equations of particle number or mass, momentum and energy.
In the Newtonian case these equations are (without gravitational force):

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \left(\rho \vec{u}\right) = 0, \tag{2.51a}$$

$$\frac{\partial \left(\rho \vec{u}\right)}{\partial t} + \nabla \cdot \left(\vec{u} \otimes \left(\rho \vec{u}\right)\right) = -\nabla P, \tag{2.51b}$$

$$\frac{\partial E}{\partial t} + \nabla \cdot \left(\vec{u}\left(E + P\right)\right) = 0, \tag{2.51c}$$

where ρ is the density, $\vec{u}$ the velocity vector, E the total energy and P the pressure.

2.3.3 Conservation of Mass

The continuity equation, which describes the conservation of mass, can be derived from the conservation of the particle number density n:

$$\left(n u^{\mu}\right)_{;\mu} = 0, \tag{2.52}$$

where $u^{\mu} = \frac{dx^{\mu}}{d\tau} = \begin{pmatrix} u^t \\ \vec{u} \end{pmatrix}$ is the 4-velocity.

In *special relativity*, where the metric simplifies to the Minkowski metric: $g^{\mu\nu} = \eta^{\mu\nu}$ in the case of Cartesian coordinates, u^t simplifies to the special relativistic pseudo Lorentz factor Γ_{SR} times the speed of light: $\Gamma_{SR} = \frac{u^t}{c} = \left(1 - \frac{u_i u^i}{c^2}\right)^{-1/2}$.
In general relativity the contravariant time component of the 4-velocity u^t includes also other metric terms which can be derived taking into account the normalisation condition $u^{\mu}u_{\mu} = u^{\mu}g_{\mu\nu}u^{\nu} = g^{\mu\nu}u_{\nu}u_{\mu} = -c^2$ and the metric relation $g^{\mu\nu}g_{\nu\alpha} = \delta^{\mu}_{\alpha}$, e.g. for a gas at rest ($u^i = 0$), $u^t = \frac{c}{\sqrt{-g_{tt}}} = \alpha\, c$, where α is called the *lapse function*.

With the definition of the fluid density $\rho = n\, m$, where m is the rest-mass per particle which can be factored out in Eq. (2.52) since it is a constant, one gets the continuity equation:

$$\left(\rho u^{\mu}\right)_{;\mu} = 0. \tag{2.53}$$

Using the expression of the covariant 4-velocity divergence $u^{\mu}{}_{;\mu} = \frac{1}{\sqrt{-g}}\partial_{\mu}\left(\sqrt{-g}u^{\mu}\right)$ and splitting-up in temporal and spatial parts, the so-called 3+1 split of spacetime, the continuity equation can be written as follows:

$$\frac{1}{\sqrt{-g}}\partial_t\left(\sqrt{-g}\frac{u^t}{c}\rho\right) + \frac{1}{\sqrt{-g}}\partial_k\left(\sqrt{-g}\rho u^k\right) = 0. \tag{2.54}$$

Assuming that the gravitational field is dominated by the central compact object (e.g. a black hole or a neutron star) and the contribution of the atmosphere or accretion disc to the gravitational field and the accumulation of mass and the spin-up of the central object during the simulated time can be neglected, one can use a static background metric ($\partial_t g = 0$).

Introducing the **relativistic density** $D = \frac{u^t}{c}\,\rho = \Gamma\,\rho$ and the **transport velocity** $V^\mu = u^\mu/(\frac{u^t}{c}) = u^\mu/\Gamma$ yields to the **continuity equation in general relativity**

$$\boxed{\partial_t D + \frac{1}{\sqrt{-g}}\partial_k\left(\sqrt{-g}DV^k\right) = 0.} \qquad [ML^{-3}\,T^{-1}] \tag{2.55}$$

2.3.4 Conservation of the Stress-energy Tensor

For the derivation of the other hydrodynamical equations the stress-energy tensor is used, since it includes the energy density, the momentum density, the energy flux and momentum flux. This tensor makes up the right-hand side of Einstein's field equation (2.3) and is a source of curvature of spacetime, which is neglected here, since a fixed background metric, the Kerr metric, is assumed.
The stress-energy tensor of an ideal gas has the following form:

$$T^{\mu\nu} = \rho\frac{h}{c^2}u^\mu u^\nu + Pg^{\mu\nu} \qquad \text{or} \qquad T^{\mu\nu} = \frac{(\hat{\rho}+P)}{c^2}u^\mu u^\nu + Pg^{\mu\nu} \qquad [ML^{-3}\,L^2T^{-2}], \tag{2.56}$$

where P is the pressure, $h = c^2 + \epsilon + \frac{P}{\rho}$ is the relativistic enthalpy, ϵ is the specific internal energy, $g^{\mu\nu}$ is the metric tensor and u^μ is the 4-velocity. ρ is the mass density, while $\hat{\rho}$ is the energy density. The entropy is $H = \rho h = \rho c^2 + \rho\epsilon + P = \hat{\rho} + P$.

To derive the general relativistic momentum and energy conservation equations the divergence of the stress-energy tensor

$$T^{\mu\nu}{}_{;\mu} = \left[u^\nu\partial_\mu\frac{(\hat{\rho}+P)}{c^2} + \frac{(\hat{\rho}+P)}{c^2}u^\nu{}_{;\mu}\right]u^\mu + \frac{(\hat{\rho}+P)}{c^2}u^\nu u^\mu{}_{;\mu} + \partial^\nu P, \tag{2.57}$$

(which has for example for $\nu = r$ the dimensions: $[ML^{-3}L^2T^{-2}L^{-1}]$) has to be *conserved*,

$$\boxed{T^{\mu\nu}{}_{;\mu} = 0.} \tag{2.58}$$

Einstein assumed the simplest spacetime connection, which means that he assumed spacetime to be torsion free, $\Gamma^\alpha{}_{\beta\gamma} = \Gamma^\alpha{}_{\gamma\beta}$, and that the inner product (norm) is preserved, which results in the metric compatibility $g^{\mu\nu}{}_{;\mu} = 0$ (Camenzind 2007). This property is used in the calculation of the divergence of the stress-energy tensor of an ideal gas (3.82).
The conservation of the stress-energy tensor can be broken up into parts: For the conservation of energy one looks at the parallel components of the stress-energy tensor to u^ν, while for the momentum conservation at its components transversal to u^ν.

2.3.5 Conservation of Energy

Internal Energy Equation

By multiplying the conservation equation of the stress-energy tensor with the 4-velocity u_ν one gets:

$$\begin{aligned} u_\nu T^{\mu\nu}{}_{;\mu} = &\left[u_\nu u^\nu \partial_\mu \frac{(\hat{\rho}+P)}{c^2} + \frac{(\hat{\rho}+P)}{c^2} u_\nu u^\nu{}_{;\mu} \right] u^\mu \\ &+ (\hat{\rho}+P)\, u^\nu u_\nu u^\mu{}_{;\mu} + u_\nu \partial^\nu P = 0 \end{aligned} \tag{2.59}$$

Use of the normalisation of the 4-velocity $u_\mu u^\mu = -c^2$ gives

$$u_\nu T^{\mu\nu}{}_{;\mu} = -\partial_\mu \frac{(\hat{\rho}+P)}{c^2} u^\mu c^2 - \frac{(\hat{\rho}+P)}{c^2} c^2 u^\mu{}_{;\mu} + u_\nu \partial^\nu P = 0, \tag{2.60}$$

where we got rid of the term $\frac{(\hat{\rho}+P)}{c^2} u_\nu u^\nu{}_{;\mu}$, because of $u^\nu{}_{;\mu} u_\nu = \frac{1}{2} (u^\nu u_\nu)_{;\mu} = 0$.

Next the substitution $(\hat{\rho}+P) = h\rho$, where h is the enthalpy, and the expression $u_\nu \partial^\nu P = u^\mu \partial_\mu P$ are used to further simplify the equation:

$$u_\nu T^{\mu\nu}{}_{;\mu} = -\left(h\rho u^\mu\right)_{;\mu} + u^\mu \partial_\mu P = 0. \tag{2.61}$$

Using the chain rule $u^\mu \partial_\mu P = (P u^\mu)_{;\mu} - u^\mu{}_{;\mu} P$ and gather terms, one gets

$$\left[(h\rho - P)\, u^\mu\right]_{;\mu} + u^\mu{}_{;\mu} P = 0. \tag{2.62}$$

Adding the continuity equation $(\rho u^\mu)_{;\mu} = 0$ times the square of the speed of light gives

$$\left[\left(h\rho - P - \rho c^2\right) u^\mu\right]_{;\mu} + u^\mu{}_{;\mu} P = 0, \tag{2.63}$$

and using the definition of the internal energy $\rho\epsilon = h\rho - \rho c^2 - P$ yields

$$\left(\rho\epsilon u^\mu\right)_{;\mu} + u^\mu{}_{;\mu} P = 0. \tag{2.64}$$

Finally applying the 3+1 split of spacetime and introducing the **relativistic internal energy density** $\epsilon^{\mathrm{d}} = \epsilon D = \rho \frac{u^t}{c} \epsilon = \rho \Gamma \epsilon$ and the transport velocity $V^\mu = u^\mu / u^t c = u^\mu / \Gamma$ gives the equation that describes the **time evolution of the relativistic internal energy density** in general relativity:

$$\boxed{\partial_t \epsilon^{\mathrm{d}} + \frac{1}{\sqrt{-g}} \partial_k \left(\sqrt{-g}\, \epsilon^{\mathrm{d}} V^k \right) = -P \left[\partial_t \left(\frac{u^t}{c} \right) + \frac{1}{\sqrt{-g}} \partial_k \left(\sqrt{-g}\, \frac{u^t}{c} V^k \right) \right].} \tag{2.65}$$

Assuming an ideal gas the pressure is defined as $P(\epsilon, \rho) = (\gamma - 1)\rho\epsilon = (\gamma - 1)\, \epsilon^{\mathrm{d}} / \Gamma$.

Note, that the equation of time evolution of the relativistic internal energy density includes the time derivative of the pseudo Lorentz factor $\Gamma = \frac{u^t}{c}$ on the right hand side, which is highly nonlinear, so special care has to be taken to accurately estimate $\partial_t \Gamma$ in numerical simulations.

If using the internal energy formulation in numerics an artificial viscosity is added at shock discontinuities, which acts like a scalar pressure to numerically ensure the conservation of total energy and give the right results at shock fronts.

Conservation of Total Energy

A disadvantage of the internal energy formulation is that one has to fine-tune the artificial viscosity parameter at shock fronts to ensure total energy conservation.
The total energy conservation does not have this problem since it is numerically formulated in such a way to ensure the conservation of total energy.
The *equation of total energy conservation* is following Font (2008) and using a static background metric:

$$\boxed{\partial_t \tau + \frac{1}{\sqrt{-g}} \partial_k \left(\sqrt{-g} \left(\tau + P \right) v^k \right) = \alpha \left(T^{\rho t} \partial_\rho \ln \alpha - T^{\rho\sigma} \Gamma^t{}_{\rho\sigma} \right),} \tag{2.66}$$

where $\tau = \overline{\overline{D}} c^2 - P - Dc^2 = \Gamma^2 \rho \tilde{h} c^2 - P - \Gamma \rho c^2$ is the real total energy $E_{tot} = \Gamma^2 \rho \tilde{h} c^2 - P$ subtracted by the relativistic rest mass energy $Dc^2 = \Gamma \rho c^2$

and α is the lapse function. The source terms on the RHS arise due to the spacetime curvature, but vanish in the case of a flat space.

Especially for high Lorentz factors but also in other circumstances the total energy formulation may break down and gives negative pressure values and one has to use the internal energy formulation.

2.3.6 Conservation of Momentum

Extracting the transversal components of the conservation of the stress-energy tensor equation (2.57) by using the projection tensor, which is defined as $h_{\mu\nu} = u_\mu u_\nu + g_{\mu\nu} c^2$, one gets the equation of momentum conservation.
Using the projection operator on u^ν, one can see that it only extracts only the transversal contributions:

$$h_{\mu\nu} u^\nu = \left(u_\mu u_\nu + g_{\mu\nu} c^2 \right) u^\nu = (-u_\mu + u_\mu) c^2 = 0 \tag{2.67}$$

Applying this operator $h_{\mu\nu}$ on $T^{\lambda\nu}{}_{;\lambda}$ one gets

$$h_{\mu\nu} T^{\lambda\nu}{}_{;\lambda} = \left(u_\mu u_\nu + g_{\mu\nu} c^2 \right) T^{\lambda\nu}{}_{;\lambda} = u_\mu u_\nu T^{\lambda\nu}{}_{;\lambda} + g_{\mu\nu} c^2 T^{\lambda\nu}{}_{;\lambda}. \tag{2.68}$$

The term $u_\mu u_\nu T^{\lambda\nu}{}_{;\lambda}$ can be identified as the energy conservation equation $u_\mu \left(u_\nu T^{\lambda\nu}{}_{;\lambda} \right) = 0$ and therefore vanishes. For the term $g_{\mu\nu} c^2 T^{\lambda\nu}{}_{;\lambda}$ one can use the metric compatibility property $g^{\mu\nu}{}_{;\mu} = 0$

and gets:

$$\begin{aligned} h_{\mu\nu} T^{\lambda\nu}{}_{;\lambda} &= u_\mu u_\nu T^{\lambda\nu}{}_{;\lambda} + g_{\mu\nu} c^2 T^{\lambda\nu}{}_{;\lambda} \\ &= \left(c^2 g_{\mu\nu} T^{\lambda\nu} \right)_{;\lambda} \\ &= c^2 T^\lambda_{\mu;\lambda}. \end{aligned} \tag{2.69}$$

Renaming dummy indices first μ to ν and then λ to μ and expanding the stress-energy tensor and setting the divergence to zero:

$$0 = T^\mu_{\nu;\mu} = \left(g_{\nu\alpha} T^{\mu\alpha} \right)_{;\mu} = \left(\rho \frac{h}{c^2} g_{\nu\alpha} u^\alpha u^\mu + P g_{\nu\alpha} g^{\mu\alpha} \right)_{;\mu} = \left(\rho \frac{h}{c^2} u_\nu u^\mu + P g^\mu_\nu \right)_{;\mu}. \tag{2.70}$$

Writing out the divergence of the above equation, while keeping in mind that $u_\nu u^\mu$ is a tensor, results in

$$0 = \frac{1}{\sqrt{-g}}\partial_\mu\left(\sqrt{-g}\rho\frac{h}{c^2}u_\nu u^\mu\right) - \rho\frac{h}{c^2}\Gamma^\lambda{}_{\mu\nu}u_\lambda u^\mu + \partial_\nu P. \tag{2.71}$$

With the introduction of the **relativistic momentum density** $M_\mu = \rho\frac{h}{c^2}\frac{u^t}{c}u_\mu = \rho\frac{h}{c^2}\Gamma u_\mu = \overline{\mathrm{D}}u_\mu$ and the transport velocity $V^\mu = \frac{u^\mu}{\Gamma}$ this equation gives

$$\begin{aligned}\frac{1}{\sqrt{-g}}\partial_\mu\left(\sqrt{-g}M_\nu V^\mu\right) &= \Gamma^\lambda{}_{\mu\nu}M_\lambda V^\mu - \partial_\nu P \\ &= \Gamma^\lambda{}_{\mu\nu}\frac{M_\lambda M^\mu\, c}{M^t} - \partial_\nu P.\end{aligned} \tag{2.72}$$

For the next step, the connection coefficients multiplied by the symmetric tensor $M_\lambda M^\mu$ can be calculated as follows:

$$\begin{aligned}\Gamma^\lambda{}_{\mu\nu}M_\lambda M^\mu &= \frac{1}{2}g^{\lambda\sigma}\left(g_{\sigma\mu,\nu} + g_{\sigma\nu,\mu} - g_{\mu\nu,\sigma}\right)M^\mu M_\lambda \\ &= \frac{1}{2}M^\mu M^\sigma\left(g_{\sigma\mu,\nu} + g_{\sigma\nu,\mu} - g_{\mu\nu,\sigma}\right) \\ &= \frac{1}{2}M^\mu M^\sigma g_{\sigma\mu,\nu} + \frac{1}{2}\left(M^\mu M^\sigma g_{\sigma\nu,\mu} - M^\mu M^\sigma g_{\mu\nu,\sigma}\right) \\ &= \frac{1}{2}M^\mu M^\sigma g_{\sigma\mu,\nu} \\ &= \frac{1}{2}M^\mu M^\lambda g_{\lambda\mu,\nu}\,,\end{aligned} \tag{2.73}$$

where the symmetry property of $M^\mu M^\sigma$ was used in the summation of $M^\mu M^\sigma g_{\sigma\nu,\mu} = M^\mu M^\sigma g_{\mu\nu,\sigma}$. With this formula equation (2.72) can be simplified to

$$\frac{1}{\sqrt{-g}}\partial_\mu\left(\sqrt{-g}M_\nu V^\mu\right) = +\frac{M^\mu M^\lambda\, c}{2M^t}g_{\lambda\mu,\nu} - \partial_\nu P. \tag{2.74}$$

With the help of the normalisation of the metric $g^{\alpha\beta}g_{\beta\gamma} = \delta^\alpha_\gamma$ and the symmetry properties of the metric $g^{\alpha\beta} = g^{\beta\alpha}$ and $g_{\alpha\beta} = g_{\beta\alpha}$ one gets:

$$\partial_\nu\left(g^{\alpha\mu}g_{\mu\gamma}\right) = \partial_\nu\left(g^{\alpha\mu}\right)g_{\mu\gamma} + g^{\alpha\mu}\partial_\nu\left(g_{\mu\gamma}\right) \tag{2.75}$$

Since $\partial_\nu\left(g^{\alpha\mu}g_{\mu\gamma}\right) = \partial_\nu\left(\delta^\alpha_\gamma\right) = 0$ one gets:

$$g^{\alpha\mu}\partial_\nu\left(g_{\mu\gamma}\right) = -\partial_\nu\left(g^{\alpha\mu}\right)g_{\mu\gamma} \tag{2.76}$$

This can be used to get:

$$+\frac{1}{2}M^\gamma M^\mu\partial_\nu(g_{\mu\gamma}) = +\frac{1}{2}M^\gamma M_\alpha g^{\alpha\mu}\partial_\nu(g_{\mu\gamma}) = -\frac{1}{2}M^\gamma M_\alpha g_{\mu\gamma}\partial_\nu(g^{\alpha\mu}) = -\frac{1}{2}M_\mu M_\alpha\partial_\nu(g^{\alpha\mu}) \tag{2.77}$$

Performing the 3+1 spacetime split and taking into account that the time component of M_t is determined by the 4-velocity normalisation $u_\mu u^\mu = -c^2$ gives the **general relativistic equation of momentum conservation** of an ideal gas:

$$\boxed{\partial_t M_a + \frac{1}{\sqrt{-g}}\partial_k\left(\sqrt{-g}M_a V^k\right) = -\frac{M_\lambda M_\mu\, c}{2M^t}g^{\lambda\mu}{}_{,a} - \partial_a P.} \tag{2.78}$$

2.3.7 Summary of the Euler Equations

The equation of momentum conservation and the equation of energy conservation can be derived from the conservation of the stress-energy tensor. Together with the continuity equation they form the **general relativistic Euler equations**:

$$\partial_t D + \frac{1}{\sqrt{-g}}\partial_k\left(\sqrt{-g}DV^k\right) = 0, \tag{2.79a}$$

$$\partial_t M_a + \frac{1}{\sqrt{-g}}\partial_k\left(\sqrt{-g}M_a V^k\right) = -\frac{M_\lambda M_\mu\, c}{2M^t} g^{\lambda\mu}{}_{,a} - \partial_a P, \tag{2.79b}$$

$$\partial_t \epsilon^{\mathrm{d}} + \frac{1}{\sqrt{-g}}\partial_k\left(\sqrt{-g}\,\epsilon^{\mathrm{d}} V^k\right) = -P\left[\partial_t\left(\frac{u^t}{c}\right) + \frac{1}{\sqrt{-g}}\,\partial_k\left(\sqrt{-g}\,\frac{u^t}{c}V^k\right)\right], \tag{2.79c}$$

where the conservative variables are the relativistic density $D = \frac{u^t}{c}\rho = \Gamma\rho$, the relativistic momenta $M_\mu = \rho\frac{h}{c^2}\frac{u^t}{c}u_\mu = \overline{\mathrm{D}}u_\mu$, and the internal energy density $\epsilon^{\mathrm{d}} = \epsilon\, D$.

These equations together with an equation of state (EoS), which relates the pressure P with the density ρ and internal energy ϵ, form the system of hydrodynamic equations of an ideal fluid in general relativity.
Note that by using this formulation of the hydrodynamics-equations, the so-called flux-conservative form, in combination with finite volume discretization, mass and momenta are conserved up to small discretization errors.

2.3.8 Newtonian case and Newtonian limit

From these derived general relativistic hydrodynamics equations one can easily recover the pure Newtonian equations of a flat spacetime for very large radii (there the spherical coordinates also go asymptotically into Cartesian ones) by setting the pseudo Lorentz-factor $\Gamma = 1$ (slow motions relative to the speed of light c), the enthalpy $h = 1$ (thermodynamical non-relativistic energy densities), the metric component $g_{t\phi} = 0$ (no frame-dragging effect $a = 0$) and the other metric components $g_{ii} = 1 \quad i = 1, 2, 3$. Set $\sqrt{\Upsilon} = \sqrt{-g}/\alpha$ to a flat space determinant of the 3-metric, but leave $g_{tt} = -\alpha^2 = -(1 - 2\frac{r_g}{r})$ to account for the usual Newtonian gravitational potential in the momentum equations.

If one replaces the metric tensor g in Cartesian coordinates by the Minkowski metric η, general relativity is reduced to special relativity, which has no gravity term.

To recover the pure Newtonian equations with Newtonian gravity in spherical coordinates, i.e. also for small radii, one has to set $\Gamma = 1$, $h = 1$, $a = 0$ and the metric components to following values: $g_{tt} = -\alpha^2 = -(1 - 2\frac{r_g}{r}) = -(1 + 2\Phi)$ with gravitational potential $\Phi = -\frac{r_g}{r}$, $g_{rr} = 1$ (and not as in the Schwarzschild case to: $g_{rr} = \frac{1}{\alpha^2}$), $g_{\theta\theta} = r^2$ and $g_{\phi\phi} = r^2\cos^2\theta$.

2.4 Navier-Stokes Equations in General Relativity

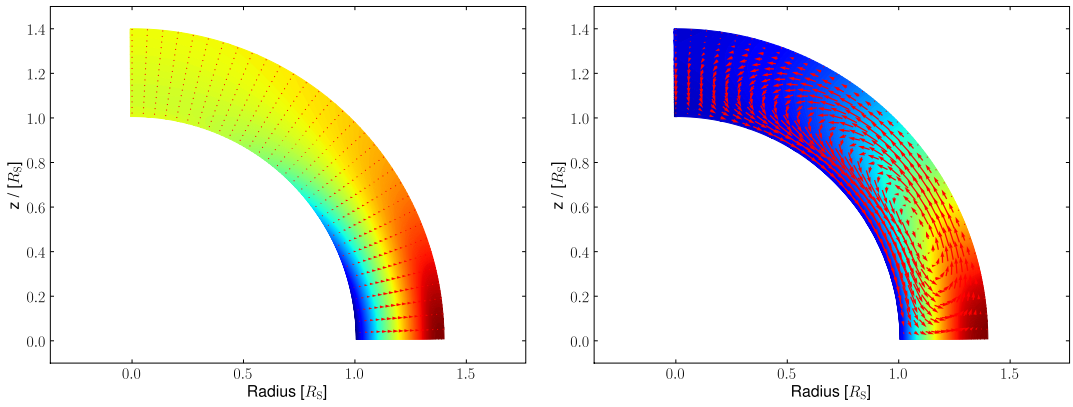
The Navier-Stokes equations additionally take diffusive effects due to viscosity into account, which can be of crucial importance in some cases. The effect of viscosity was often neglected in Astrophysical simulations up to now, because the effects of molecular viscosity are in general very low compared to other physical processes.
Especially the impact of shear-viscosity between fluid layers with large differences in velocities on turbulence, momentum transport or angular momentum transport can be huge and result in different results.

For example in the Taylor-Couette flow simulations in Fig. (2.4) depending on the value of the viscosity laminar or turbulent flow patterns can be obtained. If the so-called *Reynolds number*, which describes the ratio between the inertial forces and the viscous forces, is large enough, there can occur turbulence.

The momentum equations of the Newtonian Navier-Stokes equations are:

$$\rho \left[\frac{\partial \vec{v}}{\partial t} + \left(\vec{v} \cdot \nabla \right) \vec{v} \right] = -\nabla P - \nabla \times \left[\eta \left(\nabla \times \vec{v} \right) \right] + \tfrac{4}{3} \nabla \left(\eta \, \nabla \vec{v} \right) + \vec{F}, \tag{2.80}$$

where η is the dynamic viscosity, $\vec{v}$ the velocity of the fluid and $\vec{F}$ additional external forces, like gravity or electrical forces.



(a) *Re*(ν) < *Re*(ν_{crit})*: Laminar flow* **(b)** *Re*(ν) > *Re*(ν_{crit})*: Taylor vortex flow*

Figure 2.4: *Simulation of the Taylor-Couette flow between two rotating spheres. Plotted is the density (colour) and the velocity field (vectors). For the same time and setup we get very different flow configurations, depending on the Reynolds number (viscosity ν). For Re(ν) ≫ Re(ν_{crit}) the flow is turbulent. (Hilscher 2009)*

The Navier-Stokes equation (see Choudhuri 1998) is:

$$\rho \frac{\mathrm{d}v_i}{\mathrm{d}t} = \rho F_i - \frac{\partial P}{\partial x_i} + \frac{\partial}{\partial x_j}\left[\mu\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} - \frac{2}{3}\delta_{ij}\vec{\nabla}\cdot\vec{v}\right)\right] \tag{2.81}$$

In most situations the spatial variation of μ is not important, so one gets:

$$\rho \frac{\mathrm{d}\vec{v}}{\mathrm{d}t} = \rho\vec{F} - \vec{\nabla}P + \mu\left[\vec{\nabla}^2\,\vec{v} + \frac{1}{3}\,\vec{\nabla}(\vec{\nabla}\cdot\vec{v})\right] \tag{2.82}$$

The term containing $\vec{\nabla}(\vec{\nabla}\cdot\vec{v})$ is only important for flows with variable compression, for example in the viscous dissipation of acoustic waves. So it is neglected here and one gets the simpler version of the Navier-Stokes equations:

$$\rho \frac{\mathrm{d}\vec{v}}{\mathrm{d}t} = \rho\vec{F} - \vec{\nabla}P + \mu\vec{\nabla}^2\,\vec{v} \tag{2.83}$$

which using the kinematic viscosity $\nu = \mu/\rho$ can also be written as:

$$\frac{\partial\vec{v}}{\partial t} + (\vec{v}\cdot\vec{\nabla})\vec{v} = \vec{F} - \frac{1}{\rho}\vec{\nabla}P + \nu\vec{\nabla}^2\,\vec{v} \tag{2.84}$$

where $(\vec{v}\cdot\vec{\nabla})\vec{v}$ is the vector-gradient of v.
Note that this form of the Navier-Stokes equations only differ from the Euler equations by the additional second order spatial derivative term $\nu\vec{\nabla}^2\,\vec{v}$. But this term leads to a totally different behaviour of the solutions. The Navier-Stokes equations require more boundary conditions for solutions in finite regions. For an ideal fluid at a solid boundary the normal velocity component is set to zero, for viscous fluids one imposes the extra boundary condition that the tangential velocity component is also zero there, this then gives a unique solution.
There is an additional important property of viscous flows: Compared to ideal fluids where the production and decay of vorticity is not possible due to Kelvin's vorticity theorem for viscous fluids this is now possible.

If one wants to write the Navier-Stokes equations in an curvilinear coordinate system special care has to be taken about the right form of the vector-gradient and the second order operator. The easiest way to get this transformation right is to replace these operators with the following vector identities:

$$(\vec{v}\cdot\vec{\nabla})\vec{v} = \frac{1}{2}\vec{\nabla}(\vec{v}\cdot\vec{v}) - \vec{v}\times(\vec{\nabla}\times\vec{v}) \tag{2.85}$$

$$\vec{\nabla}^2\,\vec{v} = \vec{\nabla}(\vec{\nabla}\cdot\vec{v}) - \vec{\nabla}\times(\vec{\nabla}\times\vec{v}) \tag{2.86}$$

The simpler version of the Newtonian Navier-Stokes equations in spherical coordinates are (see Choudhuri 1998, Appendix C.4) where here the polar distance $\bar{\theta}$ is replaced by the latitude $\theta = \frac{\pi}{2} - \bar{\theta}$ and $v_\theta = -v_{\bar{\theta}}$ and $\frac{\partial}{\partial\theta} = -\frac{\partial}{\partial\bar{\theta}}$, so one changes from the right-hand coordinate system $(r, \bar{\theta}, \phi)$ to the other right-hand coordinate system (r, ϕ, θ), so that the vectors v_r and v_ϕ do not have to change sign!):

$$\begin{aligned}
&\frac{\partial v_r}{\partial t} + v_r\frac{\partial v_r}{\partial r} + \frac{v_\theta}{r}\frac{\partial v_r}{\partial\theta} + \frac{v_\phi}{r\cos(\theta)}\frac{\partial v_r}{\partial\phi} - \frac{v_\theta^2 + v_\phi^2}{r} \\
&\quad = -\frac{1}{\rho}\frac{\partial P}{\partial r} + \nu\left[\frac{1}{r}\frac{\partial^2}{\partial r^2}(rv_r) + \frac{1}{r^2}\frac{\partial^2 v_r}{\partial\theta^2} + \frac{1}{r^2\cos^2(\theta)}\frac{\partial^2 v_r}{\partial\phi^2}\right. \\
&\qquad \left. - \frac{\tan(\theta)}{r^2}\frac{\partial v_r}{\partial\theta} - \frac{2}{r^2}\frac{\partial v_\theta}{\partial\theta} - \frac{2}{r^2\cos(\theta)}\frac{\partial v_\phi}{\partial\phi} - \frac{2v_r}{r^2} + \frac{2\tan(\theta)}{r^2}v_\theta\right] + F_r, \qquad (2.87)\\
&\frac{\partial v_\theta}{\partial t} + v_r\frac{\partial v_\theta}{\partial r} + \frac{v_\theta}{r}\frac{\partial v_\theta}{\partial\theta} + \frac{v_\phi}{r\cos(\theta)}\frac{\partial v_\theta}{\partial\phi} + \frac{v_r\, v_\theta}{r} + \frac{v_\phi^2\,\tan(\theta)}{r} \\
&\quad = -\frac{1}{\rho\, r}\frac{\partial P}{\partial\theta} + \nu\left[\frac{1}{r}\frac{\partial^2}{\partial r^2}(rv_\theta) + \frac{1}{r^2}\frac{\partial^2 v_\theta}{\partial\theta^2} + \frac{1}{r^2\cos^2(\theta)}\frac{\partial^2 v_\theta}{\partial\phi^2}\right. \\
&\qquad \left. - \frac{\tan(\theta)}{r^2}\frac{\partial v_\theta}{\partial\theta} + \frac{2\,\sin(\theta)}{r^2\,\cos^2(\theta)}\frac{\partial v_\phi}{\partial\phi} + \frac{2}{r^2}\frac{\partial v_r}{\partial\theta} - \frac{v_\theta}{r^2\,\cos^2(\theta)}\right] + F_\theta, \qquad (2.88)\\
&\frac{\partial v_\phi}{\partial t} + v_r\frac{\partial v_\phi}{\partial r} + \frac{v_\theta}{r}\frac{\partial v_\phi}{\partial\theta} + \frac{v_\phi}{r\cos(\theta)}\frac{\partial v_\phi}{\partial\phi} + \frac{v_r\, v_\phi}{r} - \frac{v_\theta\, v_\phi\,\tan(\theta)}{r} \\
&\quad = -\frac{1}{\rho\, r\,\cos(\theta)}\frac{\partial P}{\partial\phi} + \nu\left[\frac{1}{r}\frac{\partial^2}{\partial r^2}(rv_\phi) + \frac{1}{r^2}\frac{\partial^2 v_\phi}{\partial\theta^2} + \frac{1}{r^2\cos^2(\theta)}\frac{\partial^2 v_\phi}{\partial\phi^2}\right. \\
&\qquad \left. - \frac{\tan(\theta)}{r^2}\frac{\partial v_\phi}{\partial\theta} + \frac{2}{r^2\,\cos(\theta)}\frac{\partial v_r}{\partial\phi} - \frac{2\,\sin(\theta)}{r^2\,\cos^2(\theta)}\frac{\partial v_\theta}{\partial\phi} - \frac{v_\phi}{r^2\,\cos^2(\theta)}\right] + F_\phi. \qquad (2.89)
\end{aligned}$$

The continuity equation in spherical coordinates is:

$$\frac{\partial\rho}{\partial t} + \frac{1}{r^2}\frac{\partial}{\partial r}(r^2\,\rho v_r) + \frac{1}{r\,\cos(\theta)}\frac{\partial}{\partial\theta}(cos(\theta)\,\rho v_\theta) + \frac{1}{r\,\cos(\theta)}\frac{\partial}{\partial\phi}(\rho v_\phi) = 0 \qquad (2.90)$$

The momenta are: $m = \rho v_r$, $n = r\rho v_\theta$ and $l = r\cos(\theta)\rho v_\phi$

The Navier-Stokes equations can now be brought in momentum form:
The radial momentum equation can be constructed by:
$\rho \cdot$ (Eq. 2.87) + (Eq. 2.90) $\cdot\, v_r$:

$$\begin{aligned}
&\frac{\partial m}{\partial t} + \frac{\partial}{\partial r}(v_r m) + \frac{1}{r^2}\frac{\partial}{\partial r}(r^2 v_r m) + \frac{1}{r\cos(\theta)}\frac{\partial}{\partial\theta}(\cos(\theta)v_\theta m) + \frac{1}{r\,\cos(\theta)}\frac{\partial}{\partial\phi}(v_\phi m) - \frac{\rho\left(v_\theta^2 + v_\phi^2\right)}{r} \\
&\quad = -\frac{\partial P}{\partial r} + \mu\left[\frac{1}{r}\frac{\partial^2}{\partial r^2}(rv_r) + \frac{1}{r^2}\frac{\partial^2 v_r}{\partial\theta^2} + \frac{1}{r^2\cos^2(\theta)}\frac{\partial^2 v_r}{\partial\phi^2}\right. \\
&\qquad \left. - \frac{\tan(\theta)}{r^2}\frac{\partial v_r}{\partial\theta} - \frac{2}{r^2}\frac{\partial v_\theta}{\partial\theta} - \frac{2}{r^2\cos(\theta)}\frac{\partial v_\phi}{\partial\phi} - \frac{2v_r}{r^2} + \frac{2\tan(\theta)}{r^2}v_\theta\right] + \rho\, F_r, \qquad (2.91)
\end{aligned}$$

The latitudinal momentum equation can be constructed by:
$r\,\rho \cdot$ (Eq. 2.88) $+ r \cdot$ (Eq. 2.90) $\cdot\, v_\theta$:

$$\begin{aligned}
&\frac{\partial n}{\partial t} + \frac{1}{r^2}\frac{\partial}{\partial r}(r^2 v_r n) + \frac{1}{r\cos(\theta)}\frac{\partial}{\partial \theta}(\cos(\theta) v_\theta n) + \frac{1}{r\ \cos(\theta)}\frac{\partial}{\partial \phi}(v_\phi n) + \rho v_\phi^2 \tan(\theta) \\
&\quad = -\frac{\partial P}{\partial \theta} + \mu\ \left[\frac{\partial^2}{\partial r^2}(r v_\theta) + \frac{1}{r}\frac{\partial^2 v_\theta}{\partial \theta^2} + \frac{1}{r\cos^2(\theta)}\frac{\partial^2 v_\theta}{\partial \phi^2}\right. \\
&\qquad \left. - \frac{\tan(\theta)}{r}\frac{\partial v_\theta}{\partial \theta} + \frac{2\ \sin(\theta)}{r\ \cos^2(\theta)}\frac{\partial v_\phi}{\partial \phi} + \frac{2}{r}\frac{\partial v_r}{\partial \theta} - \frac{v_\theta}{r\ \cos^2(\theta)}\right] + \rho r\, F_\theta,
\end{aligned} \tag{2.92}$$

And finally the angular momentum equation can be constructed by:
$r\cos(\theta)\,\rho \cdot$ (Eq. 2.89) $+ r\cos(\theta) \cdot$ (Eq. 2.90) $\cdot\, v_\phi$:

$$\begin{aligned}
&\frac{\partial l}{\partial t} + \frac{1}{r^2}\frac{\partial}{\partial r}(r^2 v_r l) + \frac{1}{r\cos(\theta)}\frac{\partial}{\partial \theta}(\cos(\theta) v_\theta l) + \frac{1}{r\cos(\theta)}\frac{\partial}{\partial \phi}(v_\phi l) \\
&\quad = -\frac{\partial P}{\partial \phi} + \mu\left[\cos(\theta)\frac{\partial^2}{\partial r^2}(r v_\phi) + \frac{\cos(\theta)}{r}\frac{\partial^2 v_\phi}{\partial \theta^2} + \frac{1}{r\cos(\theta)}\frac{\partial^2 v_\phi}{\partial \phi^2}\right. \\
&\qquad \left. - \frac{\sin(\theta)}{r}\frac{\partial v_\phi}{\partial \theta} + \frac{2}{r}\frac{\partial v_r}{\partial \phi} - \frac{2\ \tan(\theta)}{r}\frac{\partial v_\theta}{\partial \phi} - \frac{v_\phi}{r\ \cos(\theta)}\right] + r\cos(\theta)\,\rho\, F_\phi.
\end{aligned} \tag{2.93}$$

These equations can later be used to control the Newtonian limit of the relativistic Navier-Stokes equations.

2.4.1 Derivation of the Navier-Stokes Equations

The stress-energy tensor for a non-ideal plasma, as derived by Misner et al. (1973), consists of $T^{(\mathrm{nid})} = T^{(\mathrm{id})} + T^{(\mathrm{visc})} + T^{(\mathrm{heat})}$, where $T^{(\mathrm{id})}$ is the stress-energy tensor for the ideal-fluid, $T^{(\mathrm{visc})}$ are the contributions due to viscosity and $T^{(\mathrm{heat})}$ is the contribution due to heat conduction.

$$T^{\mu\nu} = \hat{\rho}\frac{u^\mu u^\nu}{c^2} + (P - \zeta\Theta)\, h^{\mu\nu} - 2\eta\sigma^{\mu\nu} + q^\mu u^\nu + q^\nu u^\mu, \tag{2.94}$$

where $h^{\mu\nu} = \frac{u^\mu u^\nu}{c^2} + g^{\mu\nu}$ is the spatial projection tensor, $\Theta = u^\mu{}_{;\mu}$ is the *expansion scalar* of the fluid, η is the *dynamic viscosity* and ζ is the coefficient for the *bulk viscosity* and q^μ is the *heat flux*. The shear tensor is defined as

$$\sigma^{\mu\nu} = \tfrac{1}{2}\left[\left(u^\mu{}_{;\rho}\right) h^{\rho\nu} + \left(u^\nu{}_{;\rho}\right) h^{\rho\mu}\right] - \tfrac{1}{3}\Theta h^{\mu\nu}. \tag{2.95}$$

Assuming a non-conducting fluid, the bulk viscosity ζ and the heat flux q^μ vanish. So with $\zeta = 0$ and $q^\mu = 0$ equation (2.94) simplifies to

$$T^{\mu\nu} = \hat{\rho}\frac{u^\mu u^\nu}{c^2} + P h^{\mu\nu} - 2\eta\sigma^{\mu\nu}. \tag{2.96}$$

Similar to the derivation of the general relativistic Euler equation, the conservation equation for the stress-energy tensor can be decomposed into two equations: the conservation of energy $u_\mu T^{\mu\nu}{}_{;\mu} = 0$,

given by the components parallel to u^ν and the conservation of momentum $h_{\mu\nu}T^{\mu\nu}{}_{;\mu} = 0$,given by the components orthogonal to u^ν.
Since Einstein's field equations (2.3) are locally linear in the stress-energy tensor, the additional terms due to the non-ideal gas can be handled as a correction to the ideal stress-energy tensor.
So only the non-ideal contribution of the stress-energy tensor $\widetilde{T}^{\mu\nu} = -2\eta\sigma^{\mu\nu}$ needs to be additionally looked at (Hujeirat and Thielemann 2009).

Modification of the Energy Equation

Heat conduction is neglected in the derivation here since it is still an underdeveloped research field, for an insight the reader is referred to Maartens (1996) and Geroch (1995).
Heating due to viscous terms in the stress-energy tensor is not derived from adding the longitudinal components of $u_\mu \widetilde{T}^{\mu\nu}{}_{;\mu}$ to Eq. (2.65), but is included approximately by adding a quantity Λ to the relativistic internal energy density equation

$$\partial_t \epsilon^{\mathrm{d}} + \frac{1}{\sqrt{-g}}\partial_{x_k}\left(\sqrt{-g}\,\epsilon^{\mathrm{d}} V^k\right) = -P\left[\partial_t\left(\frac{u^t}{c}\right) + \frac{1}{\sqrt{-g}}\,\partial_{x_k}\left(\sqrt{-g}\,\frac{u^t}{c}V^k\right)\right] + \Lambda. \tag{2.97}$$

Λ may consist of $\Lambda = \Lambda_{\mathrm{visc}} + \Lambda_{\mathrm{br}}$, where Λ_{visc} is the *heating term* due to dissipation or viscous heating, and Λ_{br} is the *cooling term* due to thermal bremsstrahlung.

Modification of the Momentum Equation

The modifications of the momentum conservation equation are given by the spatial projection of the viscous terms of the stress-energy tensor along the vector normal to the hyperspace and are given by

$$L2^a{}_{r\theta} := \widetilde{T}^{\mu a}{}_{;\mu} = \widetilde{T}^{\mu a}{}_{,\mu} + \widetilde{T}^{\zeta a}\Gamma^\mu{}_{\zeta\mu} + \widetilde{T}^{\mu\zeta}\Gamma^a{}_{\zeta\mu}, \tag{2.98}$$

where $L2^a{}_{r\theta}$ contains terms due to viscosity and terms with the second order derivatives of the velocity, therefore they are called **second order viscous operators**.
For the derivation the connection coefficients $\Gamma^\zeta{}_{\mu\lambda}$ are calculated using equation (2.2).
For the Boyer-Lindquist coordinate system with $g^{tt}, g^{rr}, g^{\phi\phi}, g^{\theta\theta}$ and $g^{\phi t}$, the result presented by Richardson and Chung (2002) are used. Since only 3D-axial symmetry around the rotation axis z is considered here[3], one can set $\partial_\phi g^{\mu\nu} = 0$, which then results in (Hujeirat and Thielemann 2009) (here: c=G=1 !!!):

$$\begin{aligned}
\mathrm{L2}^{\mathrm{r}}_{\mathrm{r}\theta} = \quad & \bar{\nabla}_{\mathrm{r}} \cdot \eta \quad \Big[\left(\tfrac{\partial \mathrm{u}^{\mathrm{r}}}{\partial \mathrm{r}} + \tfrac{1}{2}(\mathrm{g}^{\mathrm{rr}}\tfrac{\partial \mathrm{g}_{\mathrm{rr}}}{\partial \mathrm{r}})\mathrm{u}^{\mathrm{r}}\right)(\mathrm{u}_{\mathrm{r}}\mathrm{u}^{\mathrm{r}} + 1) \\
& \qquad + \tfrac{\partial \mathrm{u}_{\mathrm{r}}}{\partial \mathrm{r}} - \tfrac{1}{2}(\mathrm{g}^{\mathrm{rr}}\tfrac{\partial \mathrm{g}_{\mathrm{rr}}}{\partial \mathrm{r}})\mathrm{u}^{\mathrm{r}}\right)((\mathrm{u}^{\mathrm{r}})^2 + \mathrm{g}^{\mathrm{rr}}) \\
& \qquad - \tfrac{2}{3}(\nabla_{\mathrm{r}} \cdot \mathrm{u}^{\mathrm{r}} + \nabla_\theta \cdot \mathrm{u}^\theta)(\mathrm{u}_{\mathrm{r}}\mathrm{u}^{\mathrm{r}} + 1)\Big] \\
+ \quad & \bar{\nabla}_\theta \cdot \eta \quad \Big[\left(\tfrac{\partial \mathrm{u}^\theta}{\partial \mathrm{r}} + \tfrac{1}{2}(\mathrm{g}^{\theta\theta}\tfrac{\partial \mathrm{g}_{\theta\theta}}{\partial \mathrm{r}})\right)(\mathrm{u}_{\mathrm{r}}\mathrm{u}^{\mathrm{r}} + 1) \\
& \qquad + \tfrac{\partial \mathrm{u}^\theta}{\partial \theta} + \tfrac{1}{2}(\mathrm{g}^{\theta\theta}\tfrac{\partial \mathrm{g}_{\theta\theta}}{\partial \theta})\right)(\mathrm{u}_{\mathrm{r}}\mathrm{u}^\theta) \\
& \qquad + \tfrac{\partial \mathrm{u}_{\mathrm{r}}}{\partial \mathrm{r}} - \tfrac{1}{2}(\mathrm{g}^{\mathrm{rr}}\tfrac{\partial \mathrm{g}_{\mathrm{rr}}}{\partial \mathrm{r}})\mathrm{u}_{\mathrm{r}}\right)(\mathrm{u}_{\mathrm{r}}\mathrm{u}^\theta) \\
& \qquad + \tfrac{\partial \mathrm{u}_{\mathrm{r}}}{\partial \theta} - \tfrac{1}{2}(\mathrm{g}^{\mathrm{rr}}\tfrac{\partial \mathrm{g}_{\mathrm{rr}}}{\partial \theta})\mathrm{u}_{\mathrm{r}}\right)((\mathrm{u}^\theta)^2 + \mathrm{g}^{\theta\theta}) \\
& \qquad - \tfrac{2}{3}(\nabla_{\mathrm{r}} \cdot \mathrm{u}^{\mathrm{r}} + \nabla_\theta \cdot \mathrm{u}^\theta)(\mathrm{u}_{\mathrm{r}}\mathrm{u}^{\mathrm{r}} + 1)\Big],
\end{aligned} \tag{2.99}$$

[3] up to now Astro-GRIPS is 3D-axisymmetric (some people call this 2.5D) and not fully 3D. An extension to full 3D is a future research topic

$$
\begin{aligned}
\mathrm{L2}^{\theta}_{\mathrm{r}\theta} &= \bar{\nabla}_{\mathrm{r}} \cdot \eta \quad [(\tfrac{\partial \mathrm{u}_\theta}{\partial \mathrm{r}} - \tfrac{1}{2}(\mathrm{g}^{\theta\theta}\tfrac{\partial \mathrm{g}_{\theta\theta}}{\partial \mathrm{r}})\mathrm{u}_\theta)\,(\mathrm{u}^{\mathrm{r}})^2 + \mathrm{g}^{\mathrm{rr}}) \\
&\qquad -\tfrac{2}{3}(\nabla_{\mathrm{r}} \cdot \mathrm{u}^{\mathrm{r}} + \nabla_\theta \cdot \mathrm{u}^\theta)\,(\mathrm{u}_\theta \mathrm{u}^{\mathrm{r}})] \\
&+ \bar{\nabla}_\theta \cdot \eta \quad [(\tfrac{\partial \mathrm{u}^\theta}{\partial \mathrm{r}} + \tfrac{1}{2}(\mathrm{g}^{\theta\theta}\tfrac{\partial \mathrm{g}_{\theta\theta}}{\partial \mathrm{r}})\,(\mathrm{u}_\theta \mathrm{u}^\theta + 1) \\
&\qquad -\tfrac{2}{3}(\nabla_{\mathrm{r}} \cdot \mathrm{u}^{\mathrm{r}} + \nabla_\theta \cdot \mathrm{u}^\theta)\,(\mathrm{u}_\theta \mathrm{u}^\theta + 1)] \\
&\qquad +(\tfrac{\partial \mathrm{u}^\theta}{\partial \theta} - \tfrac{1}{2}(\mathrm{g}^{\theta\theta}\tfrac{\partial \mathrm{g}_{\theta\theta}}{\partial \theta})\mathrm{u}^\theta)\,(\mathrm{u}^\theta)^2 + \mathrm{g}^{\theta\theta})],
\end{aligned}
\tag{2.100}
$$

$$
\begin{aligned}
\mathrm{L2}^{\varphi}_{\mathrm{r}\theta} &= \bar{\nabla}_{\mathrm{r}} \cdot \eta \quad [(\tfrac{\partial \mathrm{u}_\varphi}{\partial \mathrm{r}} - \tfrac{1}{2}(\mathrm{g}^{\varphi\mathrm{t}}\tfrac{\partial \mathrm{g}_{\varphi\mathrm{t}}}{\partial \mathrm{r}} + \mathrm{g}^{\varphi\varphi}\tfrac{\partial \mathrm{g}_{\varphi\varphi}}{\partial \mathrm{r}})\mathrm{u}_\varphi)\,(\mathrm{u}^{\mathrm{r}})^2 + \mathrm{g}^{\mathrm{rr}}) \\
&\qquad +(\tfrac{\partial \mathrm{u}_\varphi}{\partial \theta} - \tfrac{1}{2}(\mathrm{g}^{\varphi\mathrm{t}}\tfrac{\partial \mathrm{g}_{\varphi\mathrm{t}}}{\partial \theta} + \mathrm{g}^{\varphi\varphi}\tfrac{\partial \mathrm{g}_{\varphi\varphi}}{\partial \theta})\mathrm{u}_\varphi)\,(\mathrm{u}^{\mathrm{r}}\mathrm{u}^\theta) \\
&\qquad -(\tfrac{1}{2}(\mathrm{g}^{\mathrm{rr}}\tfrac{\partial \mathrm{g}_{\varphi\mathrm{t}}}{\partial \mathrm{r}})\,\mathrm{u}^\varphi \mathrm{u}_\varphi \mathrm{u}^{\mathrm{t}} - (\tfrac{1}{2}(\mathrm{g}^{\mathrm{rr}}\tfrac{\partial \mathrm{g}_{\varphi\varphi}}{\partial \mathrm{r}})\,\mathrm{u}^\varphi(\mathrm{u}_\varphi \mathrm{u}^\varphi + 1)] \\
&+ \bar{\nabla}_\theta \cdot \eta \quad [(\tfrac{\partial \mathrm{u}_\varphi}{\partial \mathrm{r}} - \tfrac{1}{2}(\mathrm{g}^{\varphi\mathrm{t}}\tfrac{\partial \mathrm{g}_{\varphi\mathrm{t}}}{\partial \mathrm{r}} + \mathrm{g}^{\varphi\varphi}\tfrac{\partial \mathrm{g}_{\varphi\varphi}}{\partial \mathrm{r}})\,(\mathrm{u}^{\mathrm{r}}\mathrm{u}^\theta) \\
&\qquad +(\tfrac{\partial \mathrm{u}_\varphi}{\partial \theta} - \tfrac{1}{2}(\mathrm{g}^{\varphi\mathrm{t}}\tfrac{\partial \mathrm{g}_{\varphi\mathrm{t}}}{\partial \theta} + \mathrm{g}^{\varphi\varphi}\tfrac{\partial \mathrm{g}_{\varphi\varphi}}{\partial \theta})\mathrm{u}_\varphi)\,((\mathrm{u}^\theta)^2 + \mathrm{g}^{\theta\theta} \\
&\qquad -(\tfrac{1}{2}\mathrm{g}^{\theta\theta}\tfrac{\partial \mathrm{g}_{\varphi\mathrm{t}}}{\partial \theta})\,\mathrm{u}^\varphi \mathrm{u}_\varphi \mathrm{u}^{\mathrm{t}} - (\tfrac{1}{2}\mathrm{g}^{\theta\theta}\tfrac{\partial \mathrm{g}_{\varphi\varphi}}{\partial \theta})\,\mathrm{u}^\varphi(\mathrm{u}_\varphi \mathrm{u}^\varphi + 1)],
\end{aligned}
\tag{2.101}
$$

Due to the lengthy form of these equations, the computational cost will be too high for an efficient numerical simulation, so these terms will be simplified by the approximation of only using second order, mixed-free and Laplace-like operators. This simplification is justified if one simulates in 3D-axis symmetry and is mainly interested in the angular momentum transport, which plays the most important role in accretion physics, and results in (here: c=G=1 !!!):

$$
\begin{aligned}
\widetilde{\mathrm{L2}}^{\mathrm{r}}_{\mathrm{r}\theta} &= \bar{\nabla}_{\mathrm{r}} \cdot \{\eta \quad [2\tfrac{\partial \mathrm{u}^{\mathrm{r}}}{\partial \mathrm{r}} - \tfrac{2}{3}(\bar{\nabla}_{\mathrm{r}} \cdot \mathrm{u}^{\mathrm{r}})]\}\,(\mathrm{u}_{\mathrm{r}}\mathrm{u}^{\mathrm{r}} + 1) \\
&+ \bar{\nabla}_\theta \cdot \{\eta \quad (\tfrac{\partial \mathrm{u}_{\mathrm{r}}}{\partial \theta})\}\,((\mathrm{u}^\theta)^2 + \mathrm{g}^{\theta\theta}),
\end{aligned}
\tag{2.102}
$$

$$
\begin{aligned}
\widetilde{\mathrm{L2}}^{\theta}_{\mathrm{r}\theta} &= \bar{\nabla}_{\mathrm{r}} \cdot \{\eta \quad [(\tfrac{\partial \mathrm{u}_\theta}{\partial \mathrm{r}})\,(\mathrm{u}^{\mathrm{r}})^2 + \mathrm{g}^{\mathrm{rr}}) \\
&+ \bar{\nabla}_\theta \cdot \{\eta \quad (\tfrac{\partial \mathrm{u}^\theta}{\partial \theta}) - \tfrac{2}{3}\bar{\nabla}_\theta \cdot \mathrm{u}^\theta)\,(\mathrm{u}_\theta \mathrm{u}^\theta + 1) + (\tfrac{\partial \mathrm{u}^\theta}{\partial \theta})\,(\mathrm{u}^\theta)^2 + \mathrm{g}^{\theta\theta})]\},
\end{aligned}
\tag{2.103}
$$

$$
\begin{aligned}
\widetilde{\mathrm{L2}}^{\varphi}_{\mathrm{r}\theta} &= \bar{\nabla}_{\mathrm{r}} \cdot \eta \quad [(\tfrac{\partial \mathrm{u}_\varphi}{\partial \mathrm{r}} - \tfrac{1}{2}(\mathrm{g}^{\varphi\mathrm{t}}\tfrac{\partial \mathrm{g}_{\varphi\mathrm{t}}}{\partial \mathrm{r}} + \mathrm{g}^{\varphi\varphi}\tfrac{\partial \mathrm{g}_{\varphi\varphi}}{\partial \mathrm{r}})\mathrm{u}_\varphi)\,(\mathrm{u}^{\mathrm{r}})^2 + \mathrm{g}^{\mathrm{rr}}) \\
&\qquad -(\tfrac{1}{2}(\mathrm{g}^{\mathrm{rr}}\tfrac{\partial \mathrm{g}_{\varphi\mathrm{t}}}{\partial \mathrm{r}})\,\mathrm{u}^\varphi \mathrm{u}_\varphi \mathrm{u}^{\mathrm{t}} - (\tfrac{1}{2}(\mathrm{g}^{\mathrm{rr}}\tfrac{\partial \mathrm{g}_{\varphi\varphi}}{\partial \mathrm{r}})\,\mathrm{u}^\varphi(\mathrm{u}_\varphi \mathrm{u}^\varphi + 1)] \\
&+ \bar{\nabla}_\theta \cdot \eta \quad (\tfrac{\partial \mathrm{u}_\varphi}{\partial \theta} - \tfrac{1}{2}(\mathrm{g}^{\varphi\mathrm{t}}\tfrac{\partial \mathrm{g}_{\varphi\mathrm{t}}}{\partial \theta} + \mathrm{g}^{\varphi\varphi}\tfrac{\partial \mathrm{g}_{\varphi\varphi}}{\partial \theta})\mathrm{u}_\varphi)\,((\mathrm{u}^\theta)^2 + \mathrm{g}^{\theta\theta}) \\
&\qquad -(\tfrac{1}{2}\mathrm{g}^{\theta\theta}\tfrac{\partial \mathrm{g}_{\varphi\mathrm{t}}}{\partial \theta})\,\mathrm{u}^\varphi \mathrm{u}_\varphi \mathrm{u}^{\mathrm{t}} - (\tfrac{1}{2}\mathrm{g}^{\theta\theta}\tfrac{\partial \mathrm{g}_{\varphi\varphi}}{\partial \theta})\,\mathrm{u}^\varphi(\mathrm{u}_\varphi \mathrm{u}^\varphi + 1)].
\end{aligned}
\tag{2.104}
$$

2.4.2 Summary of the Navier-Stokes Equations

The **general relativistic Navier-Stokes equations** for a non-ideal, non-conducting fluid are:

$$
\partial_t D + \frac{1}{\sqrt{-g}}\partial_k\left(\sqrt{-g}DV^k\right) = 0, \tag{2.105a}
$$

$$
\partial_t M_a + \frac{1}{\sqrt{-g}}\partial_k\left(\sqrt{-g}M_a V^k\right) = -\frac{S_\lambda M_\mu\, c}{2M^t}g^{\lambda\mu}{}_{,a} - \partial_a P + \widetilde{L2}^{a}{}_{r\theta}, \tag{2.105b}
$$

$$
\partial_t \epsilon^{\mathrm{d}} + \frac{1}{\sqrt{-g}}\partial_k\left(\sqrt{-g}\,\epsilon^{\mathrm{d}}V^k\right) = -P\left[\partial_t\left(\frac{u^t}{c}\right) + \frac{1}{\sqrt{-g}}\,\partial_k\left(\sqrt{-g}\,\frac{u^t}{c}V^k\right)\right] + \Lambda, \tag{2.105c}
$$

with the conservative variables $D = \frac{u^t}{c}\rho = \Gamma\rho$, $M_\mu = \rho\tilde{h}\Gamma u_\mu = \overline{\mathrm{D}}u_\mu$, $\epsilon^{\mathrm{d}} = \epsilon D$ and the second-order approximated corrections $\widetilde{L2}^{a}{}_{r\theta}$ due to viscosity, see (2.104). $\Lambda = \Lambda_1 + \Lambda_2 + \ldots$ are additional heating or cooling terms.

2.5 Equation of State - the Closure of the Hydrodynamic System of Equations

To close the system of equations, either Euler equations or Navier-Stokes equations, one has to now an equation of state (EoS), which describes the fluid pressure as a function of the internal properties of the fluid. Usually the EoS is of the form $P = P(\rho, \epsilon, ...)$, but more parameters may be involved e.g. its composition, the ionisation state of the gas or the adiabatic index γ. Because in general the real EoS can be quiet complex, one usually uses a simplification.
The *polytropic equation of state* has the form $P = K\rho^{\gamma}$, where K is a constant depending on the species of the gas and γ the polytropic index.
In this case there is usually no need to solve the energy equation, since there is no coupling between it and the momenta equations through the equation of state and if one assumes that there are no energy (only pressure) dependent source terms in the momenta equations.
Due to its simplicity this equation of state was mostly used in the study of hydrostatic equilibria of stellar atmospheres, neutron stars and globular clusters where it leads to the famous *Lane-Emden equations*.
For hydrodynamical simulations the equation of state of an ideal gas $P = (\gamma - 1)\rho\epsilon = (\gamma - 1)\frac{\epsilon^d}{\Gamma}$ is often used. The adiabatic index γ is determined by the composition of the gas, e.g. for a mononuclear gas it has the value of $\gamma = 5/3$.
At very high temperatures where particles move with relativistic speeds another equation of state has to be used. In the *ultra-relativistic limit* the equation of state is very similar to the Newtonian EoS, but the adiabatic index for a mononuclear gas is then $\gamma = 4/3$.
For flows with velocities between the non-relativistic and the ultra-relativistic limit a generalized EoS has to be applied.

2.5.1 Equation of State for a relativistic fluid

With the Newtonian adiabatic index of $\gamma = 5/3$ in the relativistic regime with high temperatures the equation of state of an ideal gas would give a sound speed which exceeds the speed of light. To avoid this contradiction with causality, Taub (1948) showed that the EoS of a relativistic gas has to obey certain rules, to ensure that the speed of sound is always lower than the speed of light.

By using the relativistic Maxwell-Boltzmann distribution function, called *Maxwell-Juttner distribution*, Synge (1957) found the correct EoS which is valid in the whole range of fluid velocities.
The relativistic enthalpy density per c^2 denoted by $\tilde{h} = h/c^2$ of the gas is given by

$$\tilde{h} = \frac{K_3(1/\Theta)}{K_2(1/\Theta)} \qquad \text{(EoS of Synge)}, \tag{2.106}$$

where $\Theta = P/\rho$ is the pressure-density ratio and K_2, K_3 are the modified Bessel functions of second and third order.

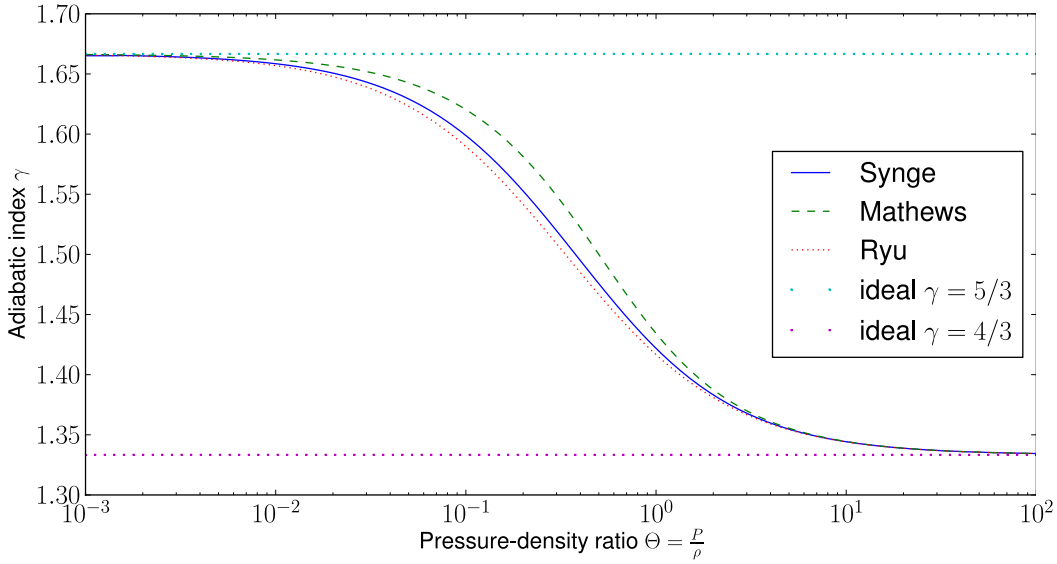


Figure 2.5: *Adiabatic index γ of an ideal gas dependent on the pressure-density ratio $\Theta = \frac{p}{\rho}$ for the Synge equation of state and the approximations of Mathews and Ryu. (Plot taken from Hilscher (2009))*

Since the solution of this correct EoS is numerically expensive, one approximation was presented by Mathews (1971) [4]:

$$\tilde{h} = \tfrac{5}{2}\Theta + \tfrac{3}{2}\sqrt{\Theta^2 + \tfrac{4}{9}} \qquad \text{(EoS of Mathews).} \tag{2.107}$$

Another approximation was found by Ryu et al. (2006), which is simpler in its form but reproduces Synge's solution even better:

$$\tilde{h} = 2\frac{6\Theta^2 + 4\Theta + 1}{3\Theta + 2} \qquad \text{(EoS of Ryu).} \tag{2.108}$$

Unfortunately, it is not possible to give a direct expression for the pressure. Instead, first the relativistic enthalpy is determined by finding the root of

$$\tilde{h}(\Theta) - \frac{\epsilon^{\mathrm{d}}}{D\,c^2} - \Theta - 1 = 0. \tag{2.109}$$

Using the Newton-Raphson iteration method[5] gives excellent convergence behaviour and Ryu's EoS performs slightly better than Mathews EoS.

The enthalpy $\tilde{h}(\Theta)$ can now be easily calculated from the found root Θ. Solving the normalisation equation $M_\mu M^\mu = -c^2\overline{\mathrm{D}}^2 = -c^2(D\tilde{h})^2$ for M_t and using $M^t = g^{tt}M_t + g^{t\phi}M_\phi$ and $u^t = M^t/(D\tilde{h})$ the

[4] This approximation is used by Mignone et al. (2007) in the PLUTO code `http://plutocode.to.astro.it`.

[5] Therefore one has to calculate the derivative of $\tilde{h}(\Theta)$, which is e.g. for Ryu's EoS (2.108):
$\frac{\partial}{\partial\Theta}\left(2\frac{6\Theta^2+4\Theta+1}{3\Theta+2} - \frac{\epsilon^{\mathrm{d}}}{Dc^2} - \Theta - 1\right) = \frac{2(4+12\Theta)}{2+3\Theta} - \frac{6\left(1+4\Theta+6\Theta^2\right)}{(2+3\Theta)^2} - 1$

pseudo Lorentz-factor $\Gamma = \frac{u^t}{c}$ can be determined and then the pressure can be calculated by $P = \Theta \cdot \frac{D}{\Gamma}$.

For the ideal gas law the enthalpy is calculated by $\tilde{h} = 1 + \frac{\gamma\Theta}{(\gamma-1)}$, which can be solved for γ resulting in:

$$\gamma(\Theta) = \frac{\gamma_h}{\gamma_h - 1} \qquad \text{with } \gamma_h = \frac{\tilde{h} - 1}{\Theta}. \tag{2.110}$$

Using the generalisation that now the adiabatic index γ is not a constant anymore, the ideal equation of state reads:

$$P = (\gamma(\Theta) - 1)\,\rho\epsilon = (\gamma(\Theta) - 1)\,\frac{\epsilon^d}{\Gamma}. \tag{2.111}$$

The dependence of γ on Θ is shown in Fig. (2.5) for the exact Synge equations of state and the approximations of *Mathews* and *Ryu*. It can be seen that both approximations fit very well and approach the Newtonian limit with $\gamma = 5/3$ and the ultra-relativistic limit with $\gamma = 4/3$.

The equation of state for an ideal gas is:

$$P = (\gamma - 1)\rho\epsilon. \tag{2.112}$$

Following the ideal gas law: $PV = n\mathcal{R}_{gas}T$, with pressure P, volume V and amount of gas n in moles, temperature T and the molar gas constant $\mathcal{R}_{gas} = 8.314\frac{\text{J}}{\text{K mol}}$ introducing the molecular mass $\mu_{gas} = \frac{m}{n}$ in g/mol and taking into account the definition of the density $\rho = \frac{m}{V}$ one gets for the temperature in K:

$$T = \frac{\mu_{gas}}{\mathcal{R}_{gas}}\,\frac{P}{\rho}. \tag{2.113}$$

The molar mass for atomic hydrogen is $\mu_H = 1.00797$ g/mol, for hydrogen molecules $\mu_{H_2} = 2\,\mu_H$ and for ionized hydrogen plasma $\mu_{gas} \approx 0.5$ g/mol.

The sound speed of a relativistic ideal gas is

$$V_s = \sqrt{\frac{\gamma\, p}{\rho\,\tilde{h}}} = \sqrt{\frac{\gamma\,\Theta}{\tilde{h}}}. \tag{2.114}$$

The dependence of the sound speed on $\Theta = \frac{P}{\rho}$ and its comparison to the Newtonian sound speed can be seen in figure 2.6.

The eigenvalues of the one-dimensional special relativistic hydrodynamic Euler equations for a sound speed $c_s = v_s = 0.5\,c$ dependent on the advection speed $v = v_E$ of the fluid can be seen in figure 2.7. The eigenvalues are (see Aloy et al. 1999, Appendix A):

$$a_1 = \frac{(1 - c_s^2)\,v - \sqrt{(1 - v^2)\,c_s^2\,(1 - v^2\,c_s^2 - (1 - c_s^2)\,v^2)}}{(1 - v^2\,c_s^2)} \tag{2.115}$$

$$a_2 = a_3 = a_4 = v \tag{2.116}$$

$$a_5 = \frac{(1 - c_s^2)\,v + \sqrt{(1 - v^2)\,c_s^2\,(1 - v^2\,c_s^2 - (1 - c_s^2)\,v^2)}}{(1 - v^2\,c_s^2)} \tag{2.117}$$

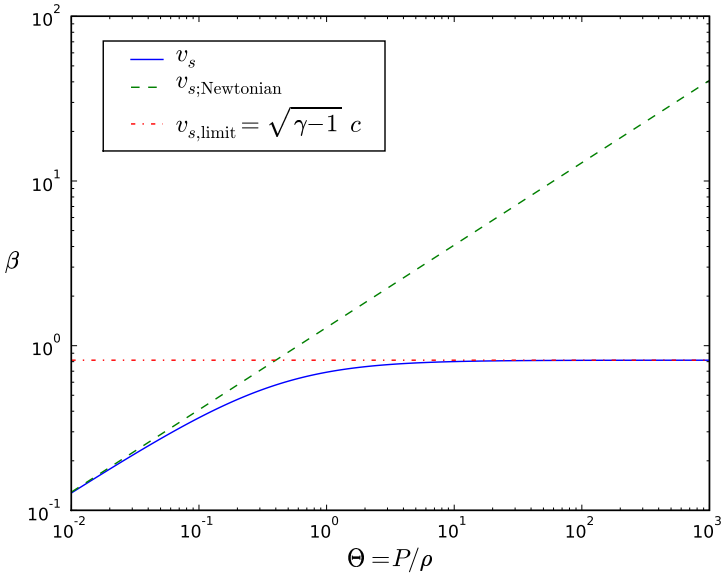


Figure 2.6: *Relativistic and Newtonian sound speed v_s and v_s; Newtonian in units of $\beta = \frac{v}{c}$ dependent on $\Theta = \frac{P}{\rho}$ for an adiabatic index $\gamma = \frac{5}{3}$. In the relativistic case there exists a limiting sound speed of $v_{s,limit} = \sqrt{\gamma - 1}\, c < c$.*

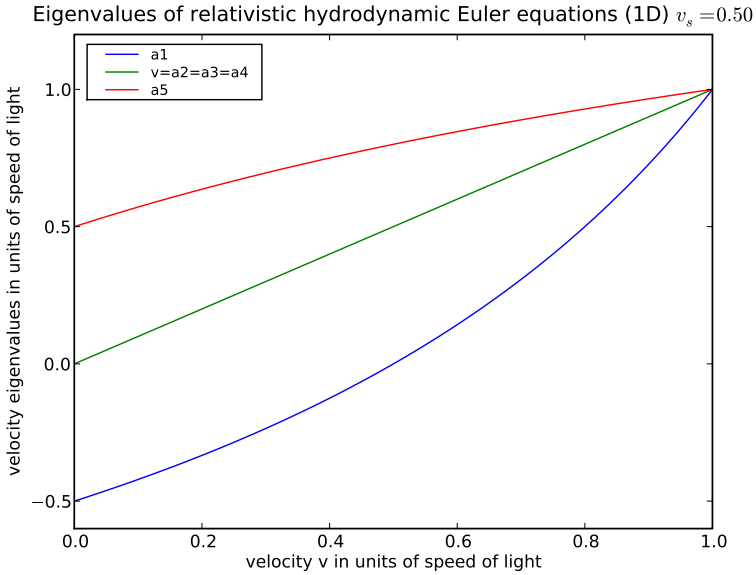


Figure 2.7: *Eigenvalues of the special relativistic hydrodynamic Euler equations for a sound speed $c_s = v_s = 0.5\ c$ dependent on the advection speed v of the fluid. If the fluid approaches the speed of light c all eigenvalues will fall together to c.*

The largest eigenvalue is used in the calculation of the explicit time step.

For a fluid moving along the *x*-direction the eigenvalues, the characteristic speeds, of the conservative Eulerian formulation of the general relativistic hydrodynamics equations are (see Font 2008, eqns. (36), (37)):

$$\lambda_0 = \alpha {v_E}^x - \beta^x \quad (triple), \tag{2.118}$$

$$\lambda_\pm = \frac{\alpha}{1 - {v_E}^2\, c_s^2} \left\{ {v_E}^x (1 - c_s^2) \pm c_s \sqrt{(1 - {v_E}^2)\left[g^{xx}(1 - {v_E}^2\, c_s^2) - {v_E}^x {v_E}^x (1 - c_s^2) \right]} \right\} - \beta^x \tag{2.119}$$

where c_s is the speed of sound and ${v_E}^x$ the velocity component in *x*-direction of the velocity in the local Euler frame.
This reduces to the special relativistic expressions if one inserts the Minkowski metric and also to the Newtonian ones ($\lambda_0 = {v_E}^x, \lambda_\pm = {v_E}^x \pm c_s$).
In the case of general relativity also the largest eigenvalue is used in the calculation of the explicit time step.

3 Numerics of General Relativistic Euler and Navier-Stokes Equations

In this chapter first a general overview about the numerical methods in Astrophysical Fluid Dynamics (AFD) is given, then the discretization of the General Relativistic Euler and Navier-Stokes Equations using a finite volume approach is shown. After the description of implicit methods used to solve these equations several iterative linear equation solvers, which are used therefore, are presented.

3.1 Numerical Methods and Time Scales in Astrophysical Fluid Dynamics (AFD)

3.1.1 Numerical Methods in AFD

Astrophysical fluid dynamics (AFD) deals with the properties and movement of gaseous-matter or plasma under a wide variety of circumstances. Most astrophysical fluid flows evolve over a large variety of different time and length scales and have a complicated structure, henceforth making their analytical treatment unfeasible.

That is where numerical simulations come into play: Due to the rapid development of computer hardware technology during the last two or three decades also the numerical treatment of Astrophysical problems by means of computer codes has grown exponentially.

Nowadays, a vast majority of numerical simulation codes are capable of treating large and sophisticated multi-scale fluid problems with high resolution and even in three-dimensions.

The numerical methods employed in AFD can basically be classified into two categories:

1. Microscopic methods, based on the treatment of single real or pseudo-particles:
   N-body (NB), Monte-Carlo (MC) and Smoothed Particle Hydrodynamics (SPH),
   which might already also belong to the second category
2. Macroscopic methods, mostly grid oriented methods based on a common statistical treatment of particles in a small volume element (hydrodynamics):
   finite difference (FDM), finite volume (FVM) and finite-element methods (FEM).

Most numerical methods used in AFD are conditionally-stable, which means that they may converge if the Courant-Friedrichs-Lewy condition for stability is fulfilled. For compressible flows that are strongly time-dependent these methods are very efficient. They may stagnate however, if important physical effects with other time-scales, like cooling, are to be considered or even if the flow is almost incompressible (like the flow of water compared to that of the air).

On the other hand, only a small number of the numerical methods employed in AFD are unconditionally stable, the implicit methods.
But implicit methods are much more effort-demanding from the programming point of view compared to explicit methods, since they contain a solution of a matrix equation.

It has been shown that strongly implicit (henceforth IM) and explicit (henceforth EM) methods are different variants of the same algebraic problem (Hujeirat 2005a,c,b), hence both methods can be unified within one simulation code.

Table 3.1 gives an overview over several relevant properties of some of the numerical simulation programs used in Astrophysics.

3.1.2 Time Scales in AFD

In nature there exist many different time scales in accretion phenomena. Following Hujeirat, Keil, and Heitsch (2007) they are described here:

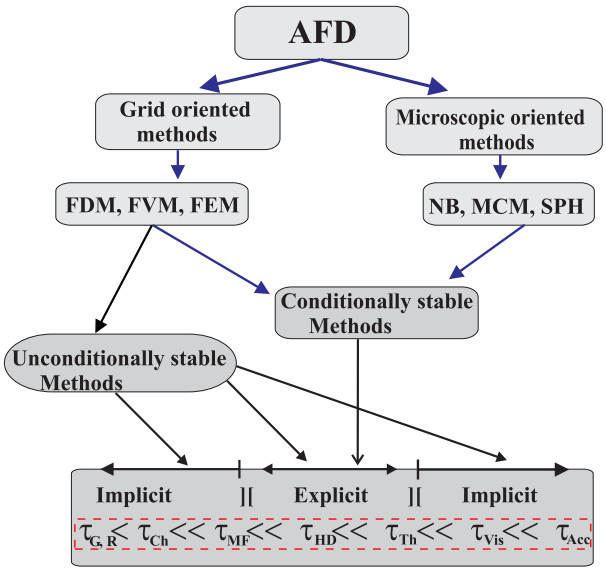


Figure 3.1: *The mostly used different numerical methods in Astrophysical Fluid Dynamics: finite difference (FDM), finite volume (FVM), finite element (FEM), N-Body (NB), Monte Carlo (MCM) and the smoothed particle hydrodynamics (SPH) and their possible regime of application from the time scale point of view. The time scales are: the radiative-$\tau_{\rm R}$, gravitative-$\tau_{\rm G}$, chemical-$\tau_{\rm Ch}$, magnetic-$\tau_{\rm MF}$, hydrodynamic-$\tau_{\rm HD}$, thermal-$\tau_{\rm Th}$, viscous-$\tau_{\rm Vis}$, and the accretion time scale-$\tau_{\rm Acc}$. (plot from Hujeirat et al. 2007).*

Assume a box of $L \times L \times L$ dimensions filled with a rotating multi-component gaseous-matter is given. The fluid is said to be radiating, magnetized, chemical-reacting, partially ionized and under the influence of its own or/and external gravitational field. The state of the gas may be described by characteristic sizes of velocity, density, temperature and magnetic field. The (approximate) time-scales associated with the flow can be obtained directly from the dimensional analysis of the radiative MHD-equations as follows (see Hujeirat 2005a, for detailed description of the set of equations see for example).

	Explicit	Implicit	HSS
solution method	$q^{n+1} = q^n + \delta t\, d^n$	$q^{n+1} = q^n + \delta t \tilde{A}^{-1} d^*$	$q^{n+1} = \alpha q^n + (1-\alpha)\delta t \tilde{A}_d^{-1} d^*$
Type of flows	Strongly time-dependent, compressible, weakly dissipative HD and MHD in 1, 2 and 3 dimensions	Stationary, quasi-stationary, highly dissipative, radiative and axi-symmetric MHD-flows in 1, 2 and 3 dimensions	Stationary, quasi-stationary, weakly compressible, highly dissipative, radiative and axi-symmetric MHD-flows in 1, 2 and 3 dimensions
Stability	conditioned	unconditioned	unconditioned
Efficiency	1 (normalized/2D)	$\sim m^2$	$\sim m_d^2$
Efficiency: Enhancement strategies	Parallelization	Parallelization, preconditioning, multigrid	HSS, parallelization, preconditioning, prolongation
Robustness: Enhancement strategies	i. subtime-stepping ii. stiff terms are solved semi-implicitly	i. multiple iteration ii. reducing the time step size	i. multiple iteration ii. reducing the time step size, HSS
Numerical Codes Newtonian	Solvers1[a] ZEUS&ATHENA[b], FLASH[c], NIRVANA[d], PLUTO[e], VAC[f]	Solver2[g]	IRMHD[h]
Numerical Codes Relativistic	Solvers3[i] GRMHD[j], ENZO[k], PLUTO[l], HARM[m], RAISHIN[n], RAM[o], GENESIS[p], WHISKY[q]	Solver4[r]	GR-I-RMHD[s], Astro-GRIPS[t]

Table 3.1: A list of only a part of the grid-oriented codes in AFD and their algorithmic properties. In these equations, $q^{n,n+1}$, δt, $\tilde{A}$, α and d^* denote the vector of variables from the old and new time levels, time step size, a preconditioning matrix, a switch on/off parameter and a time-modified defect vector, respectively. “m" in row 4 denotes the bandwidth of the corresponding matrix.

[a] Bodenheimer et al. (1978); Clarke (1996), [b] Stone and Norman (1992); Gardiner and Stone (2006), [c] Fryxell et al. (2000), [d] Ziegler (1998), [e] Mignone and Bodo (2003); Mignone et al. (2007), [f] Toth et al. (1998), [g] Wuchterl (1990); Swesty (1995), [h] Hujeirat (1995, 2005a); Falle (2003), [i] Koide et al. (1999); Komissarov (2004), [j] De Villiers and Hawley (2003), [k] O'Shea et al. (2004), [l] Mignone et al. (2007), [m] Gammie et al. (2003), [n] Mizuno et al. (2006), [o] Zhang and MacFadyen (2006), [p] Aloy et al. (1999), [q] Baiotti et al. (2003), [r] Liebendörfer et al. (2002), [s] Hujeirat et al. (2008). [t] Astro-GRIPS, the General Relativistic Implicit Parallel Solver, the more user-friendly, optimized and parallelized simulation code similar to GR-I-RMHD, which is described in this work here.

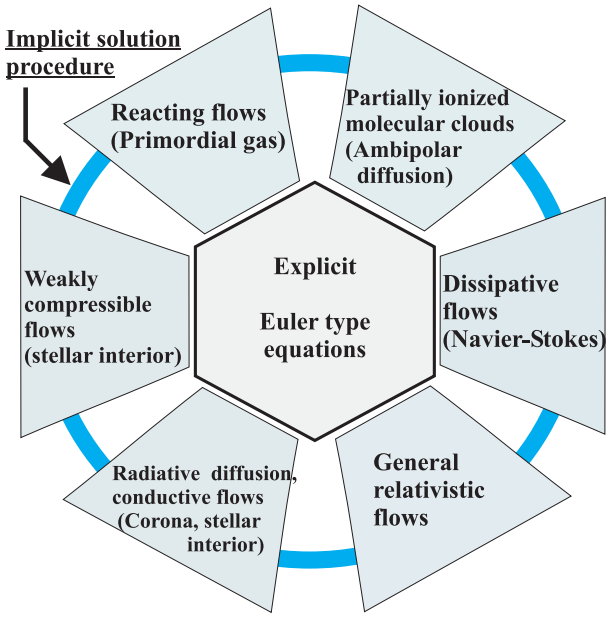


Figure 3.2: *The regime of application of the explicit methods is severely limited to Euler-type flows, whereas sophisticated treatment of most flow-problems in AFD require the employment of much more robust methods, like implicit methods (plot from Hujeirat et al. 2007).*

- Continuity equation:

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

where ρ stands for the density and V is the velocity field. Using scaling variables (see Table 3.2), one may approximate the terms of this equation as follows:

$$\frac{\partial \rho}{\partial t} \sim \frac{\rho}{\tau} \quad \text{and} \quad \nabla(\rho V) \sim \frac{\rho\, V}{L}.$$

Scaling	variables	Molecular cloud	Accretion(onto SMBH)	Accretion (onto UCO)
$\tilde{L}$	Length	$O(pc)$	O(AU)	$O(10^{6,}$ cm)
$\tilde{\rho}$	Density	10^{-22} g cm^{-3}	10^{-6} g cm^{-3}	10^{-8} g cm^{-3}
$\tilde{\mathcal{T}}$	Temperature	10 K	10^{6} K	10^{7} K
$\tilde{V}$	Velocity	0.3 km s^{-1}	10^{2} km s^{-1}	10^{2-3} km s^{-1}
$\tilde{B}$	Magnetic Fields	30 μ G	10^{2} G	10^{4} G
$\tilde{\mathcal{M}}$	Mass	10^{3} M$_{\odot}$	10^{6} M$_{\odot}$	M$_{\odot}$
$\tilde{\dot{\mathcal{M}}}$	Accretion rate		10^{-2} M$_{\odot}$ Y^{-1}	10^{-10} M$_{\odot}$ Y^{-1}

Table 3.2: A list of possible scaling variables typical for three different astrophysical phenomena: giant molecular clouds, accretion onto super-massive black holes (SMBHs) and accretion onto ultra-compact objects (UCO). These scaling variables are used to determine the typical time scales involved in such accretion phenomena (Hujeirat et al. 2007).

This yields the hydrodynamical time scale

$$\tau_{HD} = \frac{L}{V}. \tag{3.2}$$

The so-called accretion time scale can be obtained by integrating the continuity equation over the whole fluid volume. Specifically,

$$\int_V \frac{\partial \rho}{\partial t}\, dV = \frac{\partial M}{\partial t} \sim \frac{M}{\tau}, \quad \int_V (\nabla \cdot \rho V)\, dV = \int_S \rho V \cdot n \cdot dS \sim \dot{M},$$

where 'V' denotes the total volume of the gas and 'S' corresponds to its surface.
Equating the latter two terms, one obtains: $\frac{M}{\tau} \sim \dot{M}$ which gives the accretion time scale:

$$\tau_{acc} \sim \frac{M}{\dot{M}}. \tag{3.3}$$

In general τ_{acc} is one of the longest time scales characterizing astrophysical flows connected to the accretion phenomena.

- The momentum equations:

$$\frac{\partial V}{\partial t} + \nabla V \otimes V = -\frac{1}{\rho}\nabla P + f_{cent} + \frac{f_{\rm rad}}{\rho} + \nabla\psi + \frac{\nabla \times B \times B}{4\pi\rho} + Q^{\rm r}_{\rm vis}, \tag{3.4}$$

where $P,\ f_{cent},\ f_{\rm rad},\ \psi,\ B,\ Q^{\rm r}_{\rm vis}$ denote gas pressure, centrifugal force, radiative force, gravitational potential, magnetic field and viscous operators, respectively. From this equation, we may obtain the following time scales:

1. The sound speed crossing time can be obtained by comparing the following two terms: $\frac{\partial V}{\partial t} \approx \frac{\nabla P}{\rho}$, which yields:

$$\tau_{\rm s} \approx \tau_{HD} \left(\frac{V}{V_s}\right)^2, \tag{3.5}$$

   where $V_s$is the sound speed.
2. The gravitational time scale from $\frac{\partial V}{\partial t} \approx \nabla\psi$:

$$\tau_G = \tau_{HD} \left(\frac{V}{V_g}\right)^2, \tag{3.6}$$

   where $V_g^2 = GM/L$ and G is the gravitational constant.
3. Similarly, the Alfv̀en-wave crossing-time from $\frac{\partial V}{\partial t} \approx \frac{\nabla \times B \times B}{4\pi\rho}$:

$$\tau_{mag} = \tau_{HD} \left(\frac{V}{V_A}\right)^2, \tag{3.7}$$

   where $V_A^2 (= B^2/4\pi\rho)$ denotes the Alfv̀en speed squared.
4. Radiative effects in moving flows propagate on the radiative scale, which is obtained from $\frac{\partial V}{\partial t} \approx \frac{f_{\rm rad}}{\rho}$:

$$\tau_{rad} = \tau_{HD} \left(\frac{V}{c}\right)^2, \tag{3.8}$$

   where c is the speed of light.

5. The viscous time scale from $\frac{\partial V}{\partial t} \approx Q^{\rm r}_{\rm vis} \sim \frac{\nu V}{L^2}$:

$$\tau_{vis} = \frac{L^2}{\nu} \tag{3.9}$$

where ν is a viscosity coefficient.

- The induction equation, taking into account the effects of $\alpha_{\rm dyn}-$dynamo, magnetic diffusivity $\nu_{\rm diff}$ and of ambipolar diffusion reads:

$$\frac{\partial B}{\partial t} = \nabla \times \langle V \times B + \alpha_{\rm dyn} B - \nu_{\rm mag} \nabla \times B \rangle + \nabla \times \{ \frac{B}{4\pi\gamma\rho_i\rho_n} \times [B \times (\nabla \times B)] \}, \tag{3.10}$$

where $\rho_{\rm i,n}$ denote the ion and neutral densities.

Thus, the induction equation contains several important time scales:

1. The dynamo amplification time scale, which results from the equality: $\frac{\partial B}{\partial t} = \nabla \times \alpha_{\rm dyn} B$ and gives:

$$\tau_{dyn} = \frac{L}{\alpha_{\rm dyn}} \tag{3.11}$$

2. The magnetic-diffusion time scale can be determined from $\frac{\partial B}{\partial t} = \nabla \times (\nu_{\rm mag} \nabla \times B)$:

$$\tau_{diff} = \frac{L^2}{\nu_{mag}} \tag{3.12}$$

3. The ambipolar diffusion time scale from

$$\frac{\partial B}{\partial t} = \nabla \times \{ \frac{B}{4\pi\gamma\rho_i\rho_n} \times [B \times (\nabla \times B)] \}$$

$$\Leftrightarrow \frac{B}{\tau} \sim \frac{1}{L} \left(\frac{B^2}{4\pi\rho_n} \right) \left(\frac{1}{\gamma\rho_i} \right) \left(\frac{B}{L} \right) \sim \frac{V_A^2}{\gamma\rho_i} \frac{B}{L^2} = \mathcal{D}_{amb} \frac{B}{L^2}$$

which gives:

$$\tau_{amb} = \frac{L^2}{\mathcal{D}_{amb}}, \tag{3.13}$$

where $\mathcal{D}_{amb} (= V_A^2/(\gamma\rho_i))$ is the ambipolar diffusion coefficient.

- The chemical reaction equations,
where the equation describing the chemical-evolution of species $'i'$ is:

$$\frac{\partial \rho_i}{\partial t} = \sum_m \sum_n k_{mn} \rho_m \rho_n + \sum_m I_m \rho_m, \tag{3.14}$$

where k_{mn} denotes the reaction rate between the species m and n. I_m stands for other external sources. For example, the reaction equation of atomic hydrogen in a primordial gas reads:

$$\frac{\partial \rho_H}{\partial t} = \frac{k_2}{m_H} \rho_{H^+} \rho_e - \frac{k_1}{m_H} \rho_H \rho_e \Leftrightarrow \frac{\rho_H}{\tau} \sim \frac{k_2}{m_H} \rho_H \, \rho_e$$

which gives a time scale of:

$$\frac{m_H}{k_2\,\rho_e}, \tag{3.15}$$

where ρ_e is the electron density and $k_2(= 10^{-10}\ \mathrm{cm}^3\ \mathrm{s}^{-1})$ correspond to the generation rate of atomic hydrogen through the capture of electrons by ionized atomic hydrogen. m_H denotes the mass of atomic hydrogen.

- Equations of relativistic MHD

  The velocities in relativistic flows are comparable to the speed of light. This implies that the hydrodynamical τ_{HD} and radiative τ_{rad} time scales are comparable and that both are much shorter than in Newtonian flows.

Time scales	Molecular cloud	Accretion(onto SMBH)	Accretion (onto UCO)
τ_{HD}	$\sim 10^6$ Yr	$\sim$ months	~ 1 s
τ_{rad}/τ_{HD}	$\sim 10^{-6}$	$\sim 10^{-3}$	$\sim 10^{-3}$
τ_{grav}/τ_{HD}	$\sim 10^{-2}$	$\sim 10^{-3}$	$\sim 10^{-3}$
τ_{ch}/τ_{HD}	$\sim 10^{-1}$	$\sim 10^{-5}$	$\sim 10^{-4}$
τ_{mag}/τ_{HD}	$\sim 10^{-2}$	$\sim 10^{0}$	$\sim 10^{-1}$
τ_{vis}/τ_{HD}	$\sim 10^{1}$	$\sim 10^{2}$	$\sim 10^{2}$
τ_{acc}/τ_{HD}		$\sim 10^{4}$	$\sim 10^{12}$

Table 3.3: A list of the time scales relative to the hydrodynamical time scale for three different astrophysical phenomena.

Why using implicit methods in General Relativity?

For velocities near the speed of light the time step has to be limited to get a physical consistent solution e.g. for the shock propagation.
In implicit methods the physical convergence determines the time step size:
large time steps give no numerical instabilities like in explicit methods, but may give physically wrong or too diffusive solutions if the time step size is too large.
So in time-dependent general relativistic simulations the time step should be of order dx/c due to accuracy reasons even for implicit methods.
But with implicit methods it is possible to simulate large Lorentz-factor flows (with Lorentz-factors between 100–500), like in Gamma Ray Bursts (GRBs), which usually cannot be reached by explicit methods.

Although the dynamical time scale in relativistically moving flows is relatively short and therefore a large time step cannot be used for physical reasons anyway, there are enough other reasons to use implicit numerical procedures in this case:

1. The (general-) relativistic (magneto-) hydrodynamic equations are strongly non-linear, giving rise to fast growing non-linear perturbations, imposing thereby a further restriction on the size of the time step (for explicit methods)
2. The extreme spacetime curvature in the vicinity of the black hole, results in other non-linear effects in fluid flows. Therefore, to accurately capture such flow structures, a non-linear distribution of the grid points is necessary, which may destabilize explicit schemes.

3. Initially non-relativistic flows may become ultra-relativistic or vice versa. Since almost all non-relativistic astrophysical flows known to date are considered to be dissipative and diffusive. To track their time-evolution reliably, it is necessary that the numerical solver is capable of treating the corresponding second order viscous terms properly.
4. The accumulated round off errors resulting from performing a large number of time-steps for time-advancing a numerical hydrodynamical solution, which may be necessary when using explicit methods, may easily cause divergence from the real physical solution.

3.2 Non-dimensional formulation (Scaling) of Equations

The numerical algorithm described here should be able to solve the time-evolution of hydrodynamical flows both in the non-relativistic as well as in the extreme-relativistic regime. Therefore the equations are scaled to non-dimensional units to ensure, that there are no problems with extremely small and extremely large numerical values and that the linear equation solver is well behaved, which is the case when the Jacobian matrix is always well diagonal dominant.

Instead of the usual convention to set the speed of light and the gravitational constant to unity, one can use the sound speed as the basic measure for velocities. This is reasonable as the radial motion of rotating flows around compact objects can be as low as about 10^{-5} the speed of light, whereas that is about 10^{-2} the sound speed. But close to the event horizon, all velocities become quantitatively comparable and are almost reaching the speed of light, so there scaling with the speed of light is also right.

With the definition of the conservative variables

$$\begin{aligned}
D &\doteq \rho\,\Gamma && [ML^{-3}] \\
m \doteq M_r &= \sqrt{g_{rr}}\,\overline{\overline{\mathrm{D}}}\,U && [ML^{-3}LT^{-1}] \\
n \doteq M_\theta &= \sqrt{g_{\theta\theta}}\,\overline{\overline{\mathrm{D}}}\,V && [ML^{-3}L^2T^{-1}] \\
l \doteq M_\phi &= \overline{\overline{\mathrm{D}}}\,g_{\phi\phi}\,d\omega && [ML^{-3}L^2T^{-1}] \\
\epsilon^d &\doteq \epsilon\,D && [ML^{-3}L^2T^{-2}]
\end{aligned} \tag{3.16}$$

and the primitive variables ρ, U, V, ω and P (given by the equation of state), where the velocities are defined by

$$\begin{aligned}
U &= \sqrt{g_{rr}}\,V^r && [LT^{-1}] \\
V &= \sqrt{g_{\theta\theta}}\,V^\theta && [LT^{-1}] \\
\omega = V^\phi &= \omega_{FDE} + d\omega && [T^{-1}]
\end{aligned} \tag{3.17}$$

one gets for the axi-symmetrical ($\partial\phi = 0$) general relativistic Navier-Stokes equations (see 2.105) in the so called flux-conservative form, that smoothly adapts to the Newtonian form in the non-relativistic regime:

$$\frac{\partial D}{\partial t} + \frac{1}{\sqrt{-g}}\frac{\partial}{\partial r}\left(\sqrt{\frac{-g}{g_{rr}}}\, U\, D\right) + \frac{1}{\sqrt{-g}}\frac{\partial}{\partial \theta}\left(\sqrt{\frac{-g}{g_{\theta\theta}}}\, V\, D\right) = 0$$

$$[ML^{-3}\, T^{-1}] = [\rho_{DIM}\, t_{DIM}^{-1}] \tag{3.18}$$

$$\frac{\partial m}{\partial t} + \frac{1}{\sqrt{-g}}\frac{\partial}{\partial r}\left(\sqrt{\frac{-g}{g_{rr}}}\, U\, m\right) + \frac{1}{\sqrt{-g}}\frac{\partial}{\partial \theta}\left(\sqrt{\frac{-g}{g_{\theta\theta}}}\, V\, m\right) = -\frac{\partial P}{\partial r} + \frac{c}{2}\left(\frac{M^\mu M^\nu}{M^t}\right)\frac{\partial g_{\mu\nu}}{\partial r} + Q^r_{vis}$$

$$[ML^{-3}\, LT^{-1}\, T^{-1}] = [\rho_{DIM}\, V_{DIM}\, t_{DIM}^{-1}] \tag{3.19}$$

$$\frac{\partial n}{\partial t} + \frac{1}{\sqrt{-g}}\frac{\partial}{\partial r}\left(\sqrt{\frac{-g}{g_{rr}}}\, U\, n\right) + \frac{1}{\sqrt{-g}}\frac{\partial}{\partial \theta}\left(\sqrt{\frac{-g}{g_{\theta\theta}}}\, V\, n\right) = -\frac{\partial P}{\partial \theta} + \frac{c}{2}\left(\frac{M^\mu M^\nu}{M^t}\right)\frac{\partial g_{\mu\nu}}{\partial \theta} + Q^\theta_{vis}$$

$$[ML^{-3}\, L\, LT^{-1}\, T^{-1}] = [\rho_{DIM}\, R_{DIM}\, V_{DIM}\, t_{DIM}^{-1}] \tag{3.20}$$

$$\frac{\partial l}{\partial t} + \frac{1}{\sqrt{-g}}\frac{\partial}{\partial r}\left(\sqrt{\frac{-g}{g_{rr}}}\, U\, l\right) + \frac{1}{\sqrt{-g}}\frac{\partial}{\partial \theta}\left(\sqrt{\frac{g}{g_{\theta\theta}}}\, V\, l\right) = Q^\phi_{vis}$$

$$[ML^{-3}\, L\, LT^{-1}\, T^{-1}] = [\rho_{DIM}\, R_{DIM}\, V_{DIM}\, t_{DIM}^{-1}] \tag{3.21}$$

$$\frac{\partial \epsilon^d}{\partial t} + \frac{1}{\sqrt{-g}}\frac{\partial}{\partial r}\left(\sqrt{\frac{-g}{g_{rr}}}\, U\, \epsilon^d\right) + \frac{1}{\sqrt{-g}}\frac{\partial}{\partial \theta}\left(\sqrt{\frac{-g}{g_{\theta\theta}}}\, V\, \epsilon^d\right) =$$
$$-P\left[\frac{\partial}{\partial t}\left(\frac{u^t}{c}\right) + \frac{1}{\sqrt{-g}}\frac{\partial}{\partial r}\left(\sqrt{\frac{-g}{g_{rr}}}\, U\, \frac{u^t}{c}\right) + \frac{1}{\sqrt{-g}}\frac{\partial}{\partial \theta}\left(\sqrt{\frac{-g}{g_{\theta\theta}}}\, V\, \frac{u^t}{c}\right)\right] + \Phi + \hat{\Gamma} - \Lambda,$$

$$[ML^{-3}\, L^2 T^{-2}\, T^{-1}] = [\rho_{DIM}\, V_{DIM}^2\, t_{DIM}^{-1}] \tag{3.22}$$

where $+M^\gamma M^\mu \partial_\nu(g_{\mu\gamma}) = -M_\mu M_\alpha \partial_\nu(g^{\alpha\mu})$ and with $t_{DIM} = R_{DIM} V_{DIM}^{-1}$.

The basic scaling variables are:

- the typical density ρ_{DIM},
- the typical length scale R_{DIM}, which is typically set to the inner radius,
- the typical velocity of the flow V_{DIM},
  typically set to the sound speed $V_{s,DIM}$ or to the speed of light c_{DIM}.
- the typical temperature T_{DIM}.

From these basic scaling variables one can determine derived scaling variables, e.g.:

- time scale $t_{DIM} = \frac{R_{DIM}}{V_{DIM}}$
- typical energy density and pressure scale $E_{DIM} = P_{DIM} = \rho_{DIM}\, V_{DIM}^2$

and for the scaling factors of the conservative variables one gets:

- $D_{DIM} = \rho_{DIM}$

- $m_{DIM} = \rho_{DIM}\, V_{DIM}$
- $n_{DIM} = \rho_{DIM}\, R_{DIM}\, V_{DIM}$
- $l_{DIM} = \rho_{DIM}\, R_{DIM}\, V_{DIM}$
- $\epsilon^d_{DIM} = \rho_{DIM}\, V^2_{DIM}$

But there is a slight problem:
The scaled conservative variables D, m, n, l, ϵ^d should be all of the same order, so that the resulting Jacobian Matrix for implicit methods is always well diagonal dominant, which is important to guaranty (good) convergence of the iterative linear equation solver.
But since usually there are two different velocity scales in the radial and azimuthal direction, where the velocities are of the order of the sound speed $V_{s;DIM} = \sqrt{\frac{\gamma\, P_{DIM}}{\rho_{DIM}\, \tilde{h}}}$, and in the ϕ-direction, where the accretion disc is rotating with the Keplerian velocity $V_{K;DIM} = \sqrt{\frac{G\, M_{BH}}{R_{DIM}}} = \sqrt{\frac{r_{g;DIM}}{R_{DIM}}}\, c_{DIM}$, which is usually much larger than the sound speed.
To avoid such problems, one can multiply each equation with a constant factor $f_{i_{eq}}$, where the index i_{eq} denotes the i-th equation of the system. To avoid the aforementioned problems with different velocity scales, one can set $f_D = f_m = f_n = f_{\epsilon^d} = 1$, so leave these equations as they are, and only multiply the angular momentum equation by f_l. To take into account the different typical velocity in the ϕ direction one can define a modified angular momentum:

$$\tilde{l} = f_l\, l \quad \text{with } f_l = \frac{V_{DIM}}{V_{K;DIM}} \tag{3.23}$$

which results in a modified angular velocity equation:

$$\frac{\partial \tilde{l}}{\partial t} + \frac{1}{\sqrt{-g}} \frac{\partial}{\partial r} \left(\sqrt{\frac{-g}{g_{rr}}}\, U\, \tilde{l} \right) + \frac{1}{\sqrt{-g}} \frac{\partial}{\partial \theta} \left(\sqrt{\frac{g}{g_{\theta\theta}}}\, V\, \tilde{l} \right) = f_l\, Q^{\phi}_{vis} \tag{3.24}$$

In that case the variable $l = \frac{\tilde{l}}{f_l}$ is calculated from $\tilde{l}$ every time $\tilde{l}$ is determined, to ensure that the right l is used in the other equations. This procedure ensures that only one equation has to be scaled differently and the other equations, e.g. the velocity normalization equation used to determine the Lorentz factor, have not to be modified.

Note that for general relativistic simulations where V_{DIM} is set to c_{DIM} one should set $f_l = 1$, so that all velocities are scaled with the speed of light, because in relativity all velocities are limited by the speed of light and in the proximity of the black hole all velocities will almost reach the speed of light.

Scaling from the dimensional variables to the non-dimensional variables is usually only done once at the input (or maybe restart) and the back transformation to dimensional variables may be done during data analysis and visualization.

3.3 Grid and Discretization

3.3.1 Grid Generation

With a few parameters either a uniform or a stretched grid can be generated. Giving the inner and outer radial and latitudinal limits of the computational domain, possibly some intermediate values and the cell numbers between these values, the minimal grid spacing in r and θ direction and where this should be located, a suitable grid is generated. See figures 3.3 and 3.4 for details.
Sometimes this can give errors if the parameters are not well selected, because an iterative routine is used to distribute the cells between the given points. In such a case one has to tweak the parameters to get a successful result. In the future it might be possible to replace this grid generation method with a more user friendly method like the one used in the PLUTO code (Mignone et al. 2007). It is also possible to define more than one grid level and then later at a specified iteration time step or a specified physical time prolongate from one grid level to the next level which - in general - possesses a finer grid. With that method it is possible to save computational time, by first calculating the solution on a coarse grid and then continue on a more refined grid.

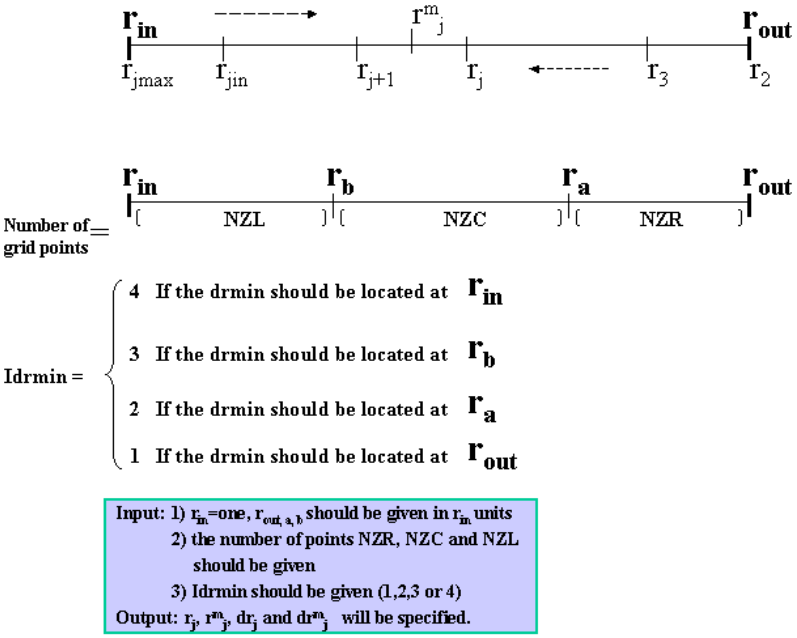


Figure 3.3: *Radial grid structure: jmax := J2G + 1, jin := J2G, J1G = 2. Note that the grid indices run from outward to inwards! r_{in} and r_{out} are the radial grid boundaries, NZL describes the number of grid cells between the inner boundary r_{in} and an intermediate radius r_b, NZC the number of grid cells between the inner radii r_b and r_a and NZR the number of grids cells between r_a and the outer boundary r_{out}. The parameter Idrmin describes where the minimal radial grid spacing dr_{min} should be located. Given all this parameters an iterative routine is used to generate the grid spacing (plot from Hujeirat for GR-I-RMHD).*

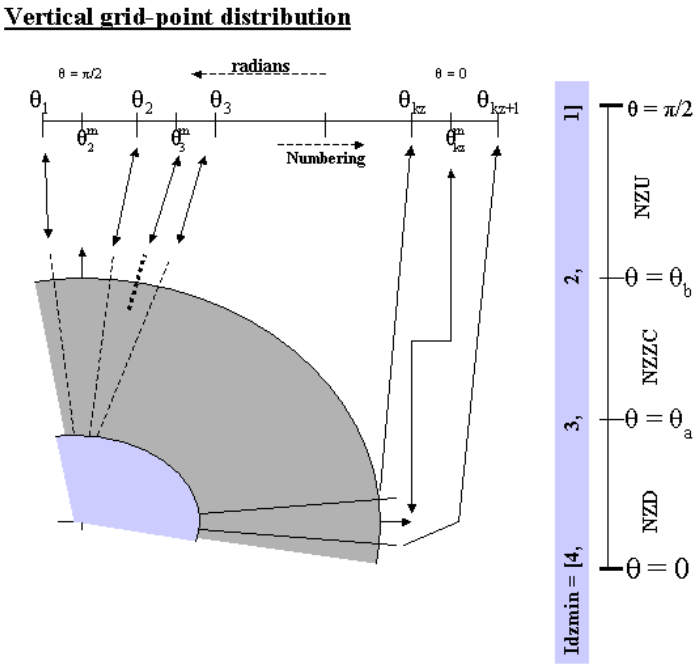


Figure 3.4: *Latitudinal/vertical grid structure: kz := K2G, K1G = 2. The latitudinal grid is setup similar to the radial grid in figure 3.3 (plot from Hujeirat for GR-I-RMHD).*

3.3.2 Staggered Grid and Grid Structure

For the discretization a staggered grid is used. The structure of the different staggered grid cells and the location of the variables can be seen in figures 3.6, 3.7 and 3.8.

For the axi-symmetric two dimensional grid there exist three different grid cells:

- the cells with centred variables D, ϵ^d
  and in the axi-symmetric case also for the vector components in ϕ-direction: ω and l,
- the cells shifted in the radial direction for the radial vector components U and m
- and the cells shifted in the θ direction for the azimuthal/latitudinal vector components V and n.

For a full 3 dimensional grid one would also have a further grid cell which would be shifted in ϕ-directions for the vector components in ϕ-direction: ω and l.

The staggered grid with these different cells is used so that in most cases there is no need to interpolate variables which therefore should give a more accurate result. E.g. in case of the continuity equation there is no need to interpolate the velocities from the cell centre of the D-cell to the cell boundaries.

3.3.3 Finite Volume Discretization

In the Finite Volume method a cell, a small volume, is looked at.

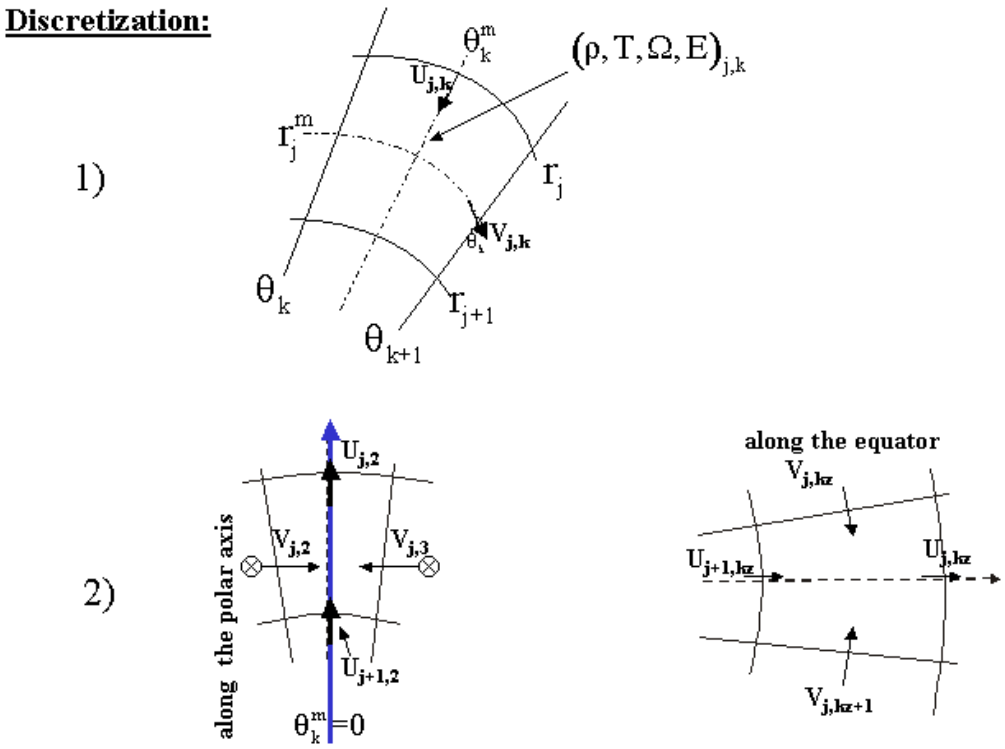


Figure 3.5: *Five star staggered grid discretization: 1) shows the location of the grid variables: density, temperature, angular momentum and forces are stored in the grid centre of the 'density'-cell, whereas the velocity components are stored at the cell interfaces. 2) shows the boundary cells at the polar axis and the midplane (equator) (plots from Hujeirat for GR-I-RMHD).*

The hydrodynamic equations can all be written in flux-conservative vector form:

$$\frac{\partial \vec{q}}{\partial t} + L_{r,rr}\vec{F} + L_{\theta,\theta\theta}\vec{G} = \vec{f}, \tag{3.25}$$

with the vector of conservative variables $\vec{q}$ and where $\vec{F}$ and $\vec{G}$ are the fluxes of q in r and θ direction, and $L_{r,rr}$, $L_{\theta,\theta\theta}$ are first and second order operators that describe the advection and diffusion of the vector variables $\vec{q}$ in r and θ directions. $\vec{f}$ corresponds to the vector of source functions.

For clarity let us consider now only a single equation with only first order operators, for example the continuity equation. This equation has the form:

$$\frac{\partial Q}{\partial t} + \mathrm{div}\left(\vec{F}\right) = S, \tag{3.26}$$

where Q is the conservative variable, $\vec{F}$ is the vector of fluxes and S is the source term, which is in case of the continuity equation zero.

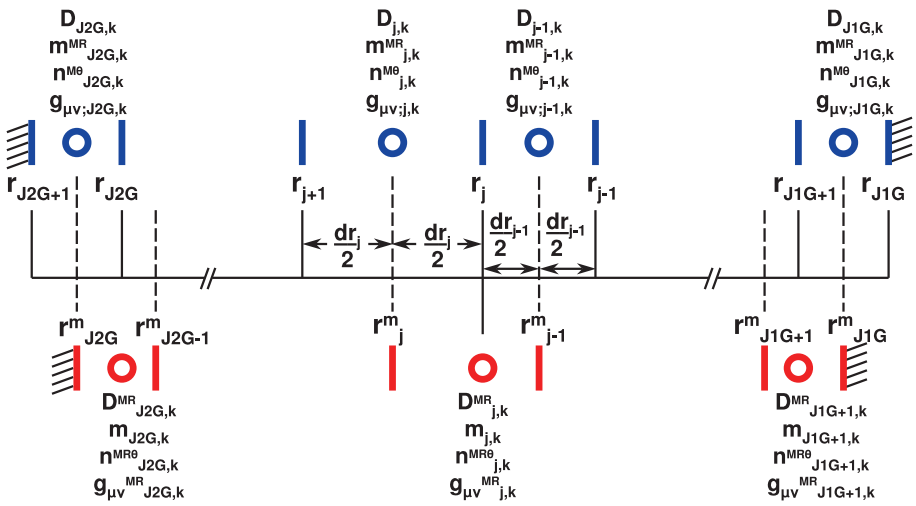


Figure 3.6: *Staggered grid structure in radial direction*

Due to Gauß' law one can transform the divergence of the fluxes integrated over a volume to a flux through the surface:

$$\int_{V_{cell}} \text{div}\left(\vec{F}\right) dV = \int_{S_{cell}} \vec{F} d\vec{S} \tag{3.27}$$

With this information one can interpret the equation in flux conservative form 3.26 in the following sense:
the change in time of the quantity q in a cell is equal to the net-flow of the fluxes through the cell boundaries.
In case of the continuity equation that means, that the change of mass density in a cell is equal to the net-flow of the mass fluxes through the cell boundaries.

An important property of this Finite Volume Discretization method is that it guaranties that the conserved variables are also numerically conserved (up to tiny numerical errors).

Therefore this method plays a crucial role if conservation is very important, e.g. one has to specify the angular momentum as a conserved variable and not a linear momentum in ϕ direction, elsewise the angular momentum is not conserved numerically.

Discretization of the continuity equation

The continuity equation is discretized using the staggered grid strategy within the context of finite volume philosophy.

The advection term in r-direction

$$\frac{1}{\sqrt{-g}} \frac{\partial}{\partial r} \left(\sqrt{\frac{-g}{g_{rr}}} U D \right) \Big|_{j,k}$$

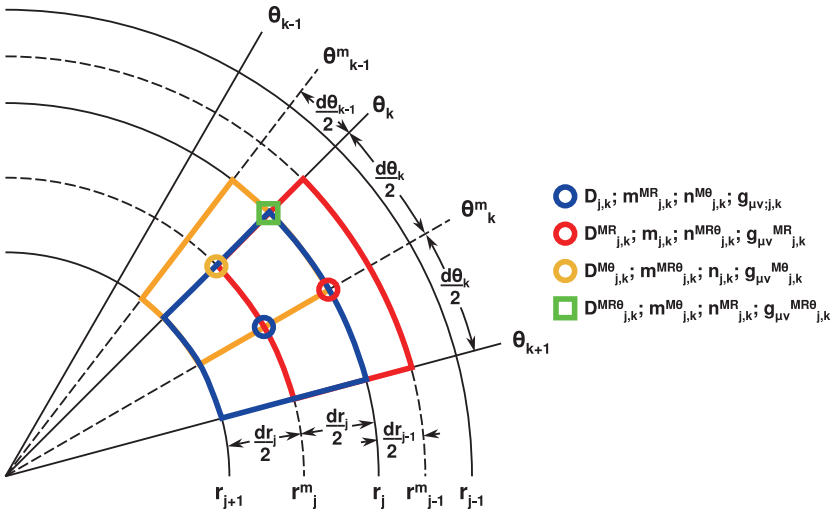


Figure 3.7: *Staggered grid with all three different cell types shown.*

using the integration over the r-interval $[r_{j+1}, r_j]; r_{j+1} < r_j$ (attention: the indices run from outwards to inwards!) gives:

$$\begin{aligned}
\left.\frac{[\sqrt{-g/g_{rr}}\,U\overrightarrow{D^r}]_{r_{j+1}}^{r_j}}{\int_{r_{j+1}}^{r_j}\sqrt{-g}dr}\right|_k &= \left.\frac{[\sqrt{-g/g_{rr}}\,U\overrightarrow{D^r}]_{r=r_j} - [\sqrt{-g/g_{rr}}\,U\overrightarrow{D^r}]_{r=r_{j+1}}}{\int_{r_{j+1}}^{r_j}\sqrt{-g}dr}\right|_k \\
&= \left.\frac{\left[\frac{\bar{\rho}^2\cos(\theta)}{\sqrt{\bar{\rho}^2/\Delta}}\,U\overrightarrow{D^r}\right]_{r=r_j} - \left[\frac{\bar{\rho}^2\cos(\theta)}{\sqrt{\bar{\rho}^2/\Delta}}\,U\overrightarrow{D^r}\right]_{r=r_{j+1}}}{\int_{r_{j+1}}^{r_j}\bar{\rho}^2\,\cos(\theta)\,dr}\right|_k \\
&= \left.\frac{\left[\bar{\rho}\sqrt{\Delta}\,U\overrightarrow{D^r}\right]_{r=r_j} - \left[\bar{\rho}\sqrt{\Delta}\,U\overrightarrow{D^r}\right]_{r=r_{j+1}}}{\int_{r_{j+1}}^{r_j}\bar{\rho}^2\,dr}\right|_k ,
\end{aligned} \tag{3.28}$$

where

$$\begin{aligned}
\left.\int_{r_{j+1}}^{r_j}\bar{\rho}^2\,dr\right|_k &= \int_{r_{j+1}}^{r_j}(r^2 + a^2\sin^2(\theta_k^m))\,dr = [\frac{r^3}{3} + a^2\sin^2(\theta_k^m)\,r]_{r_{j+1}}^{r_j} \\
&= \frac{r_j^3 - r_{j+1}^3}{3} + a^2\sin^2(\theta_k^m)\,(r_j - r_{j+1}).
\end{aligned} \tag{3.29}$$

The advection term in θ-direction

$$\left.\frac{1}{\sqrt{-g}}\frac{\partial}{\partial\theta}\left(\sqrt{\frac{-g}{g_{\theta\theta}}}\,V\,D\right)\right|_{j,k}$$

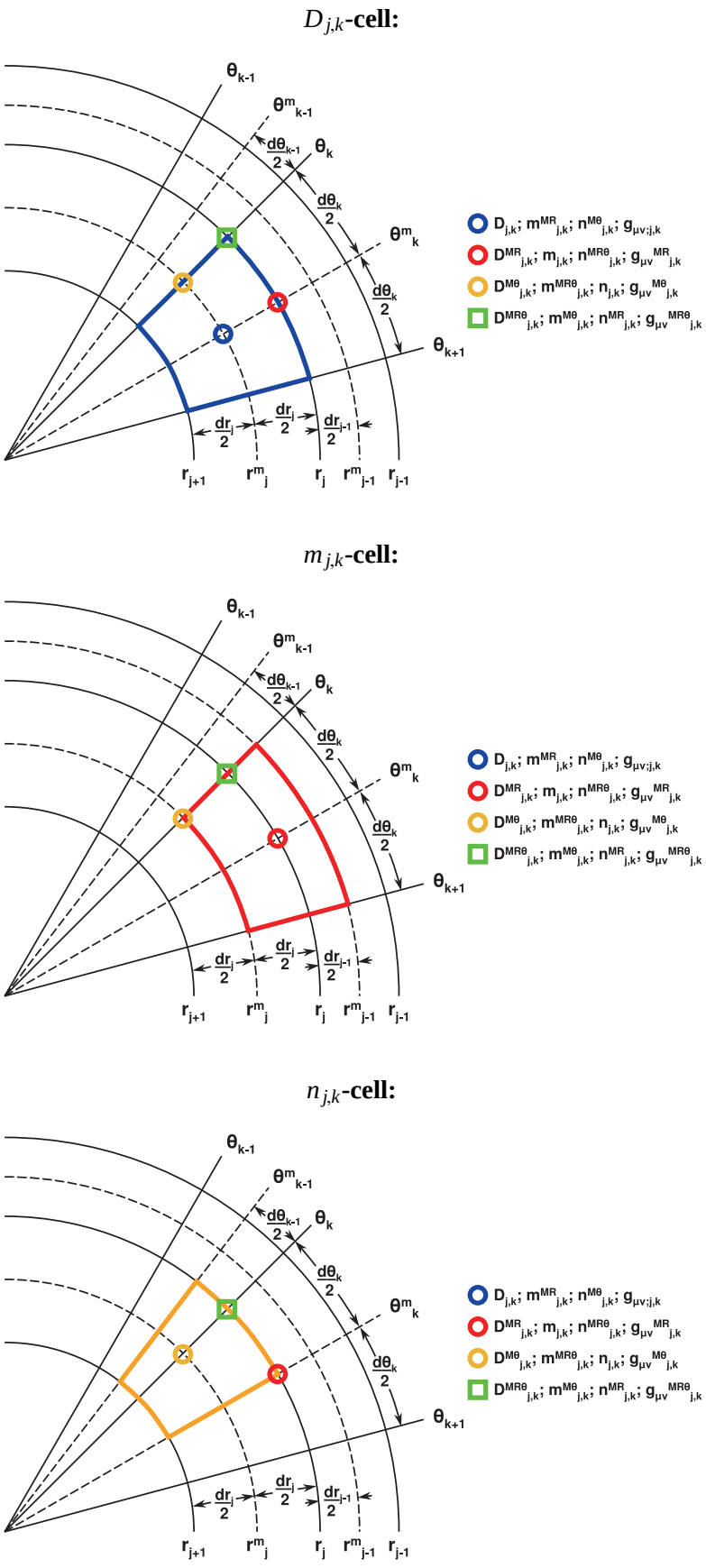


Figure 3.8: *Staggered grid structure: separate plot for each different cell type*

using the integration over the θ-interval $[\theta_{k+1}, \theta_k]; \theta_{k+1} < \theta_k$ (attention: the indices run from the axis to the midplane, whereas the latitude θ increases from the midplane to the pole) gives:

$$\left.\frac{[\sqrt{-g/g_{\theta\theta}}\,V\overrightarrow{D^\theta}]_{\theta_{k+1}}^{\theta_k}}{\int_{\theta_{k+1}}^{\theta_k}\sqrt{-g}d\theta}\right|_j = \left.\frac{[\sqrt{-g/g_{\theta\theta}}\,V\overrightarrow{D^\theta}]_{\theta=\theta_k} - [\sqrt{-g/g_{\theta\theta}}\,V\overrightarrow{D^\theta}]_{\theta=\theta_{k+1}}}{\int_{\theta_{k+1}}^{\theta_k}\sqrt{-g}d\theta}\right|_j$$

$$= \left.\frac{\left[\frac{\bar{\rho}^2\cos(\theta)}{\bar{\rho}}\,V\overrightarrow{D^\theta}\right]_{\theta=\theta_k} - \left[\frac{\bar{\rho}^2\cos(\theta)}{\bar{\rho}}\,V\overrightarrow{D^\theta}\right]_{\theta=\theta_{k+1}}}{\int_{\theta_{k+1}}^{\theta_k}\bar{\rho}^2\,\cos(\theta)\,d\theta}\right|_j$$

$$= \left.\frac{\left[\bar{\rho}\cos(\theta)\,V\overrightarrow{D^\theta}\right]_{\theta=\theta_k} - \left[\bar{\rho}\cos(\theta)\,V\overrightarrow{D^\theta}\right]_{\theta=\theta_{k+1}}}{\int_{\theta_{k+1}}^{\theta_k}\bar{\rho}^2\,\cos(\theta)\,d\theta}\right|_j \tag{3.30}$$

where

$$\left.\int_{\theta_{k+1}}^{\theta_k}\bar{\rho}^2\,\cos(\theta)\,d\theta\right|_j = \int_{\theta_{k+1}}^{\theta_k}\left(r_j^{m2} + a^2\sin^2(\theta)\right)\,\cos(\theta)\,d\theta = \left[r_j^{m2}\,\sin(\theta) + a^2\,\frac{\sin^3(\theta)}{3}\right]_{\theta_{k+1}}^{\theta_k}$$

$$= r_j^{m2}\,(\sin(\theta_k) - \sin(\theta_{k+1})) + a^2\,\frac{\left(\sin^3(\theta_k) - \sin^3(\theta_{k+1})\right)}{3}. \tag{3.31}$$

In the above expressions the following upwind values were used:

$$\overrightarrow{D^r}_{j,k} = \begin{cases} D_{j-1,k} + f^r_{j-1,k} & \text{if } U_{j,k} < 0 \\ D_{j,k} + f^r_{j,k} & \text{if } U_{j,k} \geq 0 \end{cases} ; \overrightarrow{D^\theta}_{j,k} = \begin{cases} D_{j,k-1} + f^\theta_{j,k-1} & \text{if } V_{j,k} < 0 \\ D_{j,k} + f^\theta_{j,k} & \text{if } V_{j,k} \geq 0 \end{cases} \tag{3.32}$$

The functions f^r and f^θ are corrections for maintaining higher order spatial accuracies.

Discretization of the radial momentum equation

For the discretization of the radial momentum equation the $m_{j,k}$-cell in figure 3.8 is used. On the staggered grid the radial momentum m is located at (r_j, θ_k^m) and defined as following: $m = \sqrt{g_{rr}^{MR}}\,\overline{\overline{\mathrm{D}}}^{MR}\,U$. and the radial momentum equation is (see 3.22):

$$\frac{\partial m}{\partial t} + \frac{1}{\sqrt{-g}}\frac{\partial}{\partial r}\left(\sqrt{\frac{-g}{g_{rr}}}\,U\,m\right) + \frac{1}{\sqrt{-g}}\frac{\partial}{\partial\theta}\left(\sqrt{\frac{-g}{g_{\theta\theta}}}\,V\,m\right) = -\frac{\partial P}{\partial r} + \frac{c}{2}\left(\frac{M^\mu M^\nu}{M^t}\right)\frac{\partial g_{\mu\nu}}{\partial r} + Q^r_{vis} \tag{3.33}$$

where $+M^\gamma M^\mu \partial_\nu(g_{\mu\gamma}) = -M_\mu M_\alpha \partial_\nu(g^{\alpha\mu})$

The advection term in r-direction

$$\left.\frac{1}{\sqrt{-g}}\frac{\partial}{\partial r}\left(\sqrt{\frac{-g}{g_{rr}}}\,U\,m\right)\right|_{j,k}$$

using the integration over the r-interval $[r_j^m, r_{j-1}^m]; r_j^m < r_{j-1}^m$ (attention: the indices run from outwards to inwards!) gives:

$$\begin{aligned}
\left.\frac{[\sqrt{-g/g_{rr}}\,U\overrightarrow{m^r}]_{r_j^m}^{r_{j-1}^m}}{\int_{r_j^m}^{r_{j-1}^m}\sqrt{-g}dr}\right|_k &= \left.\frac{[\sqrt{-g/g_{rr}}\,U\overrightarrow{m^r}]_{r=r_{j-1}^m} - [\sqrt{-g/g_{rr}}\,U\overrightarrow{m^r}]_{r=r_j^m}}{\int_{r_j^m}^{r_{j-1}^m}\sqrt{-g}dr}\right|_k \\
&= \left.\frac{\left[\frac{\bar{\rho}^2\cos(\theta)}{\sqrt{\bar{\rho}^2/\Delta}}\,U\overrightarrow{m^r}\right]_{r=r_{j-1}^m} - \left[\frac{\bar{\rho}^2\cos(\theta)}{\sqrt{\bar{\rho}^2/\Delta}}\,U\overrightarrow{m^r}\right]_{r=r_j^m}}{\int_{r_j^m}^{r_{j-1}^m}\bar{\rho}^2\,\cos(\theta)\,dr}\right|_k \\
&= \left.\frac{\left[\bar{\rho}\sqrt{\Delta}\,U\overrightarrow{m^r}\right]_{r=r_{j-1}^m} - \left[\bar{\rho}\sqrt{\Delta}\,U\overrightarrow{m^r}\right]_{r=r_j^m}}{\int_{r_j^m}^{r_{j-1}^m}\bar{\rho}^2\,dr}\right|_k, \qquad (3.34)
\end{aligned}$$

where $U_{j,k}^{MR} \doteq U(r_j^m, \theta_k^m) = \frac{U_{j+1,k}+U_{j,k}}{2}$ and

$$\begin{aligned}
\left.\int_{r_j^m}^{r_{j-1}^m}\bar{\rho}^2\,dr\right|_k &= \int_{r_j^m}^{r_{j-1}^m}(r^2 + a^2\sin^2(\theta_k^m))\,dr = [\frac{r^3}{3} + a^2\sin^2(\theta_k^m)\,r]_{r_j^m}^{r_{j-1}^m} \\
&= \frac{{r_{j-1}^m}^3 - {r_j^m}^3}{3} + a^2\sin^2(\theta_k^m)\,(r_{j-1}^m - r_j^m). \qquad (3.35)
\end{aligned}$$

The advection term in θ-direction

$$\left.\frac{1}{\sqrt{-g}}\frac{\partial}{\partial\theta}\left(\sqrt{\frac{-g}{g_{\theta\theta}}}\,V\,m\right)\right|_{j,k}$$

using the integration over the θ-interval $[\theta_{k+1}, \theta_k]; \theta_{k+1} < \theta_k$ (attention: the indices run from the axis to the midplane, whereas the latitude θ increases from the midplane to the pole) gives similar to the advection term in θ-direction of the continuity equation:

$$\begin{aligned}
\left.\frac{[\sqrt{-g/g_{\theta\theta}}\,V\overrightarrow{m^\theta}]_{\theta_{k+1}}^{\theta_k}}{\int_{\theta_{k+1}}^{\theta_k}\sqrt{-g}d\theta}\right|_j &= \left.\frac{[\sqrt{-g/g_{\theta\theta}}\,V\overrightarrow{m^\theta}]_{\theta=\theta_k} - [\sqrt{-g/g_{\theta\theta}}\,V\overrightarrow{m^\theta}]_{\theta=\theta_{k+1}}}{\int_{\theta_{k+1}}^{\theta_k}\sqrt{-g}d\theta}\right|_j \\
&= \left.\frac{\left[\frac{\bar{\rho}^2\cos(\theta)}{\bar{\rho}}\,V\overrightarrow{m^\theta}\right]_{\theta=\theta_k} - \left[\frac{\bar{\rho}^2\cos(\theta)}{\bar{\rho}}\,V\overrightarrow{m^\theta}\right]_{\theta=\theta_{k+1}}}{\int_{\theta_{k+1}}^{\theta_k}\bar{\rho}^2\,\cos(\theta)\,d\theta}\right|_j \\
&= \left.\frac{\left[\bar{\rho}\cos(\theta)\,V\overrightarrow{m^\theta}\right]_{\theta=\theta_k} - \left[\bar{\rho}\cos(\theta)\,V\overrightarrow{m^\theta}\right]_{\theta=\theta_{k+1}}}{\int_{\theta_{k+1}}^{\theta_k}\bar{\rho}^2\,\cos(\theta)\,d\theta}\right|_j \qquad (3.36)
\end{aligned}$$

where $V^{MR}_{j,k} \doteq V(r_j,\theta_k) = V_{j,k} + \frac{dr_j}{2}\frac{(Vj-1,k-V_{j,k})}{dr^m_j}$ with $V_{j,k} = V(r^m_j,\theta_k)$ and $dr^m_j = \left(\frac{dr_{j-1}}{2} + \frac{dr_j}{2}\right)$ and

$$\int_{\theta_{k+1}}^{\theta_k} \bar{\rho}^2\ \cos(\theta)\, d\theta\Big|_j = \int_{\theta_{k+1}}^{\theta_k} \left(r_j^2 + a^2\sin^2(\theta)\right)\ \cos(\theta)\, d\theta = \left[r_j^2\ \sin(\theta) + a^2\ \frac{\sin^3(\theta)}{3}\right]_{\theta_{k+1}}^{\theta_k}$$
$$= r_j^2\,(\sin(\theta_k) - \sin(\theta_{k+1})) + a^2\,\frac{\left(\sin^3(\theta_k) - \sin^3(\theta_{k+1})\right)}{3}. \tag{3.37}$$

In the above expressions the following upwind values were used:

$$\overrightarrow{m^r}_{j,k} = \begin{cases} m_{j,k} + f^{m;r}_{j,k} & \text{if } U^{MR}_{j,k} < 0 \\ m_{j+1,k} + f^{m;r}_{j+1,k} & \text{if } U^{MR}_{j,k} \geq 0 \end{cases};\ \overrightarrow{m^\theta}_{j,k} = \begin{cases} m_{j,k-1} + f^{\theta}_{j,k-1} & \text{if } V^{MR}_{j,k} < 0 \\ m_{j,k} + f^{\theta}_{j,k} & \text{if } V^{MR}_{j,k} \geq 0 \end{cases} \tag{3.38}$$

The functions $f^{m;r}$ and f^θ are corrections for maintaining higher order spatial accuracies.

The discretization of the source terms give:

For the radial pressure gradient $-\frac{\partial P}{\partial r}$ one gets:

$$-\frac{P_{j-1} - P_j}{dr^m_j} \tag{3.39}$$

The geometrical source terms including gravity are:

$$+\frac{c}{2}\left(\frac{M^\mu M^\nu}{M^t}\right)\frac{\partial g_{\mu\nu}}{\partial r} = -\frac{c}{2}\left(\frac{M_\mu M_\nu}{M^t}\right)\frac{\partial g^{\mu\nu}}{\partial r} \tag{3.40}$$

where $+M^\gamma M^\mu \partial_\nu(g_{\mu\gamma}) = -M_\mu M_\alpha \partial_\nu(g^{\alpha\mu})$

Written out this gives:

$$+\frac{c}{2}\left(\frac{M^\mu M^\nu}{M^t}\right)\frac{\partial g_{\mu\nu}}{\partial r} = \frac{c}{2M^t}\,M^\mu M^\nu \frac{\partial g_{\mu\nu}}{\partial r}$$
$$= \frac{1}{2M^t}\left(M^t M^t \frac{\partial g_{tt}}{\partial r} + 2\,M^t M^\phi \frac{\partial g_{t\phi}}{\partial r} + M^r M^r \frac{\partial g_{rr}}{\partial r} + M^\theta M^\theta \frac{\partial g_{\theta\theta}}{\partial r} + M^\phi M^\phi \frac{\partial g_{\phi\phi}}{\partial r}\right)$$
$$= \frac{M^t}{2}\frac{c\,\partial g_{tt}}{\partial r} + M^\phi\, c\,\frac{\partial g_{t\phi}}{\partial r} + \frac{M^{r2}}{2M^t}\frac{c\,\partial g_{rr}}{\partial r} + \frac{M^{\theta 2}}{2M^t}\frac{c\,\partial g_{\theta\theta}}{\partial r} + \frac{M^{\phi 2}}{2M^t}\frac{c\,\partial g_{\phi\phi}}{\partial r} \tag{3.41}$$

where $M^\alpha = \overline{\mathrm{D}}u^\alpha = \overline{\overline{\mathrm{D}}}V^\alpha$ with $V^t = c$, $V^r = \frac{U}{\sqrt{g_{rr}}}$, $V^\theta = \frac{V}{\sqrt{g_{\theta\theta}}}$, $V^\phi = \omega$ and $\overline{\overline{\mathrm{D}}} = \overline{\mathrm{D}}\Gamma = D\tilde{h}\Gamma$.
All variables are located where $m_{j,k}$ is located, at the point (r_j,θ^m_k).

The gravity term (Newtonian: $-\rho G M_{BH}/r^2$) is:

$$\frac{M^{tMR}\,c}{2}\,\frac{\partial g^{MR}_{tt}}{\partial r} = \frac{\overline{\overline{\mathrm{D}}}^{MR}c^2}{2}\,\frac{\partial g^{MR}_{tt}}{\partial r} \tag{3.42}$$

$$\overset{a=0}{=}\ \frac{\overline{\overline{\mathrm{D}}}^{MR}c^2}{2}\,\frac{-2r_g}{r^2} = \frac{\overline{\overline{\mathrm{D}}}^{MR}c^2}{2}\,\frac{-2GM_{BH}}{c^2r^2} = \overline{\overline{\mathrm{D}}}^{MR}\,\frac{-GM_{BH}}{r^2} \tag{3.43}$$

The frame dragging dependent term is (Newtonian and Schwarzschild: 0):

$$M^{\phi MR}\, c\, \frac{\partial g_{t\phi}^{MR}}{\partial r} = \overline{\overline{\mathrm{D}}}^{MR}\, c\, V^{\phi MR}\, \frac{\partial g_{t\phi}^{MR}}{\partial r} = \overline{\overline{\mathrm{D}}}^{MR}\, c\, \omega^{MR}\, \frac{\partial g_{t\phi}^{MR}}{\partial r} \tag{3.44}$$

$$\overset{a=0}{=} 0 \tag{3.45}$$

The pseudo 'centripetal' force due to the motion in the radial direction in general relativity (GR) (Newtonian: 0)
using

$$M^{r2} = (\overline{\overline{\mathrm{D}}}^{MR} V^{r})^2 = \frac{(\overline{\overline{\mathrm{D}}}^{MR} U)^2}{g_{rr}^{MR}} = \frac{(\sqrt{g_{rr}^{MR}}\overline{\overline{\mathrm{D}}}^{MR} U)^2}{g_{rr}^{MR2}} = \frac{m^2}{g_{rr}^{MR2}} \tag{3.46}$$

gives:

$$\frac{M^{r2}\, c}{2M^{tMR}}\, \frac{\partial g_{rr}^{MR}}{\partial r} = \frac{m^2\, c}{2\overline{\overline{\mathrm{D}}}^{MR}\, c\, g_{rr}^{MR2}}\, \frac{\partial g_{rr}^{MR}}{\partial r} = \frac{m^2}{2\overline{\overline{\mathrm{D}}}^{MR}\, g_{rr}^{MR2}}\, \frac{\partial g_{rr}^{MR}}{\partial r} \tag{3.47}$$

$$\overset{a=0}{=} \frac{m^2}{2\overline{\overline{\mathrm{D}}}^{MR}\, g_{rr}^{MR2}}\, \frac{-2r_g}{r^2} g_{rr}^{MR2} = \frac{m^2}{\overline{\overline{\mathrm{D}}}^{MR}}\, \frac{-r_g}{r^2} = \frac{m^2}{\overline{\overline{\mathrm{D}}}^{MR}}\, \frac{-GM_{BH}}{c^2 r^2}$$

$$= \frac{g_{rr}^{MR}\overline{\overline{\mathrm{D}}}^{MR2} U^2}{\overline{\overline{\mathrm{D}}}^{MR}}\, \frac{-r_g}{r^2} = \overline{\overline{\mathrm{D}}}^{MR} U^2\, g_{rr}^{MR}\, \frac{-r_g}{r^2}$$

$$= \overline{\overline{\mathrm{D}}}^{MR} U^2\, \frac{1}{\left(1 - \frac{2r_g}{r}\right)}\, \frac{-r_g}{r^2} = \overline{\overline{\mathrm{D}}}^{MR} U^2\, \frac{1}{\left(1 - \frac{2r_g}{r}\right)}\, \frac{-GM_{BH}}{c^2 r^2} \tag{3.48}$$

The pseudo centrifugal force due to the motion in theta direction for GR as well as in Newtonian (Newtonian: $\rho\frac{v_\theta^2}{r}$)
using

$$M^{\theta MR\theta 2} = (\overline{\overline{\mathrm{D}}}^{MR} V^{\theta MR\theta})^2 = \frac{(\overline{\overline{\mathrm{D}}}^{MR} V^{MR\theta})^2}{g_{\theta\theta}^{MR}} \tag{3.49}$$

gives:

$$\frac{M^{\theta MR\theta 2}\, c}{2M^{tMR}}\, \frac{\partial g_{\theta\theta}^{MR}}{\partial r} = \frac{(\overline{\overline{\mathrm{D}}}^{MR} V^{MR\theta})^2\, c}{2\overline{\overline{\mathrm{D}}}^{MR}\, c}\, \frac{1}{g_{\theta\theta}^{MR}}\, \frac{\partial g_{\theta\theta}^{MR}}{\partial r} = \overline{\overline{\mathrm{D}}}^{MR} \frac{V^{MR\theta 2}\, c}{2c}\, \frac{1}{g_{\theta\theta}^{MR}}\, \frac{\partial g_{\theta\theta}^{MR}}{\partial r} \tag{3.50}$$

$$\overset{a=0}{=} \overline{\overline{\mathrm{D}}}^{MR}\, \frac{V^{MR\theta 2}}{2}\, \frac{1}{r_j^2} \frac{\partial r^2}{\partial r} = \overline{\overline{\mathrm{D}}}^{MR}\, \frac{V^{MR\theta 2}}{2}\, \frac{2}{r_j} = \overline{\overline{\mathrm{D}}}^{MR}\, \frac{V^{MR\theta 2}}{r_j} \tag{3.51}$$

The pseudo centrifugal force term (Newtonian: $\rho\frac{v_\phi^2}{r} = \omega^2\, r\, cos^2(\theta)$ using $v_\phi = \omega\, r_{cyl} = \omega\, r\, cos(\theta)$):

$$\begin{aligned}\frac{M^{\phi MR}\, c^2}{2M^{tMR}}\,\frac{\partial g^{MR}_{\phi\phi}}{\partial r} &= \frac{(\overline{\overline{\mathrm{D}}}^{MR} V^{\phi MR})^2\, c}{2\overline{\overline{\mathrm{D}}}^{MR} c}\,\frac{\partial g^{MR}_{\phi\phi}}{\partial r} = \overline{\overline{\mathrm{D}}}^{MR}\,\frac{\omega^{MR2}}{2}\,\frac{\partial g^{MR}_{\phi\phi}}{\partial r} \qquad (3.52)\\ &\overset{a=0}{=} \overline{\overline{\mathrm{D}}}^{MR}\,\frac{\omega^{MR2}}{2}\,\frac{\partial r^2 cos^2(\theta^m_k)}{\partial r} = \overline{\overline{\mathrm{D}}}^{MR}\,\frac{\omega^{MR2}}{2}\,2\, r_j\, cos^2(\theta^m_k)\\ &= \overline{\overline{\mathrm{D}}}^{MR}\,\omega^{MR2}\, r_j\, cos^2(\theta^m_k) \qquad (3.53)\end{aligned}$$

The discretization of the viscous source terms Q^r_{vis} is not presented here.

Discretization of the latitudinal/vertical momentum equation

For the discretization of the latitudinal/vertical momentum equation the $n_{j,k}$-cell in figure 3.8 is used. On the staggered grid the latitudinal momentum n is located at (r^m_j, θ_k) and defined as following: $n = \sqrt{g^{M\theta}_{\theta\theta}}\,\overline{\overline{\mathrm{D}}}^{M\theta}\, V.$ and the latitudinal momentum equation is (see 3.22):

$$\frac{\partial n}{\partial t} + \frac{1}{\sqrt{-g}}\frac{\partial}{\partial r}\left(\sqrt{\frac{-g}{g_{rr}}}\, U\, n\right) + \frac{1}{\sqrt{-g}}\frac{\partial}{\partial \theta}\left(\sqrt{\frac{-g}{g_{\theta\theta}}}\, V\, n\right) = -\frac{\partial P}{\partial \theta} + \frac{c}{2}\left(\frac{M^\mu M^\nu}{M^t}\right)\frac{\partial g_{\mu\nu}}{\partial \theta} + Q^\theta_{vis} \quad (3.54)$$

where $+M^\gamma M^\mu \partial_\nu(g_{\mu\gamma}) = -M_\mu M_\alpha \partial_\nu(g^{\alpha\mu})$

The advection term in r-direction

$$\frac{1}{\sqrt{-g}}\frac{\partial}{\partial r}\left(\sqrt{\frac{-g}{g_{rr}}}\, U\, n\right)\Bigg|_{j,k}$$

using the integration over the r-interval $[r_{j+1}, r_j]; r_{j+1} < r_j$ (attention: the indices run from outwards to inwards!) gives similar to the advection term in r-direction of the continuity equation:

$$\begin{aligned}\frac{[\sqrt{-g/g_{rr}}\, U\vec{n}^{\vec{r}}]^{r_j}_{r_{j+1}}}{\int^{r_j}_{r_{j+1}}\sqrt{-g}dr}\Bigg|_k &= \frac{[\sqrt{-g/g_{rr}}\, U\vec{n}^{\vec{r}}]_{r=r_j} - [\sqrt{-g/g_{rr}}\, U\vec{n}^{\vec{r}}]_{r=r_{j+1}}}{\int^{r_j}_{r_{j+1}}\sqrt{-g}dr}\Bigg|_k\\ &= \frac{\left[\frac{\bar\rho^2\cos(\theta)}{\sqrt{\bar\rho^2/\Delta}}\, U\vec{n}^{\vec{r}}\right]_{r=r_j} - \left[\frac{\bar\rho^2\cos(\theta)}{\sqrt{\bar\rho^2/\Delta}}\, U\vec{n}^{\vec{r}}\right]_{r=r_{j+1}}}{\int^{r_j}_{r_{j+1}}\bar\rho^2\,\cos(\theta)\, dr}\Bigg|_k\\ &= \frac{\left[\bar\rho\sqrt{\Delta}\, U\vec{n}^{\vec{r}}\right]_{r=r_j} - \left[\bar\rho\sqrt{\Delta}\, U\vec{n}^{\vec{r}}\right]_{r=r_{j+1}}}{\int^{r_j}_{r_{j+1}}\bar\rho^2\, dr}\Bigg|_k, \qquad (3.55)\end{aligned}$$

where $U^{M\theta}_{j,k} \doteq U(r_j,\theta_k) = U_{j,k} + \frac{dr_j}{2}\frac{(U_{j,k-1}-U_{j,k})}{d\theta^m_k}$ with $U_{j,k} = U(r_j,\theta^m_k)$ and $d\theta^m_k = \left(\frac{d\theta_{k-1}}{2} + \frac{d\theta_k}{2}\right)$ and

$$\begin{aligned}\int_{r_{j+1}}^{r_j} \bar{\rho}^2\, dr\Big|_k &= \int_{r_{j+1}}^{r_j} (r^2 + a^2\sin^2(\theta_k)\, dr = [\frac{r^3}{3} + a^2\sin^2(\theta_k)\, r]_{r_{j+1}}^{r_j} \\ &= \frac{r_j^3 - r_{j+1}^3}{3} + a^2\sin^2(\theta_k)\,(r_j - r_{j+1}).\end{aligned} \tag{3.56}$$

The advection term in θ-direction

$$\frac{1}{\sqrt{-g}}\frac{\partial}{\partial\theta}\left(\sqrt{\frac{-g}{g_{\theta\theta}}}\, V\, n\right)\Big|_{j,k}$$

using the integration over the θ-interval $[\theta^m_k, \theta^m_{k-1}]; \theta^m_k < \theta^m_{k-1}$ (attention: the indices run from the axis to the midplane, whereas the latitude θ increases from the midplane to the pole) gives:

$$\begin{aligned}\frac{[\sqrt{-g/g_{\theta\theta}}\, V\overrightarrow{n^\theta}]_{\theta^m_k}^{\theta^m_{k-1}}}{\int_{\theta^m_k}^{\theta^m_{k-1}} \sqrt{-g} d\theta}\Bigg|_j &= \frac{[\sqrt{-g/g_{\theta\theta}}\, V\overrightarrow{n^\theta}]_{\theta=\theta^m_{k-1}} - [\sqrt{-g/g_{\theta\theta}}\, V\overrightarrow{n^\theta}]_{\theta=\theta^m_k}}{\int_{\theta^m_k}^{\theta^m_{k-1}} \sqrt{-g} d\theta}\Bigg|_j \\ &= \frac{\left[\frac{\bar{\rho}^2\cos(\theta)}{\bar{\rho}}\, V\overrightarrow{n^\theta}\right]_{\theta=\theta^m_{k-1}} - \left[\frac{\bar{\rho}^2\cos(\theta)}{\bar{\rho}}\, V\overrightarrow{n^\theta}\right]_{\theta=\theta^m_k}}{\int_{\theta^m_k}^{\theta^m_{k-1}} \bar{\rho}^2\, \cos(\theta)\, d\theta}\Bigg|_j \\ &= \frac{\left[\bar{\rho}\cos(\theta)\, V\overrightarrow{n^\theta}\right]_{\theta=\theta^m_{k-1}} - \left[\bar{\rho}\cos(\theta)\, V\overrightarrow{n^\theta}\right]_{\theta=\theta^m_k}}{\int_{\theta^m_k}^{\theta^m_{k-1}} \bar{\rho}^2\, \cos(\theta)\, d\theta}\Bigg|_j\end{aligned} \tag{3.57}$$

where $V^{M\theta}_{j,k} \doteq V(r^m_j,\theta^m_k) = \frac{V_{j,k}+V_{j,k+1}}{2}$ with $V_{j,k} = V(r^m_j,\theta_k)$ and

$$\begin{aligned}\int_{\theta^m_k}^{\theta^m_{k-1}} \bar{\rho}^2\, \cos(\theta)\, d\theta\Big|_j &= \int_{\theta^m_k}^{\theta^m_{k-1}} \left(r_j^{m2} + a^2\sin^2(\theta)\right)\, \cos(\theta)\, d\theta = \left[r_j^{m2}\, \sin(\theta) + a^2\, \frac{\sin^3(\theta)}{3}\right]_{\theta^m_k}^{\theta^m_{k-1}} \\ &= r_j^{m2}\left(\sin(\theta^m_{k-1}) - \sin(\theta^m_k)\right) + a^2\, \frac{\left(\sin^3(\theta^m_{k-1}) - \sin^3(\theta^m_k)\right)}{3}.\end{aligned} \tag{3.58}$$

In the above expressions the following upwind values were used:

$$\overrightarrow{n^r}_{j,k} = \begin{cases} n_{j,k} + f^r_{j,k} & \text{if } U^{M\theta}_{j,k} < 0 \\ n_{j+1,k} + f^r_{j+1,k} & \text{if } U^{M\theta}_{j,k} \geq 0 \end{cases}; \overrightarrow{n^\theta}_{j,k} = \begin{cases} n_{j,k-1} + f^{n;\theta}_{j,k-1} & \text{if } V^{M\theta}_{j,k} < 0 \\ n_{j,k} + f^{n;\theta}_{j,k} & \text{if } V^{M\theta}_{j,k} \geq 0 \end{cases} \tag{3.59}$$

The functions f^r and $f^{n;\theta}$ are corrections for maintaining higher order spatial accuracies.

The discretization of the source terms give:

For the latitudinal pressure gradient $-\frac{\partial P}{\partial \theta}$ one gets:

$$-\frac{P_{k-1}-P_k}{d\theta_k^m} \tag{3.60}$$

The geometrical source terms are:

$$+\frac{c}{2}\left(\frac{M^\mu M^\nu}{M^t}\right)\frac{\partial g_{\mu\nu}}{\partial\theta} = -\frac{c}{2}\left(\frac{M_\mu M_\nu}{M^t}\right)\frac{\partial g^{\mu\nu}}{\partial\theta} \tag{3.61}$$

where $+M^\gamma M^\mu \partial_\nu(g_{\mu\gamma}) = -M_\mu M_\alpha \partial_\nu(g^{\alpha\mu})$

Written out this gives:

$$\begin{aligned}
+\frac{c}{2}\left(\frac{M^\mu M^\nu}{M^t}\right)\frac{\partial g_{\mu\nu}}{\partial\theta} &= \frac{c}{2M^t}M^\mu M^\nu\frac{\partial g_{\mu\nu}}{\partial\theta}\\
&= \frac{1}{2M^t}\left(M^tM^t\frac{\partial g_{tt}}{\partial\theta} + 2\,M^tM^\phi\frac{\partial g_{t\phi}}{\partial\theta} + M^rM^r\frac{\partial g_{rr}}{\partial\theta} + M^\theta M^\theta\frac{\partial g_{\theta\theta}}{\partial\theta} + M^\phi M^\phi\frac{\partial g_{\phi\phi}}{\partial\theta}\right)\\
&= \frac{M^t\,c}{2}\frac{\partial g_{tt}}{\partial\theta} + M^\phi\,c\,\frac{\partial g_{t\phi}}{\partial\theta} + \frac{M^{r2}\,c}{2M^t}\frac{\partial g_{rr}}{\partial\theta} + \frac{M^{\theta 2}\,c}{2M^t}\frac{\partial g_{\theta\theta}}{\partial\theta} + \frac{M^{\phi 2}\,c}{2M^t}\frac{\partial g_{\phi\phi}}{\partial\theta}
\end{aligned} \tag{3.62}$$

where $M^\alpha = \overline{\mathrm{D}}u^\alpha = \overline{\overline{\mathrm{D}}}V^\alpha$ with $V^t = c$, $V^r = \frac{U}{\sqrt{g_{rr}}}$, $V^\theta = \frac{V}{\sqrt{g_{\theta\theta}}}$, $V^\phi = \omega$ and $\overline{\overline{\mathrm{D}}} = \overline{\mathrm{D}}\Gamma = D\tilde{h}\Gamma$.
All variables are located where $n_{j,k}$ is located, at the point (r_j^m, θ_k).

The 'gravity' term (Newtonian: 0) is:

$$\frac{M^{tM\theta}\,c}{2}\frac{\partial g_{tt}^{M\theta}}{\partial\theta} = \frac{\overline{\overline{\mathrm{D}}}^{M\theta}c^2}{2}\frac{\partial g_{tt}^{M\theta}}{\partial\theta} \tag{3.63}$$

$$\overset{a=0}{=} 0 \tag{3.64}$$

The frame dragging dependent term (Newtonian and Schwarzschild: 0) is:

$$M^{\phi M\theta}\,c\,\frac{\partial g_{t\phi}^{M\theta}}{\partial r} = \overline{\overline{\mathrm{D}}}^{M\theta}\,c\,V^{\phi M\theta}\frac{\partial g_{t\phi}^{M\theta}}{\partial\theta} = \overline{\overline{\mathrm{D}}}^{M\theta}\,c\,\omega^{M\theta}\frac{\partial g_{t\phi}^{M\theta}}{\partial\theta} \tag{3.65}$$

$$\overset{a=0}{=} 0 \tag{3.66}$$

The pseudo 'centripetal/centrifugal' force due to the motion in the radial direction in GR (Newtonian: 0) using

$$M^{rMR\theta 2} = (\overline{\overline{\mathrm{D}}}^{M\theta}V^{MR\theta r})^2 = \frac{(\overline{\overline{\mathrm{D}}}^{M\theta}U^{MR\theta})^2}{g_{rr}^{M\theta}} = \frac{(\sqrt{g_{rr}^{M\theta}}\overline{\overline{\mathrm{D}}}^{M\theta}U^{MR\theta})^2}{g_{rr}^{M\theta 2}} = \frac{m^{MR\theta 2}}{g_{rr}^{M\theta 2}} \tag{3.67}$$

gives:

$$\frac{M^{rMR\theta^2}\, c}{2M^{tM\theta}}\,\frac{\partial g_{rr}^{M\theta}}{\partial\theta} = \frac{m^{MR\theta^2}\, c}{2\overline{\overline{\mathrm{D}}}^{M\theta}\, c\, g_{rr}^{M\theta^2}}\,\frac{\partial g_{rr}^{M\theta}}{\partial\theta} = \frac{m^{MR\theta^2}}{2\overline{\overline{\mathrm{D}}}^{M\theta}\, g_{rr}^{M\theta^2}}\,\frac{\partial g_{rr}^{M\theta}}{\partial\theta} = \frac{\overline{\overline{\mathrm{D}}}^{M\theta}(U^{MR\theta})^2}{2\, g_{rr}^{M\theta}}\,\frac{\partial g_{rr}^{M\theta}}{\partial\theta} \tag{3.68}$$

$$\overset{a=0}{=} 0 \tag{3.69}$$

The pseudo 'centripetal/centrifugal' force due to the motion in θ direction for GR (Newtonian: 0) using

$$M^{\theta 2} = (\overline{\overline{\mathrm{D}}}^{M\theta} V^{\theta})^2 = \frac{(\overline{\overline{\mathrm{D}}}^{M\theta} V)^2}{g_{\theta\theta}^{M\theta}} \tag{3.70}$$

gives:

$$\frac{M^{\theta 2}\, c}{2M^{tM\theta}}\,\frac{\partial g_{\theta\theta}^{M\theta}}{\partial\theta} = \frac{(\overline{\overline{\mathrm{D}}}^{M\theta} V)^2\, c}{2\overline{\overline{\mathrm{D}}}^{M\theta}\, c}\,\frac{1}{g_{\theta\theta}^{M\theta}}\,\frac{\partial g_{\theta\theta}^{M\theta}}{\partial\theta} = \overline{\overline{\mathrm{D}}}^{M\theta}\frac{V^2}{2}\,\frac{1}{g_{\theta\theta}^{M\theta}}\,\frac{\partial g_{\theta\theta}^{M\theta}}{\partial\theta} = \frac{n^2}{2\overline{\overline{\mathrm{D}}}^{M\theta}\, g_{\theta\theta}^{M\theta 2}}\,\frac{\partial g_{\theta\theta}^{M\theta}}{\partial\theta} \tag{3.71}$$

$$\overset{a=0}{=} 0 \tag{3.72}$$

The pseudo centrifugal force term (Newtonian: $\rho v_\phi^2 \tan(\theta) = \rho\, \omega^2\, r^2\, \sin(\theta)\, \cos(\theta)$ using $v_\phi = \omega\, r_{cyl} = \omega\, r\, \cos(\theta)$):

$$\frac{M^{\phi M\theta}\, c^2}{2M^{tM\theta}}\,\frac{\partial g_{\phi\phi}^{M\theta}}{\partial\theta} = \frac{(\overline{\overline{\mathrm{D}}}^{M\theta} V^{\phi M\theta})^2\, c}{2\overline{\overline{\mathrm{D}}}^{M\theta}\, c}\,\frac{\partial g_{\phi\phi}^{M\theta}}{\partial\theta} = \overline{\overline{\mathrm{D}}}^{M\theta}\,\frac{\omega^{M\theta 2}}{2}\,\frac{\partial g_{\phi\phi}^{M\theta}}{\partial\theta} \tag{3.73}$$

$$\overset{a=0}{=} \overline{\overline{\mathrm{D}}}^{M\theta}\,\frac{\omega^{M\theta 2}}{2}\,\frac{\partial\left(r^{m2} cos^2(\theta_k)\right)}{\partial\theta} = \overline{\overline{\mathrm{D}}}^{M\theta}\,\frac{\omega^{M\theta 2}}{2}\, r_j^{m2}\, 2\, sin(\theta_k)\, cos(\theta_k)$$

$$= \overline{\overline{\mathrm{D}}}^{M\theta}\, \omega^{M\theta 2}\, r_j^{m2}\, sin(\theta_k)\, cos(\theta_k) \tag{3.74}$$

The discretization of the viscous source terms Q_{vis}^{θ} is not presented here.

Discretization of the angular momentum equation

For the discretization of the angular momentum equation for the axisymmetric two-dimensional grid the $D_{j,k}$-cell in figure 3.8 is used, for a fully 3-dimensional simulation in the context of staggered grid discretization one would use an in ϕ-direction shifted grid for the angular momentum equation.
So in axisymmetric discretization the same cell as for the continuity equation is used. The angular momentum l is located at (r_j^m, θ_k^m) and defined as following: $l \doteq M_\phi = \overline{\overline{\mathrm{D}}}\, g_{\phi\phi}\, d\omega$, where $d\omega = V^\phi - \omega_{FDE} = \omega - \omega_{FDE}$ and the angular momentum equation equation is (see 3.22):

$$\frac{\partial l}{\partial t} + \frac{1}{\sqrt{-g}}\frac{\partial}{\partial r}\left(\sqrt{\frac{-g}{g_{rr}}}\; U\, l\right) + \frac{1}{\sqrt{-g}}\frac{\partial}{\partial\theta}\left(\sqrt{\frac{g}{g_{\theta\theta}}}\; V\, l\right) = Q_{vis}^{\phi} \tag{3.75}$$

Due to the axisymmetric discretization, the discretization of the advection terms is similar to that of the continuity equation, just D is replaced by l.

The discretization of the source terms give:
For axisymmetric simulations with a 2-dimensional grid in r and ϕ direction, there exist no gradients and derivatives in ϕ direction, and so in case of no viscosity the source terms (longitudinal pressure gradient and possibly geometrical source terms) are all zero.

The discretization of the viscous source terms Q^{ϕ}_{vis} for the Navier-Stokes equations is not presented here.

Discretization of the internal energy equation

For the discretization of the internal energy equation the $D_{j,k}$-cell in figure 3.8, the same cell as for the continuity equation is used. On the staggered grid the internal energy density ϵ^d is located at (r^m_j, θ^m_k) and defined as following: $\epsilon^d = \epsilon\, D$. and the internal energy equation is (see 3.22):

$$\frac{\partial \epsilon^d}{\partial t} + \frac{1}{\sqrt{-g}}\frac{\partial}{\partial r}\left(\sqrt{\frac{-g}{g_{rr}}}\, U\, \epsilon^d\right) + \frac{1}{\sqrt{-g}}\frac{\partial}{\partial \theta}\left(\sqrt{\frac{-g}{g_{\theta\theta}}}\, V\, \epsilon^d\right) = - P\left[\frac{\partial}{\partial t}\left(\frac{u^t}{c}\right) + \frac{1}{\sqrt{-g}}\frac{\partial}{\partial r}\left(\sqrt{\frac{-g}{g_{rr}}}\, U\, \frac{u^t}{c}\right) + \frac{1}{\sqrt{-g}}\frac{\partial}{\partial \theta}\left(\sqrt{\frac{-g}{g_{\theta\theta}}}\, V\, \frac{u^t}{c}\right)\right] + \Phi + \hat{\Gamma} - \Lambda \tag{3.76}$$

The discretization of the advection terms is similar to that of the continuity equation, just D is replaced by ϵ^d.

The first part of the source term consists of the pressure P, which is for the equation of state of an ideal gas law: $P = (\gamma - 1)\,\rho\,\epsilon = (\gamma - 1)\,\frac{\epsilon^d}{\Gamma} = (\gamma - 1)\,\frac{\epsilon^d\, c}{u^t}$, times the advection equation for the pseudo Lorentz factor $\Gamma = \frac{u^t}{c}$. The relativistic Lorentz factor in the local Euler frame is $W = \alpha\Gamma$
The time derivative of $\Gamma = \frac{u^t}{c}$:

$$\frac{\partial \Gamma}{\partial t} \quad \text{can be discretized by:} \quad \frac{(\Gamma^{n+1} - \Gamma^n)}{\Delta t} \quad \text{or by:} \quad \Gamma\,\frac{\ln\left(\frac{\Gamma^{n+1}}{\Gamma^n}\right)}{\Delta t}, \tag{3.77}$$

whereas the advection terms of Γ, the divergence of the spatial part of the 4-velocity, are discretized in the same way the advection terms of the continuity or internal energy equation are.
Due to the time dependence of this source term there can occur problems, if the Lorentz factor is changing too quickly. Therefore care has to be taken in the method this term is discretized and how the Lorentz factor is determined.
To get a decent result for quickly changing Lorentz factors for implicit methods with large time-steps it may be necessary to do a lot of iterations.

The discretization of possible further source terms of the internal energy equation is not shown here.

Calculation of the Lorentz factor and the primitive variables

The flux-conservative form of the Hydrodynamics equations describe the time-evolution of the conserved quantities D, m, n, l and ϵ^d. However, the equation of state, the rate of transport, the applied work, cooling and heating rates used in above equations are functions of essentially the primitive variables ρ, U, V, ω and pressure P or temperature T.

Since the relation between the primitive and conservative variables is in most cases rather non-linear, it is best if an iterative solution procedure is employed (which in general is only possible if the equations are solved with an implicit method, not the explicit method).

To recover the primitive variables from the conservative ones, one first uses the normalization condition of the 4-momenta to get an updated value of the Lorentz factor:

The normalization condition of the 4-momenta is:

$$\begin{aligned} M^\mu M_\mu &= \overline{\mathrm{D}} u^\mu \, \overline{\mathrm{D}} u_\mu = \overline{\mathrm{D}}^2 \, u^\mu u_\mu = -c^2 \, \overline{\mathrm{D}}^2 \\ M^\mu M_\mu &= g^{\mu\nu} M_\nu M_\mu = g^{tt} M_t^2 + 2\, g^{t\phi} M_t M_\phi + g^{rr} M_r^2 + g^{\theta\theta} M_\theta^2 + g^{\phi\phi} M_\phi^2 = -c^2 \, \overline{\mathrm{D}}^2 \end{aligned} \tag{3.78}$$

The quantities $M_r,\ M_\theta,\ M_\phi$ are all known at the end of each time step, resulting in a quadratic equation for M_t:

$$A\, M_t^2 + B\, M_t + C = 0, \tag{3.79}$$

where $A = g^{tt},\ B = 2\, g^{t\phi} M_\phi$ and $C = g^{rr} M_r^2 + g^{\theta\theta} M_\theta^2 + g^{\phi\phi} M_\phi^2 + c^2\, \overline{\mathrm{D}}^2$ with $\overline{\mathrm{D}} = D\tilde{h}$.
Having obtained M_t, the contravariant quantity M^t can be computed using the transformation: $M^t = g^{tt}\ M_t + g^{t\phi}\ M_\phi$. Since $M^t = \overline{\mathrm{D}}\, u^t = D\tilde{h}\, u^t$ the Lorentz factor $\Gamma = \frac{u^t}{c}$ is the obtained from: $\Gamma = \frac{M^t}{D\tilde{h}c}$.
Knowing the density $\rho = \frac{D}{\Gamma}$ and the internal energy density $\epsilon = \frac{\epsilon^d}{D}$, the pressure P and the temperature T can then be calculated using the equation of state (E.o.S.), which in the case of an ideal gas is $P = (\gamma - 1)\rho\epsilon$ and $T = \frac{\mu_{gas}}{\mathcal{R}_{gas}}\ \frac{P}{\rho}$.

In the code it is also possible to use a mixed form of the normalization equation using conservative and primitive variables to calculate the Lorentz factor, which in some cases might give better results.

Advantages of the internal energy vs. total energy formulation

While analytically there is no difference if one solves an equation for the internal or the total energy in Numerics there are several reasons to use the one or the other form:

Ideally - due to the fact that the total energy should be conserved - one would ideally use the discretization of the total energy conservation equation, but there can occur some numerical problems when using the total energy equation:

$$\frac{\partial E_t}{\partial t} + \nabla(E_t + p)V = -L(T) + \ldots \tag{3.80}$$

- a problem can occur during the calculation of the internal energy if $E_{kin} \approx E_t$ and E_{in} small, since $E_{in} = E_t - E_{kin}$.

- for implicit methods it is almost impossible to determine the heating and cooling parts of the Jacobian, since they depend on the temperature and therefore on the internal energy E_{in} and not on the total energy:

$$\frac{\partial L(T)}{\partial E_t} = \frac{\partial L(T)}{\partial T}\frac{\partial T}{\partial E_t} \tag{3.81}$$

Artificial Viscosity

But there is at least one drawback in using the internal energy formulation: The energy dissipation at strong shock fronts (steep gradients in the density and pressure) is usually not calculated right and the total energy is not conserved anymore. Usually kinetic energy should be transformed into heat there. To correct this problem in the internal energy formulation one introduces the concept of **artificial viscosity**, which is only in effect at strong shock fronts and acts as an extra scalar pressure which transforms kinetic energy into heat.
As an example how artificial viscosity is used see the one-dimensional Burgers' equation on page 129. The artificial viscosity terms should be implemented in a consistent way into the equations, in order to consider the artificial viscosity as a real viscosity, which occurs in the equations like additional pressure terms and should not be omitted in the calculation of the relativistic enthalpy as in the original Wilson formulation(Wilson 1972), elsewise there can occur large errors for high Lorentz factor flows (Norman and Winkler 1986). Introducing the artificial viscosity denoted by Q, the stress-energy tensor of an ideal gas has the following modified form:

$$T^{\mu\nu} = \rho\frac{h}{c^2}u^\mu u^\nu + (P+Q)g^{\mu\nu} \qquad \text{or} \qquad T^{\mu\nu} = \frac{(\hat{\rho}+(P+Q))}{c^2}u^\mu u^\nu + (P+Q)g^{\mu\nu} \qquad [ML^{-3}L^2T^{-2}], \tag{3.82}$$

where $h = c^2 + \epsilon + \frac{(P+Q)}{\rho}$ is the relativistic enthalpy.

The artificial viscosity $Q = Q_r + Q_\theta$ is only non-zero at shock fronts:

$$Q_{dir;j,k} = \begin{cases} \alpha_{sh}\overline{\mathrm{D}}_{j,k}(\Delta V_{dir;j,k})^2 & \text{if } \Delta V_{dir;j,k} < 0 \\ 0 & \text{if } \Delta V_{dir;j,k} \geq 0 \end{cases} \tag{3.83}$$

where

$$\Delta V_{r;j,k} = \Delta U_{j,k} = \frac{[\sqrt{-g/g_{rr}}\,U\overrightarrow{u^t}]_{r_{j+1}}^{r_j}}{\int_{r_{j+1}}^{r_j}\sqrt{-g}dr}\Bigg|_k \frac{dr_j}{u^t}c \approx (U_{j,k} - U_{j+1,k}) \tag{3.84}$$

and

$$\Delta V_{\theta;j,k} = \Delta V_{j,k} = \frac{[\sqrt{-g/g_{\theta\theta}}\,V\overrightarrow{u^\theta}]_{\theta_{k+1}}^{\theta_k}}{\int_{\theta_{k+1}}^{\theta_k}\sqrt{-g}d\theta}\Bigg|_j \frac{d\theta_k}{u^t}c \approx (V_{j,k} - V_{j,k+1}). \tag{3.85}$$

This leads to following modifications and additional terms in the equations:
The relativistic enthalpy has to be modified to give:

$$h = c^2 + \epsilon + \frac{P}{\rho} + \frac{Q}{\rho}. \tag{3.86}$$

To the radial momentum equation the radial gradient of the artificial viscosity $-\frac{\partial Q}{\partial r}$ is added as source term:

$$-\frac{Q_{r;j-1,k} - Q_{r;j,k}}{dr_j^m}, \tag{3.87}$$

whereas the latitudinal gradient $-\frac{\partial Q}{\partial \theta}$ is added as source term to the latitudinal momentum equation:

$$-\frac{Q_{\theta;j,k-1} - Q_{\theta;j,k}}{d\theta_k^m}. \tag{3.88}$$

Finally to the internal energy equation following heating term due to artificial viscosity is added:

$$-Q\left[\frac{\partial u^t}{\partial t} + \frac{1}{\sqrt{-g}}\frac{\partial}{\partial r}\left(\sqrt{\frac{-g}{g_{rr}}}\,U\,\frac{u^t}{c}\right) + \frac{1}{\sqrt{-g}}\frac{\partial}{\partial \theta}\left(\sqrt{\frac{-g}{g_{\theta\theta}}}\,V\,\frac{u^t}{c}\right)\right]. \tag{3.89}$$

3.4 Explicit and Implicit Methods

3.4.1 Explicit Methods

Care has to be taken which form of equations are discretized and in which form they are discretized: The flux-conservative form of the equations is very important to obtain the right size and location of jump conditions for shocks. To numerically conserve the angular momentum, it is not sufficient to solve the equation of the velocity or linear momentum, one explicitly has to discretize the angular momentum equation, elsewise the angular momentum is not conserved numerically.

Many simulation codes are based on the **Godunov type scheme** Godunov (1959) , which can be described by a three step algorithm:

- **Reconstruct**:
  From the cell-averages given at every grid point the fluxes at the cell interfaces have to be reconstructed, see Figure 3.9:
  - donor cell: the cell average value is used as the (left or right) interface value.
  - piecewise linear method (PLM): a straight line with a slope depending on the neighbouring cell-average values is determined
  - piecewise parabolic method (PPM) (Colella and Woodward 1984): a parabola is constructed also using the neighbouring cell-average values

  As described later in the construction of the interface values slope or flux limiter are important to guaranty numerical stability for higher order methods.
- **Solve (or advect)**: Then the local Riemann problem is solved at each cell interface using approximate Riemann solvers:
  - Lax-Friedrichs
  - Roe (Roe 1981)
  - HLL (Harten et al. 1983)
  - HLLC (Toro et al. 1994)
  - ...

  Or one can use other advection schemes like the ones from van-Leer to advect the solution.
- **Average**:
  The last step is then to average the solution to get the cell averaged values at every grid point.

There is a problem for higher order schemes, because due to a theorem by Godunov every second or higher order scheme gets unstable at shock discontinuities.
These numerical instabilities near shock discontinuities can be avoided by using so-called slope limiters or flux limiters, which use higher order schemes in smooth regions and switch to first order near shock discontinuities, so the so-called total variation diminishing (TVD) property is fulfilled and the oscillations near shock fronts are not amplified. This approach was first introduced by van Leer (van Leer 1979) under the name in his Monotonic Upstream-centred Scheme for Conservation Laws (MUSCL) and the concept of flux limiter in the flux-corrected transport (FCT) algorithm by Boris and Book. Some of the different slope limiters are: minmod limiter, superbee, monotonized central-difference (MC-) limiter.

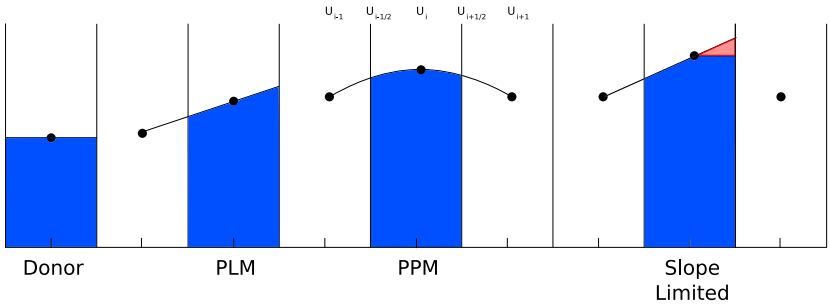


Figure 3.9: *Interpolation of the averaged grid cell values to the cell interfaces: 1st order donor cell, 2nd order PLM and 3rd order PPM interpolation and on the right side, the interpolated value is reduced by the slope limiter function $\phi(r)$ (plot from Hilscher 2009).*

There are many well developed High Resolution Shock Capturing (HRSC) methods available, which mostly use Riemann solvers.

Riemann solver based methods are best suited to resolve shocks, because shocks are itself a Riemann problem.

But Riemann solvers cannot be used in implicit methods, since they itself rely on the time-step size. Instead other advection methods (van Leer advection) have to be used, which do not have themself a strong dependence on time.

Depending on the problem one wants to solve there might be another problem for explicit methods:

For explicit methods there exists a maximum allowed time step, at which the solutions from different Riemann problems at neighbouring interfaces still do not influence each other and lead to a numerical unstable scheme:
This limiting time step is described by the **CFL condition**, which is named after Courant, Friedrichs and Lewy:

$$CFL = \frac{a_{max}\Delta t}{\Delta x} \tag{3.90}$$

where $a_{max} = \max|\lambda_j|$ is the maximum of the modulus of the characteristic speeds at all grid cells (the maximal modulus of the eigenvalue of the flux Jacobian matrix).
Therefore one gets for the time step:

$$\Delta t = CFL\frac{\Delta x}{a_{max}} \tag{3.91}$$

where CFL is limited by 1 or an even smaller value like 0.5 depending on the numerical method used.

More generally the CFL condition states: (see Leveque 1998): A numerical method can only be stable and converge if its numerical domain of dependence contains the true domain of dependence of the partial differential equations, at least in the limit of very fine grid spacing and very short time steps.

This condition restricts the time step for explicit methods.

For an implicit solver the CFL condition is satisfied for any time step, because the numerical domain of dependence is the entire grid, since, due to the coupling of the linear system, the solution at each grid point depends on the data at all grid points.

3.4.2 Implicit methods

In explicit methods the variable values of the grid cell itself and its neighbours of the current time level n are used to calculate the values of the state variables $\vec{q}$ at the future time level $n+1$
Using $\vec{q}^{\,n}_{j,k}, \vec{q}^{\,n}_{j+1,k}, \vec{q}^{\,n}_{j-1,k}, \vec{q}^{\,n}_{j,k+1}$ and $\vec{q}^{\,n}_{j,k-1}$ one determines $\vec{q}^{\,n+1}_{j,k}$.
The equation in flux conservative vector form is:

$$\frac{\Delta\vec{q}}{\Delta t} + L(\vec{q}) = \vec{f}, \tag{3.92}$$

where L correspond to the advection operator and $\vec{f}$ to external forces.
Defining $R\vec{H}S \doteq \vec{f} - L(\vec{q})$ this equation can be written in a different form:

$$\frac{\Delta\vec{q}}{\Delta t} = R\vec{H}S\,. \tag{3.93}$$

Note that $R\vec{H}S$ is zero if a stationary, that means time-independent, solution is found.

To get an explicit method by adopting a time-forward discretization procedure, the unknown vector q at the new time level can be extrapolated as follows using $\Delta\vec{q}^{\,n} = \vec{q}^{\,n+1} - \vec{q}^{\,n}$:

$$\vec{q}^{\,n+1} = \vec{q}^{\,n} + \Delta t \cdot R\vec{H}S^{\,n}, \tag{3.94}$$

where the right hand side of this equation only depends on the variables at the current time level n.
Depending on the time step size and on the number of grid points, the numerical procedure can be made sufficiently accurate in space and time.

In implicit methods the - yet not known - variable values of the grid cell neighbours already in the future time are used.
The future value of the state variable $\vec{q}^{\,n+1}_{j,k}$ is determined by: $\vec{q}^{\,n+1}_{j+1,k}, \vec{q}^{\,n+1}_{j-1,k}, \vec{q}^{\,n+1}_{j,k+1}$ and $\vec{q}^{\,n+1}_{j,k-1}$ and possibly also $\vec{q}^{\,n}_{j,k}, \vec{q}^{\,n}_{j+1,k}, \vec{q}^{\,n}_{j-1,k}, \vec{q}^{\,n}_{j,k+1}$ and $\vec{q}^{\,n}_{j,k-1}$.
This is only possible by solving a linear system of equations.
The construction of implicit methods is described in the following.

The hydrodynamics equations can be written in the discretized conservative vector form:

$$\frac{\Delta\vec{q}}{\Delta t} + L_{\mathrm{r,rr}}\vec{F} + L_{\theta,\theta\theta}\vec{G} = \vec{f}, \tag{3.95}$$

where $\vec{q}$ is the vector of the conservative variables, $\Delta\vec{q}$ the change of $\vec{q}$ during the time step Δt, $\vec{F}$ and $\vec{G}$ are the fluxes of $\vec{q}$ in r and θ-direction and $\vec{f}$ denotes the vector of source terms. $L_{\mathrm{r,rr}}$ and $L_{\theta,\theta\theta}$ are the first and second order operators that describe the advection and diffusion of the vector variables $\vec{q}$

in r and θ-direction.
Defining the vector $\vec{RHS}$ as before:

$$\vec{RHS} \doteq \vec{f} - L(\vec{q}) = \vec{f} - L_{\mathrm{r,rr}}\vec{F} - L_{\theta,\theta\theta}\vec{G} \tag{3.96}$$

one can write the equation in following form:

$$\frac{\Delta\vec{q}}{\Delta t} = \vec{RHS}, \tag{3.97}$$

For the explicit method $\vec{RHS} = \vec{RHS}^{\,n}$ is calculated using the state variables at the current time level n, whereas for implicit methods $\vec{RHS} = \vec{RHS}^{\,n+1}$ is determined using the state variables at the next time level $n+1$. In order to obtain second order temporal accuracy, one can mix explicit and implicit methods in the so-called **Crank-Nicolson method** (see Hujeirat and Rannacher 2001, for details):

$$\vec{RHS} = \vec{RHS}(\vec{q}^{\,n+1}, \vec{q}^{\,n}) = \vartheta \cdot \vec{RHS}^{\,n+1} + (1-\vartheta) \cdot \vec{RHS}^{\,n} \tag{3.98}$$

where $0 \leq \vartheta \leq 1$ is a parameter called Crank-Nicolson number, that may depend also on the time step size. The pure explicit method is achieved with $\vartheta = 0$, whereas the pure implicit method is retrieved with $\vartheta = 1$, Second temporal order is only achieved if $\vartheta = 0.5$. But only for $\vartheta > 0.5$, when the scheme is more implicit than explicit, the method is unconditionally stable.

The **Prediction-Correction Iteration Procedure**, also called defect correction strategy, is used in order to assure the accuracy of the numerical scheme. Therefore one can re-write equation 3.97 in the **Residual form**:

$$\vec{R} = L_{\vec{q}} = \vec{R}(\vec{q}^{\,n+1}, \vec{q}^{\,n}) = \frac{\Delta\vec{q}}{\Delta t} - \vec{RHS} = \frac{\vec{q}^{\,n+1} - \vec{q}^{\,n}}{\Delta t} - \vec{RHS}(\vec{q}^{\,n+1}, \vec{q}^{\,n}) = 0 \tag{3.99}$$

where $\vec{R}$ is called the residuum or the defect, therefore the name: defect correction strategy, which would ideally be zero if the equations would have been exactly solved. But in Numerics - also due to the errors caused by machine inaccuracy and the limited accuracy of machine numbers - it is only possible to solve this equation up to a specified accuracy.

One can define the Jacobian matrix J which consists of the derivatives of each equation at each grid cell over each variable at each grid cell:

$$J \doteq \frac{\partial\vec{R}}{\partial\vec{q}} \doteq \frac{\partial\vec{R}(\vec{q}^{\,n+1}, \vec{q}^{\,n})}{\partial\vec{q}^{\,n+1}} \tag{3.100}$$

where $\vec{q}$ is a vector with entries of each state variable with index $ivar$ at each grid cell j,k (vector size: $(nvar \cdot jsize \cdot ksize) \times 1$) and $\vec{R}$ is the residuum vector with entries of each equation with index ieq at each grid cell j,k (vector size: $(neq \cdot jsize \cdot ksize) \times 1$, where $neq = nvar$) and J is a $(neq \cdot jsize \cdot ksize) \times (nvar \cdot jsize \cdot ksize)$ matrix. and where $ieq, ivar$ are integers that run over the number of equations and variables. To minimize the computational effort one usually takes only the derivative of the spatially first order form of the equations to determine the Jacobian, whereas for the calculation of the residuum/defect one usually uses a high order accurate advection scheme.

Using the multi-dimensional Newton-Raphson iteration procedure for finding roots one can - starting with the values of the state vector of the old time step as initial guess $\vec{q}^{n+1;i=0} \doteq \vec{q}^{i=0} = \vec{q}^{n}$ - determine the values of $\vec{q}^{n+1}$ at the next iteration step by:

$$\vec{q}^{i+1} = \vec{q}^{i} - J^{i-1}\vec{R}^{i}, \tag{3.101}$$

where $i, i+1$ denote the current and the next iteration index ('inside' one time step!) and J^{i-1} is the inverse of the Jacobian matrix J^{i}, which represents the derivatives of the residuum over the state variables $\vec{q}^{n+1;i} \doteq \vec{q}^{i}$ and the residuum $\vec{R}^{i} \doteq \vec{R}(\vec{q}^{n+1;i}, \vec{q}^{n})$
To save computational time often the Jacobian is set constant during one time step $J^{i} = J^{i=0} = J$ and only $\vec{R}$ is updated to the new iteration level $i+1$ using the new state values $\vec{q}^{i+1}$.

Transforming this equation one gets:

$$\vec{q}^{i+1} - \vec{q}^{i} = -J^{i-1}\vec{R}^{i}, \tag{3.102}$$

and therefore by defining $\delta\vec{q}^{i} = \vec{q}^{i+1} - \vec{q}^{i}$ (which should not be confused with the discretization term $\Delta\vec{q}$ in $\frac{\Delta\vec{q}}{\Delta t} = \frac{\vec{q}^{n+1} - \vec{q}^{n}}{\Delta t} = \frac{\vec{q}^{n+1;i} - \vec{q}^{n}}{\Delta t} = \frac{\vec{q}^{i} - \vec{q}^{n}}{\Delta t}$ in the calculation of $\vec{R}^{i}$, which is the same if one does only one iteration):

$$J^{i}\delta\vec{q}^{i} = -\vec{R}^{i}. \tag{3.103}$$

Writing this out for each grid cell with indices j and k using the five star staggered grid discretization one gets following block matrix structure:

$$\frac{\delta q^{i}_{j,k}}{\Delta t} + \underline{S}^{r}\delta q^{i}_{j-1,k} + \mathcal{D}^{r}\delta q^{i}_{j,k} + \overline{S}^{r}\delta q^{i}_{j+1,k} + \underline{S}^{\theta}\delta q^{i}_{j,k-1} + \mathcal{D}^{\theta}\delta q^{i}_{j,k} + \overline{S}^{\theta}\delta q^{i}_{j,k+1} = -R^{i}_{j,k}, \tag{3.104}$$

where $\delta q^{i} = q^{i+1} - q^{i}$ and the subscripts j and k denote the cell numbering in the r and θ direction respectively.
The diagonal, sub- and super-diagonal block terms are defined as following:

$$\underline{S}^{r} = \underline{S}^{r}_{j,k} = \frac{\partial\vec{R}^{i}}{\partial\vec{q}^{i}_{j-1,k}}, \qquad \mathcal{D}^{r} = \mathcal{D}^{r}_{j,k} = \frac{\partial\vec{R}^{r;i}}{\partial\vec{q}_{j,k}}, \qquad \overline{S}^{r} = \overline{S}^{r}_{j,k} = \frac{\partial\vec{R}^{i}}{\partial\vec{q}^{i}_{j+1,k}}, \tag{3.105}$$

$$\underline{S}^{\theta} = \underline{S}^{\theta}_{j,k} = \frac{\partial\vec{R}^{i}}{\partial\vec{q}^{i}_{j,k-1}}, \qquad \mathcal{D}^{\theta} = \mathcal{D}^{\theta}_{j,k} = \frac{\partial\vec{R}^{\theta;i}}{\partial\vec{q}_{j,k}}, \qquad \overline{S}^{\theta} = \overline{S}^{\theta}_{j,k} = \frac{\partial\vec{R}^{i}}{\partial\vec{q}^{i}_{j,k+1}}, \tag{3.106}$$

where $\vec{R}^{r;i}$ is the r dependent part of $\vec{R}^{i}$, whereas $\vec{R}^{\theta;i}$ is the θ dependent part of $\vec{R}^{i}$, to also easily allow one dimensional simulations.
With $\mathcal{D}_{\mathrm{mod}} = \frac{1}{\Delta t} + \mathcal{D}^{r} + \mathcal{D}^{\theta}$ one gets the reordered block matrix structure:

$$\begin{array}{ccc} & \overline{S}^{\theta}\delta q^{i}_{j,k+1} & \\ +\underline{S}^{r}\delta q^{i}_{j-1,k} & +\mathcal{D}_{\mathrm{mod}}\delta q^{i}_{j,k} & +\overline{S}^{r}\delta q_{j+1,k} \\ & +\underline{S}^{\theta}\delta q^{i}_{j,k-1}, & \end{array} = -R^{i}_{j,k} \tag{3.107}$$

Writing this equation for all equations and all grid cells with indices j, k in matrix form:

$$A\delta\vec{q}^{i} = -\vec{R}^{i} \tag{3.108}$$

where $A = J^i$.
To get a solution of this sparse linear matrix equation one constructs a simplified matrix $\tilde{A}$, the preconditioner, which is similar to A, which means that $\tilde{A}$ and A share the same spectral properties, i.e. have the same eigenvalues.
The preconditioner is constructed in such a way, that the iterative solution procedure converges much more faster to the real solution.
So following similar equation is solved:

$$\tilde{A}\delta\vec{q}^{\,i} = -\vec{R}^i \tag{3.109}$$

From the solution $\delta\vec{q}^{\,i}$ one can determine a new $\vec{q}^{\,n+1}$:

$$\vec{q}^{\,i+1} = \vec{q}^{\,i} + \delta\vec{q}^{\,i} \tag{3.110}$$

If the maximum norm of the residuum/defect $\|\vec{R}\|_\infty$ is not sufficiently small (because in general $\tilde{A} \neq A = J^i$), or the maximum iteration number i_{max} is not reached, a new $\vec{R}$ and possibly $\tilde{A}$ is calculated and the matrix equation 3.109 is solved again.

For a stationary solution, where $\vec{q}$ does not depend on the time t anymore $R\vec{H}S$ must be zero.

Structure of the matrix equation

In this section the structure of the matrix equation, in particular the structure of the Jacobian matrix, is looked at.

For the calculation of the Jacobian at a specified grid point not only the derivatives of the residuum with respect to the variables at this grid point have to be considered, but also the derivative of the residuum with respect to the variables at the neighbouring points.

The Jacobian for one equation for a 1-dimensional grid:
In the case of one equation with a 1-dimensional grid in r-direction the Jacobian entries at the grid point j are:

$$\begin{aligned}
&\text{Diagonal :} && \mathcal{D}_j &&= \tfrac{\partial R_j}{\partial q_j} && = J_{j,j}, \\
&\text{Subdiagonal :} && \underline{\mathrm{S}}_j &&= \tfrac{\partial R_j}{\partial q_{j+1}} && = J_{j,j+1}, \\
&\text{Superdiagonal :} && \overline{\mathrm{S}}_j &&= \tfrac{\partial R_j}{\partial q_{j-1}} && = J_{j,j-1},
\end{aligned} \tag{3.111}$$

or written in matrix form $J\delta\vec{q} = -\vec{R}$, where $A = J$ and $\delta\vec{q} = \vec{q}^{\,i+1} - \vec{q}^{\,i}$:

$$\begin{pmatrix}
\mathcal{D}_{J1G} & \overline{S}_{J1G} & & & \\
\underline{S}_{J1G+1} & \mathcal{D}_{J1G+1} & \overline{S}_{J1G+1} & & \\
& \ddots & \ddots & \ddots & \\
& & & \underline{S}_{J2G} & \mathcal{D}_{J2G}
\end{pmatrix}
\begin{pmatrix} \delta q_{J1G} \\ \delta q_{J1G+1} \\ \vdots \\ \delta q_{J2G} \end{pmatrix}
=
\begin{pmatrix} -R_{J1G} \\ -R_{J1G+1} \\ \vdots \\ -R_{J2G} \end{pmatrix}, \tag{3.112}$$

where $j = J1G$ is the start index and $j = J2G$ is the end index of the computational domain in $r-$*direction*. This is a purely tridiagonal matrix, which can be solved very efficiently by the Thomas algorithm which uses the Gaussian elimination procedure: For a matrix of the size $N\times N$ the algorithm obtains the solution with a computational cost of $O(N)$ instead of $O(N^3)$ for the Gaussian elimination.

The Jacobian for a system of equations for a 1-dimensional grid:
When coupling more equations together one retains for the one dimensional grid a *block* tri-diagonal matrix structure instead of only tridiagonal matrices. As an example here the radial momentum equation with conservative variable m and the continuity equation with conservative variable D are coupled, the residuum is defined as:

$$\vec{R}_j = \begin{pmatrix} R_j^m \\ R_j^D \end{pmatrix} \tag{3.113}$$

The resulting Jacobian matrix looks similar to the matrix in (3.112) but now the $\mathcal{D}, \underline{S}, \overline{S}$ are not scalars anymore but block matrices of the form

$$\mathcal{D}_j = \begin{pmatrix} \frac{\partial R_j^m}{\partial m_j} & \frac{\partial R_j^m}{\partial D_j} \\ \frac{\partial R_j^D}{\partial m_j} & \frac{\partial R_j^D}{\partial D_j} \end{pmatrix}, \qquad \underline{S}_j = \begin{pmatrix} \frac{\partial R_j^m}{\partial m_{j-1}} & \frac{\partial R_j^m}{\partial D_{j-1}} \\ \frac{\partial R_j^D}{\partial m_{j-1}} & \frac{\partial R_j^D}{\partial D_{j-1}} \end{pmatrix}, \qquad \overline{S}_j = \begin{pmatrix} \frac{\partial R_j^m}{\partial m_{j+1}} & \frac{\partial R_j^m}{\partial D_{j+1}} \\ \frac{\partial R_j^D}{\partial m_{j+1}} & \frac{\partial R_j^D}{\partial D_{j+1}} \end{pmatrix}. \tag{3.114}$$

The resulting matrix equation looks like:

$$\begin{pmatrix} \mathcal{D}_{J1G} & \overline{S}_{J1G} & & & & & & \\ \underline{S}_{J1G+1} & \mathcal{D}_{J1G+1} & \overline{S}_{J1G+1} & & & & & \\ & \ddots & \ddots & \ddots & & & & \\ & & \underline{S}_j & \mathcal{D}_j & \overline{S}_j & & & \\ & & & \ddots & \ddots & \ddots & & \\ & & & & \ddots & \ddots & \overline{S}_{J2G-1} & \\ & & & & & \underline{S}_{J2G} & \mathcal{D}_{J2G} & \end{pmatrix} \begin{pmatrix} \vdots \\ \vdots \\ \begin{bmatrix} \delta m \\ \delta D \end{bmatrix}_{j-1} \\ \begin{bmatrix} \delta m \\ \delta D \end{bmatrix}_{j} \\ \begin{bmatrix} \delta m \\ \delta D \end{bmatrix}_{j+1} \\ \vdots \\ \vdots \end{pmatrix} = \begin{pmatrix} \vdots \\ \vdots \\ \begin{bmatrix} -R^m \\ -R^D \end{bmatrix}_{j-1} \\ \begin{bmatrix} -R^m \\ -R^D \end{bmatrix}_{j} \\ \begin{bmatrix} -R^m \\ -R^D \end{bmatrix}_{j+1} \\ \vdots \\ \vdots \end{pmatrix}. \tag{3.115}$$

Therefore for a one-dimensional grid one gets a Jacobian matrix in block tridiagonal matrix form which has a size of $mN \times mN$, where N denotes the number of grid points and m the number of equations. The computational cost to solve the linear system increases to $O((3m)^2 N)$.

The Jacobian for two-dimensions:
Since in the discretization of the equations a five-point stencil, see Fig. (3.10), is used, the Jacobian includes contributions from the two directions r and θ. So our system of equations will lead to the

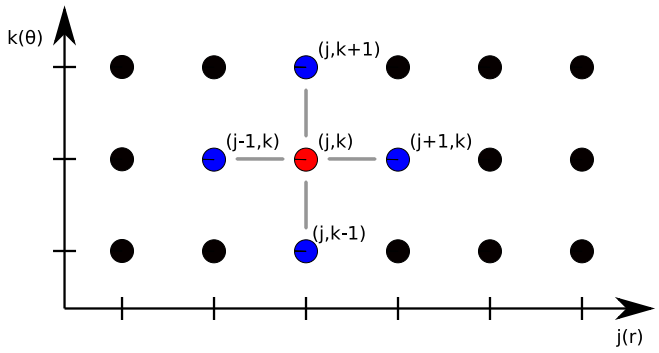


Figure 3.10: *The five-point stencil used in the discretization of the equations. Note that here the grid of the indices j and k are drawn with increasing j going from the left to the right, if one would draw the real grid cells, the index numbering in radial direction would be from right to left, in the opposite direction of increasing radius! Similar for θ: If the index k increases the value of θ decreases.*

following already mentioned block structure:

$$\begin{array}{ccc} & \overline{S}^{\theta}\delta q^{i}_{j,k+1} & \\ +\underline{S}^{r}\delta q^{i}_{j-1,k} & +\mathcal{D}_{\mathrm{mod}}\delta q^{i}_{j,k} & +\overline{S}^{r}\delta q_{j+1,k} \\ & +\underline{S}^{\theta}\delta q^{i}_{j,k-1}, & \end{array} = -R^{i}_{j,k} \tag{3.116}$$

The matrix entries are calculated by sweeping through the whole grid. First the points in the r-direction for a θ index of $K1G$, then next for the r-direction with $\theta = K1G + 1$ and so on. After proceeding through all grid points in this way the matrix looks like:

$$\begin{pmatrix} \mathcal{D}_{\tilde{j},\tilde{k}} & \overline{S}^{r}_{\tilde{j},\tilde{k}} & & \overline{S}^{\theta}_{\tilde{j},\tilde{k}} & & & & & \\ \underline{S}^{r}_{\tilde{j}+1,\tilde{k}} & \mathcal{D}_{\tilde{j}+1,\tilde{k}} & \overline{S}^{r}_{\tilde{j}+1,\tilde{k}} & & \ddots & & & & \\ & \ddots & \ddots & \ddots & & \ddots & & & \\ \underline{S}^{\theta}_{\tilde{j},\tilde{k}+1} & & \ddots & \ddots & \ddots & & \ddots & & \\ & \ddots & & \ddots & \ddots & \ddots & & \ddots & \\ & & \ddots & & \ddots & \ddots & \ddots & & \overline{S}^{\theta}_{J,K-1} \\ & & & \ddots & & \ddots & \ddots & \ddots & \\ & & & & \ddots & & \ddots & \ddots & \overline{S}^{r}_{J-1,K} \\ & & & & & \underline{S}^{\theta}_{J,K} & & \underline{S}^{r}_{J,K} & \mathcal{D}_{J,K} \end{pmatrix} \begin{pmatrix} \delta q_{\tilde{j},\tilde{k}} \\ \vdots \\ \delta q_{J,\tilde{k}} \\ \delta q_{\tilde{j},\tilde{k}+1} \\ \vdots \\ \delta q_{J,\tilde{k}+1} \\ \\ \vdots \\ \\ \delta q_{J,K} \end{pmatrix} = \begin{pmatrix} -R_{\tilde{j},\tilde{k}} \\ \vdots \\ -R_{J,\tilde{k}} \\ -R_{J1G,\tilde{k}+1} \\ \vdots \\ -R_{J,\tilde{k}+1} \\ \\ \vdots \\ \\ -R_{J,K} \end{pmatrix}, \tag{3.117}$$

where $\tilde{j} = J1G$ and $J = J2G$ are the start and end indices in r-direction and $\tilde{k} = K1G$ and $K = K2G$ are the start and end indices in θ-direction of the grid. The inclusion of the second dimension produces two bands far apart from the diagonal, which are called *fringes*. But otherwise all other entries of the matrix are zero and so the resulting Jacobian is still a highly sparse matrix.

Boundary Conditions and their Effect on the Matrix Construction

Since the numerical grid stored on a computer has to have a beginning and an end, one needs to define boundary conditions. Especially in Astrophysics one actually wants ideally simulate up to infinity since there is no physical boundary, but the computational domain used in the simulation has to be finite.

To make it possible to easily define several different boundary conditions following scheme was designed:

The boundary conditions are applied to the conservative variables and their effect is also taken into account in the construction of the solution matrix.

At the outer radial boundary R_{out} the boundary conditions are:

$$D_{J1G-1,k} = a_{D;out}(k)\, D_{J1G,k} + b_{D;out}(k)\, D_{J1G+1,k} + c_{D;out}(k) \tag{3.118}$$

$$m_{J1G,k} = a_{m;out}(k)\, m_{J1G+1,k} + b_{m;out}(k)\, m_{J1G+2,k} + c_{m;out}(k) \tag{3.119}$$

$$n_{J1G-1,k} = a_{n;out}(k)\, n_{J1G,k} + b_{n;out}(k)\, n_{J1G+1,k} + c_{n;out}(k) \tag{3.120}$$

$$l_{J1G-1,k} = a_{l;out}(k)\, l_{J1G,k} + b_{l;out}(k)\, l_{J1G+1,k} + c_{l;out}(k) \tag{3.121}$$

$$\epsilon^d_{J1G-1,k} = a_{\epsilon^d;out}(k)\, \epsilon^d_{J1G,k} + b_{\epsilon^d;out}(k)\, \epsilon^d_{J1G+1,k} + c_{\epsilon^d;out}(k) \tag{3.122}$$

Note: $m_{J1G,k}$ is still the boundary!

Whereas at the inner radial boundary R_{in} the boundary conditions are:

$$D_{J2G+1,k} = a_{D;in}(k)\, D_{J2G,k} + b_{D;in}(k)\, D_{J2G-1,k} + c_{D;in}(k) \tag{3.123}$$

$$m_{J2G+1,k} = a_{m;in}(k)\, m_{J2G,k} + b_{m;in}(k)\, m_{J2G-1,k} + c_{m;in}(k) \tag{3.124}$$

$$n_{J2G+1,k} = a_{n;in}(k)\, n_{J2G,k} + b_{n;in}(k)\, n_{J2G-1,k} + c_{n;in}(k) \tag{3.125}$$

$$l_{J2G+1,k} = a_{l;in}(k)\, l_{J2G,k} + b_{l;in}(k)\, l_{J2G-1,k} + c_{l;in}(k) \tag{3.126}$$

$$\epsilon^d_{J2G+1,k} = a_{\epsilon^d;in}(k)\, \epsilon^d_{J2G,k} + b_{\epsilon^d;in}(k)\, \epsilon^d_{J2G-1,k} + c_{\epsilon^d;in}(k) \tag{3.127}$$

Up to now the North and South boundaries are fixed:
At the North ($\theta = \pi/2$), the pole, axi-symmetric boundary conditions and at the South ($\theta = 0$), the midplane, reflecting boundary conditions are applied.

Let us take a further look at how the outer radial boundary conditions influence the matrix construction by looking only at the one-dimensional radial equations for clearness:

The residuum of the continuity equation at a inner grid point (j,k) is:

$$R^D_{j,k} = \frac{D^{n+1}_{j,k} - D^n_{j,k}}{\Delta t} + \left. \frac{\left[\bar{\rho}\sqrt{\Delta}\, U \overrightarrow{D^r}\right]_{r=r_j} - \left[\bar{\rho}\sqrt{\Delta}\, U \overrightarrow{D^r}\right]_{r=r_{j+1}}}{\int_{r_{j+1}}^{r_j} \bar{\rho}^2\, dr} \right|_k , \tag{3.128}$$

which can be rewritten in case of only using the first order donor cell method: using:

$$\left. \left[\bar{\rho}\sqrt{\Delta}\, U \overrightarrow{D^r}\right]_{r=r_j} \right|_k = -\max\left[-\left[\bar{\rho}\sqrt{\Delta}\, U\right]_{j,k}, 0\right] \; D_{j-1,k} + \max\left[\left[\bar{\rho}\sqrt{\Delta}\, U\right]_{j,k}, 0\right] \; D_{j,k} \tag{3.129}$$

gives:

$$\begin{aligned} R^D_{j,k} &= \frac{D^{n+1}_{j,k} - D^n_{j,k}}{\Delta t} + \frac{1}{\int_{r_{j+1}}^{r_j} \overline{\rho}^2 \, dr \Big|_k} \\ &\cdot \left[- \max \left[- \left[\overline{\rho} \sqrt{\Delta} \, U \right]_{j,k}, 0 \right] D_{j-1,k} \right. \\ &+ \left(\max \left[\left[\overline{\rho} \sqrt{\Delta} \, U \right]_{j,k}, 0 \right] + \max \left[- \left[\overline{\rho} \sqrt{\Delta} \, U \right]_{j+1,k}, 0 \right] \right) D_{j,k} \\ &\left. - max \left[\left[\overline{\rho} \sqrt{\Delta} \, U \right]_{j+1,k}, 0 \right] D_{j+1,k} \right] \end{aligned} \tag{3.130}$$

where $D_{j,k} = D^{n+1}_{j,k}$.

The Jacobian entries are calculated by taking the derivative of the residuum at a grid cell with respect to the state variables at this and all other grid cells:

$$\frac{\partial R^D_{j,k}}{\partial D_{j-1,k}} = \frac{- \max \left[- \left[\overline{\rho} \sqrt{\Delta} \, U \right]_{j,k}, 0 \right]}{\int_{r_{j+1}}^{r_j} \overline{\rho}^2 \, dr \Big|_k} \tag{3.131}$$

$$\frac{\partial R^D_{j,k}}{\partial D_{j,k}} = \frac{1}{\Delta t} + \frac{\max \left[\left[\overline{\rho} \sqrt{\Delta} \, U \right]_{j,k}, 0 \right] + \max \left[- \left[\overline{\rho} \sqrt{\Delta} \, U \right]_{j+1,k}, 0 \right]}{\int_{r_{j+1}}^{r_j} \overline{\rho}^2 \, dr \Big|_k} \tag{3.132}$$

$$\frac{\partial R^D_{j,k}}{\partial D_{j+1,k}} = \frac{-max \left[\left[\overline{\rho} \sqrt{\Delta} \, U \right]_{j+1,k}, 0 \right]}{\int_{r_{j+1}}^{r_j} \overline{\rho}^2 \, dr \Big|_k} \tag{3.133}$$

But at the grid point $(J1G, k)$ at the outer radial boundary the residuum taking into account the boundary condition

$$D_{J1G-1,k} = a_{D;out}(k) \, D_{J1G,k} + b_{D;out}(k) \, D_{J1G+1,k} + c_{D;out}(k) \tag{3.134}$$

is:

$$\begin{aligned} R^D_{J1G,k} &= \frac{D^{n+1}_{J1G,k} - D^n_{J1G,k}}{\Delta t} + \frac{1}{\int_{r_{J1G+1}}^{r_J 1G} \overline{\rho}^2 \, dr \Big|_k} \\ &\cdot \left[- \max \left[- \left[\overline{\rho} \sqrt{\Delta} \, U \right]_{J1G,k}, 0 \right] \left(a_{D;out}(k) \, D_{J1G,k} + b_{D;out}(k) \, D_{J1G+1,k} + c_{D;out}(k) \right) \right. \\ &+ \left(\max \left[\left[\overline{\rho} \sqrt{\Delta} \, U \right]_{J1G,k}, 0 \right] + \max \left[- \left[\overline{\rho} \sqrt{\Delta} \, U \right]_{J1G+1,k}, 0 \right] \right) D_{J1G,k} \\ &\left. - max \left[\left[\overline{\rho} \sqrt{\Delta} \, U \right]_{J1G+1,k}, 0 \right] D_{J1G+1,k} \right] \end{aligned} \tag{3.135}$$

where $D_{J1G,k} = D^{n+1}_{J1G,k}$, which results in following matrix entries:

$$\frac{\partial R^D_{J1G,k}}{\partial D_{J1G,k}} = \frac{1}{\Delta t} + \frac{\max\left[\left[\bar{\rho}\sqrt{\Delta}\,U\right]_{J1G,k}, 0\right] + \max\left[-\left[\bar{\rho}\sqrt{\Delta}\,U\right]_{J1G+1,k}, 0\right]}{\int_{r_{J1G+1}}^{r_J 1G} \bar{\rho}^2\, dr\Big|_k} + a_{D;out}(k)\frac{-\max\left[-\left[\bar{\rho}\sqrt{\Delta}\,U\right]_{J1G,k}, 0\right]}{\int_{r_{J1G+1}}^{r_J 1G} \bar{\rho}^2\, dr\Big|_k} \tag{3.136}$$

$$\frac{\partial R^D_{J1G,k}}{\partial D_{J1G+1,k}} = \frac{-max\left[\left[\bar{\rho}\sqrt{\Delta}\,U\right]_{J1G+1,k}, 0\right]}{\int_{r_{J1G+1}}^{r_J 1G} \bar{\rho}^2\, dr\Big|_k} + b_{D;out}(k)\frac{-\max\left[-\left[\bar{\rho}\sqrt{\Delta}\,U\right]_{J1G,k}, 0\right]}{\int_{r_{J1G+1}}^{r_J 1G} \bar{\rho}^2\, dr\Big|_k}, \tag{3.137}$$

where the red-coloured terms are additional terms due to the boundary conditions.

The boundary condition at the outer radial boundary R_{out} for the radial momentum equation is:

$$m_{J1G,k} = a_{m;out}(k)\, m_{J1G+1,k} + b_{m;out}(k)\, m_{J1G+2,k} + c_{m;out}(k), \tag{3.138}$$

so $m_{J1G,k}$ is still the boundary. But the matrix equation is still solved also for $m_{J1G,k}$, so it should be pseudo-solved there!
This is done by setting all entries of the matrix corresponding to $m_{J1G,k}$ to zero, except the diagonal element to $\frac{1}{\Delta t}$, and setting the right hand side vector entries at $(J1G, k)$ to zero, so that the solution $\delta m_{J1G,k}$ of this linear equation is zero.

Depending on the values of the functions $a_{X;B}(k), b_{X;B}(k), c_{X;B}(k)$, where $X = D, m, n, l, \epsilon^d$ and $B = in, out$ many **possible boundary types** can be constructed, from which some are predefined.

E.g. to get a **fixed** density boundary at *Rout* one sets:

$$a_{D,out}(k) = 0 \tag{3.139}$$

$$b_{D,out}(k) = 0 \tag{3.140}$$

$$c_{D,out}(k) = \frac{\rho_{out;0}}{\rho_{DIM}} \cdot \Gamma_{J1G-1,k} \tag{3.141}$$

A **zero gradient** boundary of the conservative variable is obtained by:

$$a_{X,B}(k) = 1 \tag{3.142}$$

$$b_{X,B}(k) = 0 \tag{3.143}$$

$$c_{X,B}(k) = 0 \tag{3.144}$$

For the **linear extrapolation** of the conservative variable at the inner boundary one sets (see staggered grid structure in radial direction in figure 3.6):

$$a_{C,in}(k) = 1 + \frac{dr^m_{J2G+1}}{dr^m_{J2G}} \tag{3.145}$$

$$b_{C,in}(k) = -\frac{dr^m_{J2G+1}}{dr^m_{J2G}} \tag{3.146}$$

$$c_{C,in}(k) = 0 \tag{3.147}$$

with $C = D, l, \epsilon^d$ and also for $C = n$, whereas for m:

$$a_{m,B}(k) = 1 + \frac{dr_{J2G}}{dr_{J2G+1}} \tag{3.148}$$

$$b_{m,B}(k) = -\frac{dr_{J2G}}{dr_{J2G+1}} \tag{3.149}$$

$$c_{m,B}(k) = 0. \tag{3.150}$$

All these boundary types can also be applied to the primitive variables by taking into account the transformation between conservative and primitive variables. For example the linear extrapolation for the primitive variable U, where $m = \sqrt{g^{MR}_{rr}}\,\overline{\overline{\mathrm{D}}}^{MR}\,U$, at the inner radial boundary looks like:

$$a_{m,in}(k) = \left(1 + \frac{dr_{J2G}}{dr_{J2G+1}}\right) \frac{\sqrt{g^{MR}_{rr}}_{J2G+1}\,\overline{\overline{\mathrm{D}}}^{MR}_{J2G+1}}{\sqrt{g^{MR}_{rr}}_{J2G}\,\overline{\overline{\mathrm{D}}}^{MR}_{J2G}} \tag{3.151}$$

$$b_{m,in}(k) = -\frac{dr_{J2G}}{dr_{J2G+1}} \frac{\sqrt{g^{MR}_{rr}}_{J2G+1}\,\overline{\overline{\mathrm{D}}}^{MR}_{J2G+1}}{\sqrt{g^{MR}_{rr}}_{J2G-1}\,\overline{\overline{\mathrm{D}}}^{MR}_{J2G-1}} \tag{3.152}$$

$$c_{m,in}(k) = 0 \tag{3.153}$$

This shows that by setting up these functions it is quite simple to setup further easily constructed boundary conditions without bothering about the change of the solution matrix.

Explicit and implicit methods:

Depending on the choice of the preconditioner $\tilde{A}$ a variety of solution methods can be constructed that range from purely explicit to fully implicit, depending on how similar the preconditioner is to the real Jacobian.

- **Explicit Method**
  (M5 in figure 3.4.2):

$$\tilde{A} = \frac{I}{\Delta t} = \frac{1}{\Delta t} \begin{pmatrix} 1 & & & & & & \\ & 1 & & & & & \\ & & 1 & & & & \\ & & & 1 & & & \\ & & & & 1 & & \\ & & & & & 1 & \\ & & & & & & 1 \end{pmatrix} \tag{3.154}$$

  If one uses $1/\Delta t$ times identity matrix I as the preconditioner, which is similar to the real Jacobian J for sufficiently small Δt, and sets the Crank-Nicolson number $\vartheta = 0$ one gets following matrix equation:

$$\frac{I}{\Delta t}\,\delta\vec{q} = -\vec{R}^{i=0} = R\vec{H}S^{n}, \tag{3.155}$$

  where $-\vec{R}^{i=0} = R\vec{H}S^{n}$ since $\vec{q}^{i=0} = \vec{q}^{n}$ and therefore $\Delta\vec{q}^{i=0} = \vec{q}^{i=0} - \vec{q}^{n} = 0$. In this case, where only the diagonal elements of the matrix are filled, the matrix equation is solved immediately after one iteration (then the residuum $\vec{R}^{i=1} = 0$), so essentially the matrix equation is solved already and the result (using $\delta\vec{q} = \delta\vec{q}^{i=0} = \vec{q}^{i+1=1} - \vec{q}^{i=0} = \vec{q}^{n+1} - \vec{q}^{n}$) is the classical explicit method:

$$\vec{q}^{n+1} = \vec{q}^{n} - \Delta t\,\vec{R}^{i=0} = \vec{q}^{n} + \Delta t\,R\vec{H}S^{n} \tag{3.156}$$

  This method is only numerically stable for CFL (Courant-Friedrich-Lewy) numbers smaller than unity or even smaller than 0.5 depending on the advection scheme used.

- **Semi-Explicit Method**
  (M4 in figure 3.4.2):

$$\tilde{A} = (\frac{1}{\Delta t} + d^{r} + d^{\theta})\,I = \begin{pmatrix} d & & & & & & \\ & d & & & & & \\ & & d & & & & \\ & & & d & & & \\ & & & & d & & \\ & & & & & d & \end{pmatrix} \tag{3.157}$$

  The semi-explicit method is obtained by only preserving the diagonal elements $d = \frac{\partial R_i}{\partial q_i}$ of the block diagonal matrices $\mathcal{D}_{\text{mod}}$ This method may be numerically stable even for CFL > 1, but stability is, depending on the problem, not always guarantied. It is absolutely stable if the flow is viscous dominated. The solution is again like in the explicit method trivial, since one can directly specify the inverse of the matrix: $\tilde{J}^{-}1 = \frac{1}{(\frac{1}{\Delta t} + d^{r} + d^{\theta})}\,I$.

- **Semi-Implicit Method**
  (M3 in figure 3.4.2):

$$\tilde{A} = \mathcal{D}_{\text{mod}} = \begin{pmatrix} \mathcal{D} & & & & & \\ & \mathcal{D} & & & & \\ & & \mathcal{D} & & & \\ & & & \mathcal{D} & & \\ & & & & \mathcal{D} & \\ & & & & & \mathcal{D} \end{pmatrix} \tag{3.158}$$

  For the semi-implicit method only the block diagonal matrices are used, neglecting sub and super diagonal block matrices

$$\mathcal{D}_{\text{mod}}\, \delta q^i_{j,k} = -\vec{R}^i_{j,k} \tag{3.159}$$

  This method is stable even with CFL $\gg 1$.
  But there might be problems due to the staggered grid discretization, since e.g. only the coupling term between m and D at one cell side is taken into account.
  The solution is much simpler to obtain than that of the fully implicit method.

- **Fully-Implicit Method**:

$$\tilde{A} = J = \begin{pmatrix} \mathcal{D} & \overline{S} & & & \overline{S} & \\ \underline{S} & \mathcal{D} & \overline{S} & & & \overline{S} \\ & \underline{S} & \mathcal{D} & \overline{S} & & \\ & & \underline{S} & \mathcal{D} & \overline{S} & \\ \underline{S} & & & \underline{S} & \mathcal{D} & \overline{S} \\ & \underline{S} & & & \underline{S} & \mathcal{D} \end{pmatrix} \tag{3.160}$$

  For fully implicit methods one is retaining all block matrices, which gives still a highly sparse global matrix, but has in case of a two-dimensional grid two fringes, for a three-dimensional grid there will be even three fringes, which drastically increases the bandwidth and therefore the solution cost. Therefore in most cases the matrix is simplified for computation or special solution methods, like the Krylov subspace iterative methods, which basically only rely on the matrix vector product, are used. The fully implicit method is the only unconditionally stable method.
  In particular implicit methods are also good suited for highly stretched grid distributions and the multi-scale nature of astrophysical flows.
  Usually one has more than one equation to solve: there is the possibility to solve each equation separately one after the other, the so called implicit operator splitting (IOS) approach (M2 in figure 3.4.2), where the order of the equations can be crucial, or many possibilities to couple several equations together.
  Due to the non-linear nature of the equations one has to iterate in any case to resolve the non-linearities, since the matrix equation is only linear!
  The resulting matrix is highly sparse (M1 in figure 3.4.2), so one can use semi-direct iterative splitting methods like the 'Approximate Factorization Method' (AFM: Warming and Beam 1979) or the 'Line Gauss-Seidel Relaxation Method' (LGS: MacCormack and Candler 1989) as efficient methods for solving the set of radiative MHD-equations within the context of the

defect-correction iteration method (see Hujeirat 2005a, and the references therein).
Furthermore the Krylov subspace iterative methods (KSIMs), which are projection methods, may be more efficient and robust than the above-mentioned semi-direct methods, especially in the case of large systems, where one uses parallelized simulation codes to solve the sparse matrix equations, since the KSIMs are better suited for parallelization.

With the degree of implicitness also the computational cost for one time step increases, because the calculation of the Jacobian and the solution gets more expensive, but on the other side one gets better stability and can use larger time steps.
So - depending on the particular problem - one should carefully choose the suited method,
e.g. for a turbulent flow, where one wants to resolve turbulence, explicit methods are the right choice, because this flows are highly time-dependent and one has to choose a small time step anyway.
Whereas if one is interested in stationary or quasi-stationary solutions, the implicit method is superior. For that purpose one best uses the so called Hierarchical Solution Scenario (HSS), which is described in the next section.
To accelerate the search for a stationary or quasi-stationary solution, one can also use a method called "**Residual Smoothing Method**" (Hujeirat 2005a), which is also called **local time-stepping**: For this method not a common global time step size is used, instead a time step size associated with the local CFL-number at each grid point is used to advance the solution. In this procedure one gets a not physically meaningful time-evolution of intermediate solutions.
If one wants to simulate time-dependent features of quasi-stationary solutions, one can use the obtained quasi-stationary solution as initial configuration and re-start the calculation using a common and physically relevant time step.

The explicit method is severely limited to Euler-type flows, whereas sophisticated treatment of most flow-problems in AFD (see regime of applications in picture 3.2) requires the employment of much more robust methods: the implicit methods.

3.4.3 Hierarchical Solution Scenario (HSS)

The Hierarchical Solution Scenario (HSS) is a multi-stage solution procedure (solver with maximum flexibility) consisting of following possible steps:

I implicit operator splitting (IOS)
sequentially solving the equations, order of equations is important for convergence;
for vortex-free compressible viscous and time-dependent flows

II HD as a single coupled system followed by that of the magneto component,
block sequential solution of equations;
high spatial and temporal accuracies in combination with the prolongation/restriction strategy may be used.

III fully coupled set of MHD equations with zero moment of the radiation field
(using pre-conditioned KSIMs) to obtain final steady state solution

IV solution of the internal energy equation weakly coupled with the 5D radiative transfer equations

With the hierarchical solution scenario one has an effective way to determine a quasi-stationary solution.

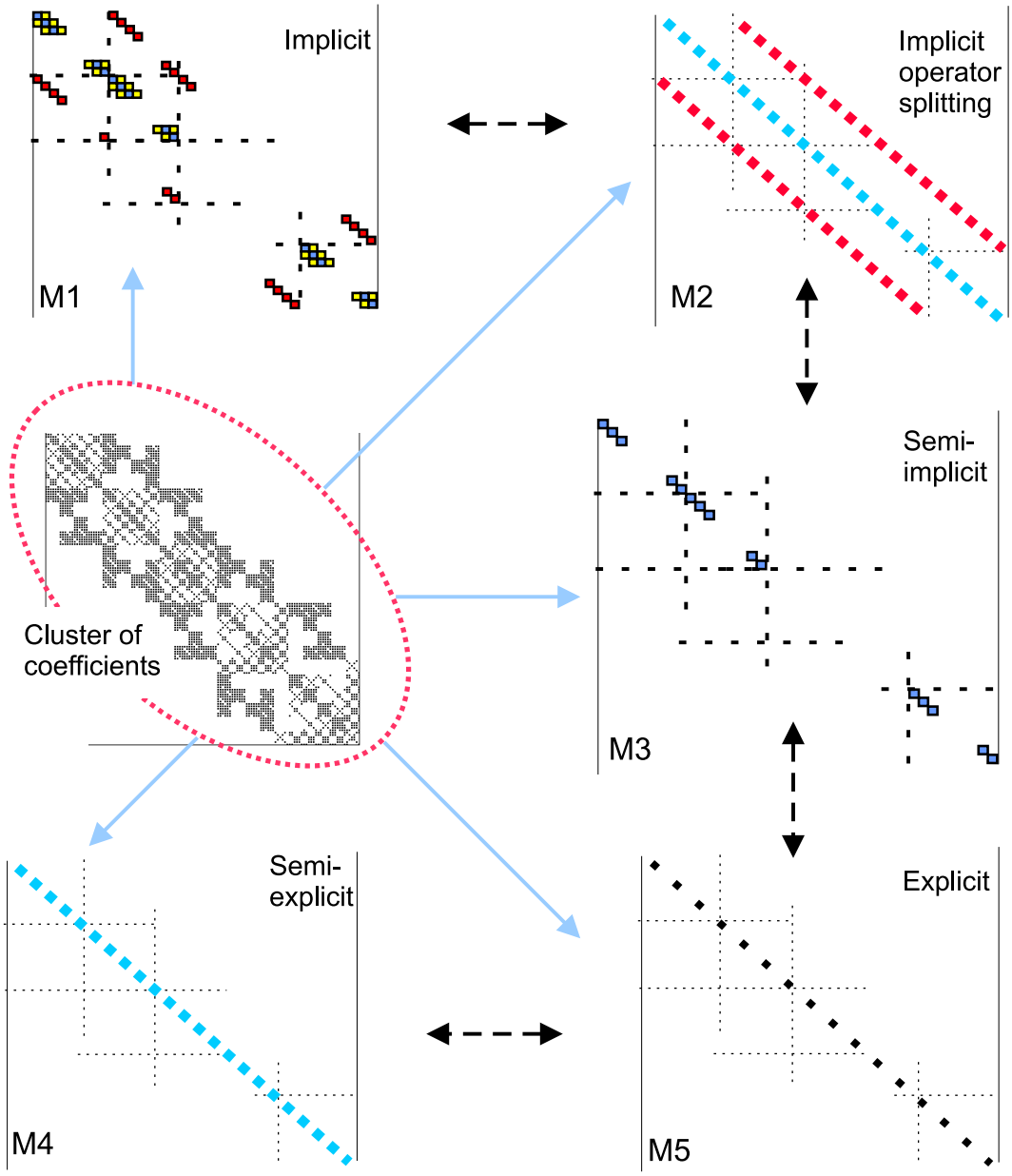


Figure 3.11: *A schematic description of the clustering of the coefficient matrix. The matrix-generator calculates only the entries to be used for the construction of the matrix appropriate for the selected solution procedure. In Astro-GRIPS these matrix entries are directly filled into the matrix structure of the selected solution method, so no extra copying process (as was used in the former simulation codes: GR-I-RMHD and before) is necessary anymore. Depending on the matrix used, the solution method may range from purely explicit to fully implicit (plot from Hujeirat et al. 2008).*

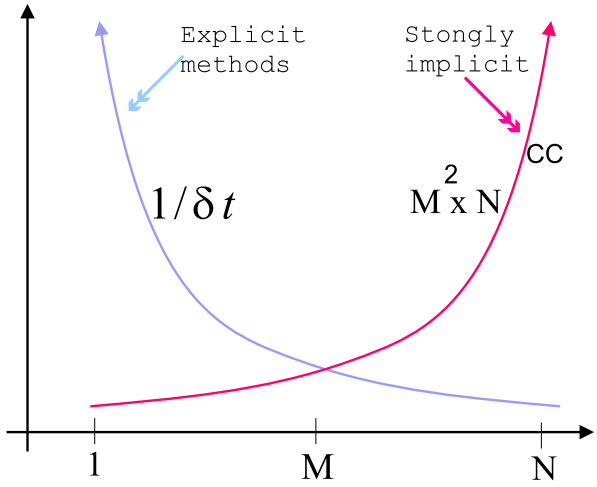


Figure 3.12: *A schematic description of the time step size and the computational costs versus the band width M of the Jacobian. N is the number of unknowns. Explicit methods correspond to $M = 1$ and large $1/\delta t$. They require minimum computational costs (CC). Large time steps (i.e., small $1/\delta t$) can be achieved using strongly implicit methods. These methods generally rely on the solution of large linear systems with matrices with large band width, hence computationally expensive, and, in most cases, are inefficient (plot from Hujeirat et al. 2008).*

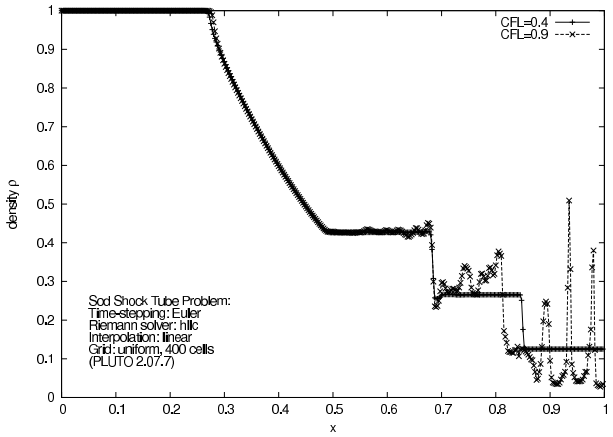


Figure 3.13: *The profile of the shock tube problem obtained with Courant-Friedrichs-Lewy numbers CFL=0.4 and 0.9 using the PLUTO code. Although both CFL-numbers are smaller than unity the numerical solution procedure does not appear to be stable even with CFL=0.9.*

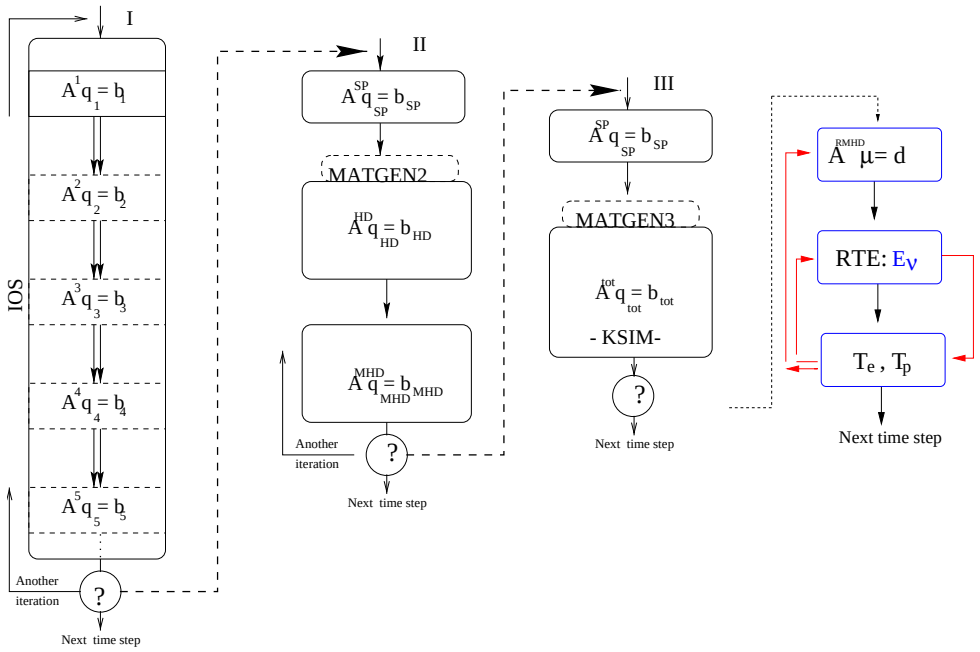


Figure 3.14: *A schematic description of the Hierarchical Solution Scenario (HSS) for finding a quasi-stationary state of the fluid motions described by the radiative MHD equations.*
The abbreviation SP stands for system of parabolic equations (e.g. spherical potential (self-gravity) for Newtonian version of the code)). The angular momentum equation may also be solved separately for axi-symmetric flows. In stage I, where the flow is in its early time-dependent phase, the equations are solved sequentially using the implicit operator splitting approach (IOS). Then in stage II, which uses the solution of stage I as initial condition, the hydrodynamics equations are solved as a single coupled system, followed by the coupled magneto component equations (the induction equation). In this process high spatial and temporal accuracies in combination with the prolongation/restriction strategy may be used. The solution is then used as starting point of stage III, where (quasi-) steady solutions for the fully coupled set of equations consisting of the zero moment of the radiation field and the MHD equations are sought. For this stage pre-conditioned Krylov sub-iterative methods are considered to be robust and efficient. In the final stage IV, one seeks for the solution of the fully coupled magneto-hydrodynamic system and the weakly to the internal energy equation coupled frequency dependent radiative transfer equation (RTE), the so-called 5D radiative transfer equation, for multi-component fluids, where the electrons and ions have different temperatures (plot from Hujeirat et al. 2008).

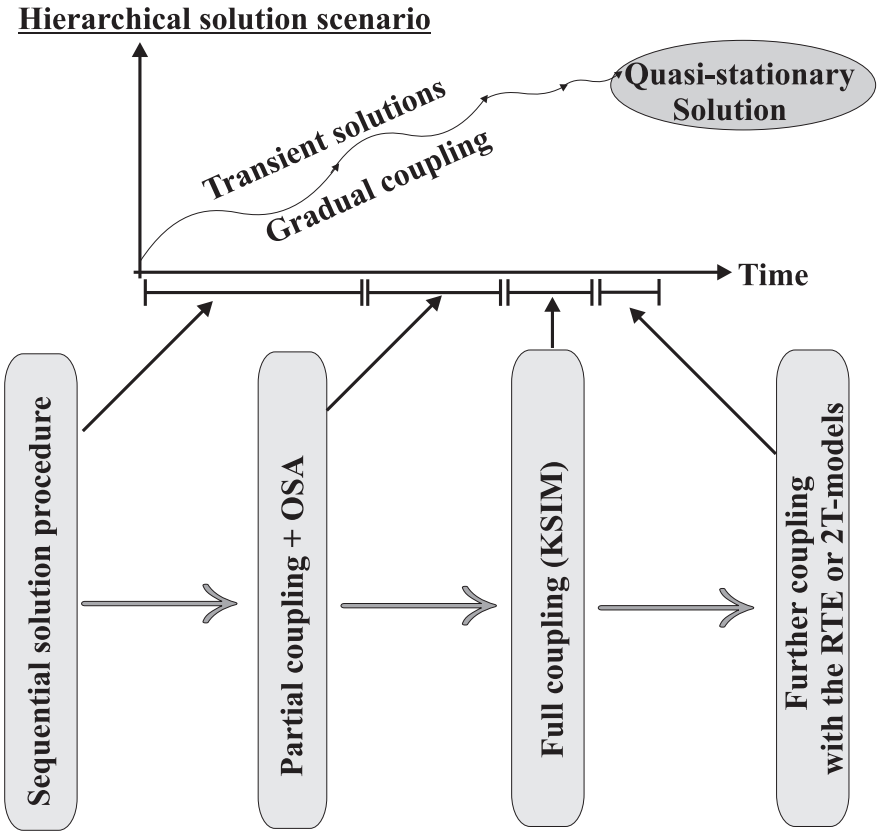


Figure 3.15: *With the hierarchical solution scenario (HSS) one effectively can determine a quasi stationary solution by gradual coupling of the equations. By dynamically varying the efficiency and robustness of the numerical method it is possible to leapfrog the transient phase very fast. This method is most suitable for searching quasi-stationary flow-configurations that depend only weakly on the initial conditions. The coupling between the equations is enhanced gradually, by starting solving them sequentially, then by partially coupling in combination with the operator splitting approach (OSA), full-coupling using the Krylov-subspace iterative methods (KSIM) and finally extending the coupling to include the radiative transfer equation (RTE) and energy equation of multi-temperature plasmas (plot from Hujeirat et al. 2007).*

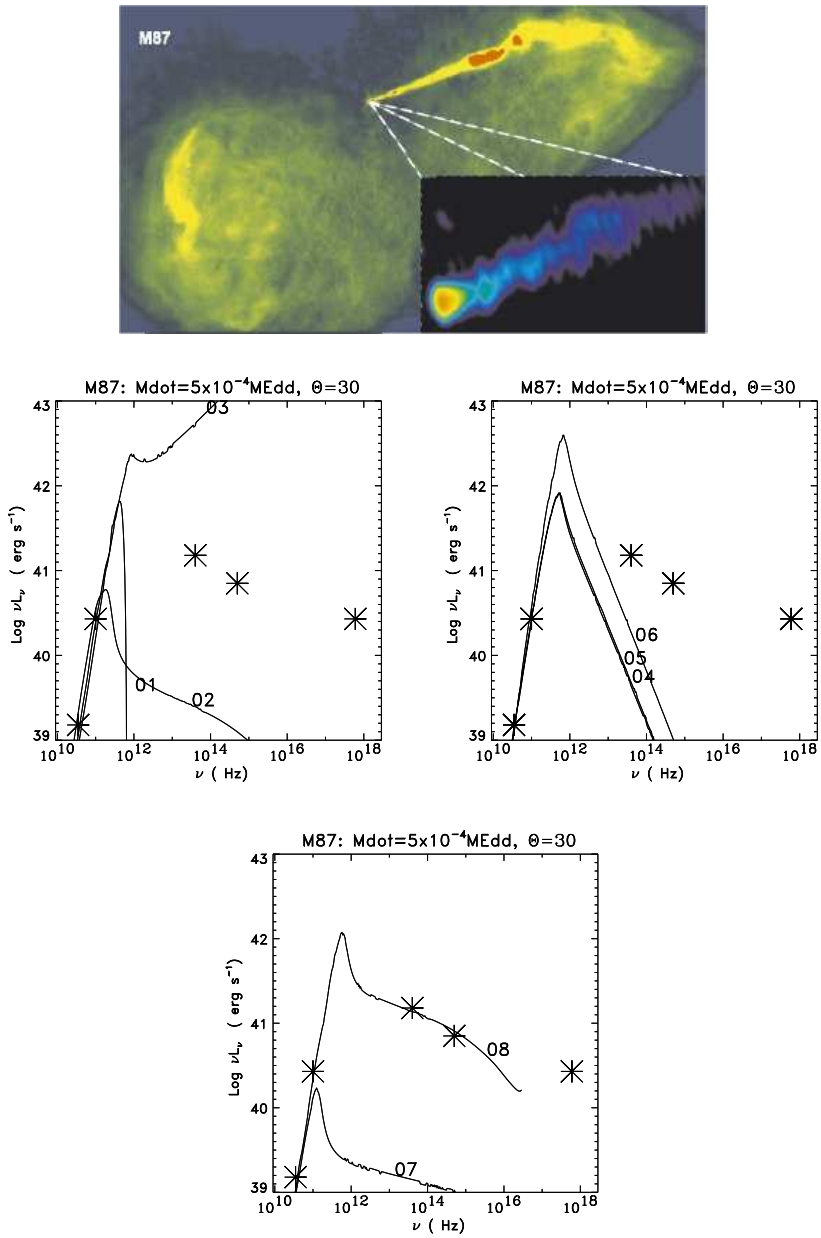


Figure 3.16: *Possible application of the Hierarchical Solution Scenario (HSS) demonstrated in Hujeirat (2005a): VLA data from the active central engine of the giant elliptical galaxy M87 and a NRAO radio image of the jet apparently emanating from within 100 gravitational radii. Solid lines correspond to calculated profiles and the asterisks to observational data. 'The profiles 01 to 06 show the spectral energy distribution calculated using different magnetic field strengths, or different truncation radii, or high/low corona temperatures. In particular, the profile 07 corresponds to a model in which the toroidal magnetic field is set to vanish artificially, whereas the poloidal magnetic field is set to be in equipartition with the thermal energy of the electrons. The profile 08 is similar to 07, except that the toroidal magnetic field is allowed to develop and reach values beyond equipartition with respect to the thermal energy of the electrons in the transition layer between the disk and the overlying corona. The above spectral energy distribution has been obtained by solving the radiative transfer equation in 5-dimensions, taking into account the Kompaneets operator for consistently modelling Comptonization. 400 non-linearly distributed frequency points have been used to cover the frequency-space, and* 125×40 *finite volume cells to cover the spatial domain of the calculation.' (plots and description from Hujeirat 2005a)*

3.5 Iterative Linear Equation Solvers

In this section solution methods for large sparse linear systems are described.

Sometimes in literature the solution of a system of linear equations is called misleadingly Inversion Procedure: In Numerics usually no one calculates the inverse of the complicated matrix and then determines the solution vector, since this is extremely costly. Instead the solution vector is determined directly or in most cases iteratively, since one is not interested in the inverse matrix.
For very large systems, where memory problems are a concern, it may be possible, that the fully occupied matrix would not fit into memory. Especially in this case the construction of the inverse matrix, which can be fully occupied even if the original matrix is highly sparse, or the use of a direct solution method is impossible. In these cases iterative methods are the the only possible procedures to determine the solution with a certain accuracy.

To have a stable solution procedure the matrix A must be strictly diagonally dominant, which means that the entries in each row of the matrix A must fulfil the following condition: the module of the diagonal element $d_{i,i}$ is larger than the sum of all moduli of the off-diagonal elements $\sum_{\mathrm{j}\neq\mathrm{i}} |a_{i,j}|$, where i and j denote the row and column numbers of the matrix.

Although the original solution matrix $A = J$, the Jacobian, is a sparse matrix with only a few non-zero matrix entries, the fringes due to the second spatial direction increase the bandwidth m of the solution matrix dramatically, which result in a usually very expensive numerical solution.

Due to this fact, there exist approximation methods, that use solution matrices without fringes. To these methods belong the Line Gauß-Seidel Method (LGS) (MacCormack and Candler 1989) and the Approximate Factorization Method (AFM) (Warming and Beam 1979).

Another class of iterative solution methods, the so-called Krylov Subspace Iterative Methods, only rely on matrix-vector products and maximum and minimum determination and are therefore - especially in the parallel case - very efficient. The matrix-vector products are not very expensive, even if the matrix has a large bandwidth m due to the existence of fringes, because there are also many zero-elements between the matrix blocks of the fringes and the tridiagonal block structure on the diagonal of the matrix. With appropriate sparse matrix storage formats, which are ideally only storing the non-zero matrix entries, the matrix calculations can be done very efficiently.

3.5.1 Black-White Line Gauß-Seidel Method (BW-LGSR2)

The Black-White Line Gauß-Seidel Method (BW-LGS), also called zebra Gauß-Seidel Method, belongs together with the red-black LGS method, which uses a chess board distribution, to the family of multicolour schemes, which were constructed as parallel-effective methods from the not for parallelization suited classical Gauß-Seidel method.

The Black-White Line Gauß-Seidel Method (BW-LGS) uses a two stage process to update the state values on the numerical grid. Therefore the grid is divided into rows with odd and even k indices in θ-direction for the LGSR2 method or into columns with odd and even j indices in r-direction for the LGSθ2(=LGSZ2) method.

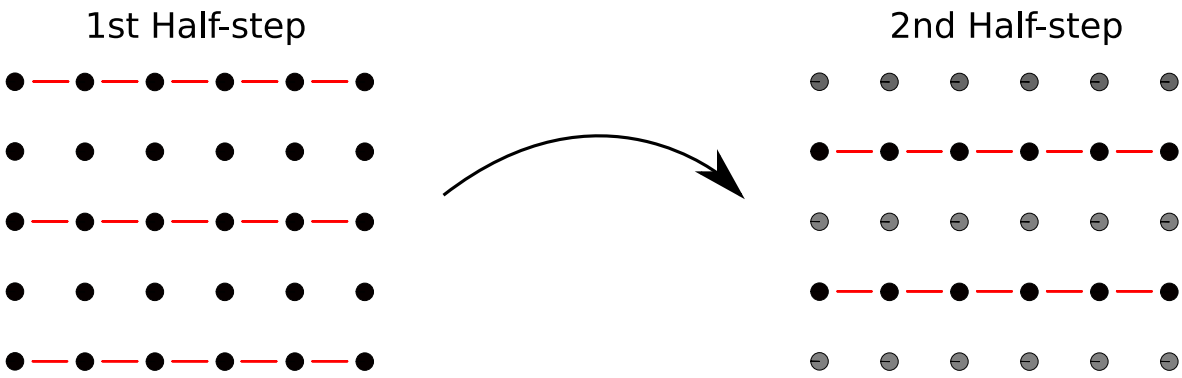


Figure 3.17: *Scheme of the Line-Gauss Seidel method (plot from Hilscher 2009)*

Remember the original block matrix structure:

$$\begin{matrix} & \overline{S}^{\theta}\delta q^{i}_{j,k+1} & \\ +\underline{S}^{r}\delta q^{i}_{j-1,k} & +\mathcal{D}_{\mathrm{mod}}\delta q^{i}_{j,k} & +\overline{S}^{r}\delta q_{j+1,k} \\ & +\underline{S}^{\theta}\delta q^{i}_{j,k-1}, & \end{matrix} = -R^{i}_{j,k} \tag{3.161}$$

with $\mathcal{D}_{\mathrm{mod}} = \frac{1}{\Delta t} + \mathcal{D}^{r} + \mathcal{D}^{\theta}$.

The two stages of the LGSR2 method are (see Fig. 3.17):

- In the first half-step only the grid points on rows with odd k are taken into account. So the Jacobian entries resulting from the neighbouring cells in θ-direction, from grid cells with even k, can be brought to the other side of the matrix equation. Note that in the first iteration step these entries are zero. So one gets the following tri-diagonal block matrix equation with - due to taking only odd rows into account - only (approximately) half the size of the original system:

$$\frac{\delta\vec{q}_{j,k_{odd}}}{\Delta t} + \underline{S}^{r}\,\delta\vec{q}_{j-1,k_{odd}} + \left(\mathcal{D}^{r} + \mathcal{D}^{\theta}\right)\,\delta\vec{q}_{j,k_{odd}} + \overline{\mathbf{S}}^{r}\,\delta\vec{q}_{j+1,k_{odd}} = -\vec{R}_{j,k_{odd}} - \overline{S}^{\theta}\,\delta\vec{q}_{j,k_{odd}+1} - \underline{S}^{\theta}\,\delta\vec{q}_{j,k_{odd}-1} \tag{3.162}$$

  or using $\mathcal{D}_{\mathrm{mod}} = \frac{1}{\Delta t} + \mathcal{D}^{r} + \mathcal{D}^{\theta}$

$$\underline{S}^{r}\,\delta\vec{q}_{j-1,k_{odd}} + \mathcal{D}_{\mathrm{mod}}\,\delta\vec{q}_{j,k_{odd}} + \overline{\mathbf{S}}^{r}\,\delta\vec{q}_{j+1,k_{odd}} = -\vec{R}_{j,k_{odd}} - \overline{S}^{\theta}\,\delta\vec{q}_{j,k_{odd}+1} - \underline{S}^{\theta}\,\delta\vec{q}_{j,k_{odd}-1}. \tag{3.163}$$

  This matrix equation is solved for $\delta\vec{q}_{j,k_{odd}}$ and then all state variables with odd k are updated.

- In the second half-step the grid points on rows with even k are taken into account. Now the Jacobian entries resulting from the variables on the grid cells with odd k are brought to the other side of the matrix equation. But this time the updated values of $\delta\vec{q}_{j,k_{odd}}$ are known from the first half step. The resulting tri-diagonal block matrix equation is:

$$\frac{\delta\vec{q}_{j,k_{even}}}{\Delta t} + \underline{S}^{r}\,\delta\vec{q}_{j-1,k_{even}} + \left(\mathcal{D}^{r} + \mathcal{D}^{\theta}\right)\,\delta\vec{q}_{j,k_{even}} + \overline{\mathbf{S}}^{r}\,\delta\vec{q}_{j+1,k_{even}} = -\vec{R}_{j,k_{even}} - \overline{S}^{\theta}\,\delta\vec{q}_{j,k_{even}+1} - \underline{S}^{\theta}\,\delta\vec{q}_{j,k_{even}-1} \tag{3.164}$$

  or using $\mathcal{D}_{\mathrm{mod}} = \frac{1}{\Delta t} + \mathcal{D}^{r} + \mathcal{D}^{\theta}$

$$\underline{S}^{r}\,\delta\vec{q}_{j-1,k_{even}} + \mathcal{D}_{\mathrm{mod}}\,\delta\vec{q}_{j,k_{even}} + \overline{\mathbf{S}}^{r}\,\delta\vec{q}_{j+1,k_{even}} = -\vec{R}_{j,k_{even}} - \overline{S}^{\theta}\,\delta\vec{q}_{j,k_{even}+1} - \underline{S}^{\theta}\,\delta\vec{q}_{j,k_{even}-1}. \tag{3.165}$$

  This matrix equation is solved for $\delta\vec{q}_{j,k_{even}}$ and then all state variables with even k are updated. Now for all grid cells the dependent variables are calculated.

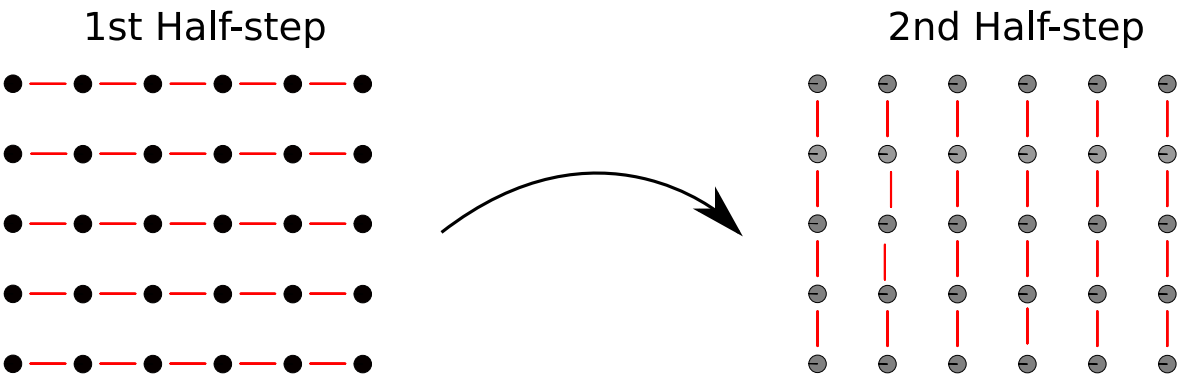


Figure 3.18: *Scheme of the approximate factorization method (plot from Hilscher 2009).*

Following the defect correction strategy one can now do a further iteration to improve the solution accuracy.

This method has a asymmetry between the directions, because the equations are solved fully coupled in the r-direction, whereas the θ-terms of the Jacobian only occur on the right hand side of the matrix equations.

This directional asymmetry is avoided in the symmetric LGS method, where the preferred direction is switched every time step (or iteration step). But due to the reordering of the matrix entries this symmetric method is - in contrast to the normal BW-LGS method with one preferred direction - not very efficient and not well-suited for parallelization.

3.5.2 Approximate Factorization Method (AFM)

The Approximate Factorization Method (AFM) uses a factorization technique to subdivide the problem into two single directional solution equations.
The original solution matrix for a 2 dimensional grid can be approximated by a product of two matrices:

$$A\,\delta\vec{q} \approx \tilde{A}_r \tilde{A}_\theta\,\delta\vec{q} \tag{3.166}$$

where $\tilde{A}_r = \frac{I}{\Delta t} + J_r$ and $\tilde{A}_\theta = (\frac{I}{\Delta t} + J_\theta)\,\Delta t = I + J_\theta\,\Delta t$, where J_r and J_θ contain only matrix derivatives in the r or θ-direction respectively. Inserting these matrix expressions gives:

$$\tilde{A}_r \tilde{A}_\theta\,\delta\vec{q} = (\frac{I}{\Delta t} + J_r)\cdot(I + J_\theta\,\Delta t) = \frac{I}{\Delta t} + J_r + J_\theta + J_r\,J_\theta\,\Delta t = A + J_r\,J_\theta\,\Delta t \approx A \tag{3.167}$$

so it is assumed here that $J_r\,J_\theta\,\Delta t$ is small, which is certainly true if Δt is small and if steady conserved fluxes are considered, where boundary conditions are time-independent. Elsewise for time-dependent simulations using large time steps this term might be large, which would result in a divergence of this method.

The two half-steps of the Approximate Factorization Methods are the following (see also figure 3.18):

- In the first step one solves for the r-terms first by introducing an intermediate solution vector defined by $\delta\vec{q} \doteq \tilde{A}_\theta \, \delta\vec{q}$:

$$\tilde{A}_r \, \delta\vec{q} = -\vec{R} \tag{3.168}$$

$$(\frac{I}{\Delta t} + J_r) \, \delta\vec{q} = -\vec{R} \tag{3.169}$$

$$\frac{\delta\vec{q}_{j,k}}{\delta t} + \underline{S}^r \, \delta\vec{q}_{j-1,k} + \mathcal{D}^r \, \delta\vec{q}_{j,k} + \overline{S}^r \, \delta\vec{q}_{j+1,k} = -\vec{R}_{j,k} \tag{3.170}$$

- In the second step, knowing now the intermediate solution vector $\delta\vec{q}$, the remaining matrix solution system is solved:

$$\frac{I}{\Delta t} \, \tilde{A}_\theta \, \delta\vec{q} = \frac{I}{\Delta t} \, \delta\vec{q} \tag{3.171}$$

$$(\frac{I}{\Delta t} + J_\theta) \, \delta\vec{q} = \frac{\delta\vec{q}}{\Delta t} \tag{3.172}$$

$$\frac{\delta\vec{q}_{j,k}}{\Delta t} + \underline{S}^\theta \, \delta\vec{q}_{j,k-1} + \mathcal{D}^\theta \, \delta\vec{q}_{j,k} + \overline{S}^\theta \, \delta\vec{q}_{j,k+1} = \frac{\delta\vec{q}}{\Delta t} \tag{3.173}$$

  With the obtained solution vector $\delta\vec{q}$ the state vector $\vec{q}$ is updated.

If the newly calculated defect, the residuum R, is still too large, then this procedure is iterated in the concept of the defect-correction strategy until one gets to the desired accuracy.

Note that here for the matrix solver to work, the intermediate solution vector has to be reordered with respect to the order of the indices j and k between the two steps.

In both steps only a tri-diagonal block matrix is occurring in the matrix equation, which can be efficiently solved with a cost of the order of $O(m^2 N)$ by using either the Thomas algorithm or an iterative algorithm like Jacobi.

3.5.3 Krylov Subspace Iterative Methods

A good overview over several direct and iterative methods for the solution of linear systems of equations can be found in Meister (2005).

There are **two major classes of iterative methods** to solve large sparse linear systems of equations: the **splitting methods** and the **projection methods**.

Splitting Methods

The splitting methods are based on the splitting of the matrix A in the form:

$$A = B + (A - B) \tag{3.174}$$

so that $Ax = b$ gives:

$$Bx = (B - A)x + b \tag{3.175}$$

and if B is regular, one gets:

$$x = B^{-1}(B - A)x + B^{-1}b \tag{3.176}$$

and the linear iteration method can be defined by:

$$x_{i+1} = \Phi(x_m, b) = Mx_i + Nb \quad \text{for } i = 0, 1, \ldots \tag{3.177}$$

with

$$M := B^{-1}(B - A) \tag{3.178}$$

and

$$N := B^{-1}. \tag{3.179}$$

The matrix B should be chosen so that it is similar to A but easy to invert.

Depending on the choice of B one gets different splitting methods:
For $B = I$, where B is the unity matrix I, one gets the trivial method.
If one sets B to the diagonal part D of the matrix A, $B = D$, the Jacobi method is obtained.
The Gauß-Seidel method can be constructed by setting $B = D + L$, where D is the diagonal part and L is the strict lower triangular part of the matrix A. Compared to the Jacobi method, the Gauß-Seidel method uses with $D + L$ a better approximation of the matrix A, so a smaller spectral radius, the radius in which the eigenvalues are contained, of the iteration matrix and therefore a faster convergence can be expected.
There also exist so called relaxation methods, which introduce a correction factor, the relaxation parameter, to the above mentioned methods to try to reduce the spectral radius of the iteration matrix to get faster convergence. For example, the SOR method (successive over-relaxation method) belongs to this class of methods.

The convergence behaviour and speed depends in general on the spectral radius and the condition number of a matrix, which is described later.

Projection Methods:

Another class of iterative methods to solve large sparse linear systems of equations are the projection methods.

A projection method, to solve the matrix equation $Ax = b$, is a method, where approximate solutions $x_i \in x_0 + K_i$ ($\dim K_i = i \leq n$) are calculated according to following condition:

$$(b - Ax_i) \perp L_i \tag{3.180}$$

where $x_0 \in \mathbb{R}^n$ is arbitrary and K_i and L_i ($\dim L_i = i \leq n$) are subspaces of $\mathbb{R}^n$. The orthogonality property is described by the Euklidian scalar product:

$$x \perp y \quad \Leftrightarrow \quad (x, y)_2 = 0 \tag{3.181}$$

In the case where $K_i = L_i$ the residuum vector $r_i = b - Ax_i$ is perpendicular to K_i and one has an orthogonal projection method and Eq. 3.180 is called Galerkin condition. If $K_i \neq L_i$ one has an inclined or tilted projection method and Eq. 3.180 is called Petrov-Galerkin condition.

The **Krylov Subspace Iterative Methods** belong to the projection methods:
A Krylov Subspace Iterative Method is a projection method to solve the matrix equation $Ax = b$ for which K_i is the Krylov-Subspace:

$$K_i = K_i(A, r_0) = \text{span}\left\{r_0, Ar_0, \ldots, A^{i-1}r_0\right\} \tag{3.182}$$

with $r_0 = b - Ax_0$.

Krylov Subspace Iterative Methods can be described as a reformulation of the linear equation system in a minimalization problem. The conjugate gradient (CG) method and the GMRES method belong to this algorithmic group. Both methods determine the optimal approximation $x_i \in x_0 + K_i$ to the searched solution $A^{-1}b$ by using the orthogonality condition 3.180, whereas at each iteration the dimension of the subspace is incremented by one. If one would neglect rounding errors both methods would give the exact solution at the latest after n iterations.

There are many Krylov Subspace Iterative Methods which differ in the stability, convergence behaviour and speed and usability for different types of matrixes.

For symmetric, positive definite matrices:

- The combination of the method of steepest gradient, which has a bad convergence behaviour due to its not optimal construction of the search directions (with respect to the orthogonality to the whole subspace), and the method of conjugate directions, which can have an unsteady error behaviour, but does an optimal construction of the subspaces, leads to the Conjugate Gradient (**CG**) method by Hestenes and Stiefel, 1952, which combines the advantages and eliminates the disadvantages of the single methods. But this method is only applicable for symmetric positive definite matrices.

For non-symmetric, non-positive definite matrices several other methods were invented:

- In the Generalized Minimal Residual (**GMRES**) method (by Saad and Schultz, 1986) the function $F(x) = \|Ax - b\|_2^2$ is minimized. The Arnoldi algorithm is used to calculate the orthonormal basis of the Krylov subspace. This method requires the storage of the whole sequence of directional vectors, a large amount of memory is usually needed. Modified versions of this method, sometimes called GMRES(l), do a restart after l iterations. Through this technique, only a limited number of vectors have to be stored, but the performance of the original method is reduced and the convergence can be very slow. Choosing the best suited restart parameter is a difficult task, which can only be done by experience. GMRES can deal with large non-symmetric matrices and can easily also be implemented on vector computers because almost all arithmetic operations are matrix-vector multiplications, vector updates and inner products.
- The Bi-Conjugate Gradient (**BiCG**) method (by Lanczos, 1952, reformulated by Fletcher, 1975) is a Krylov Subspace Method based on the Bi-Lanczos algorithm, where the orthogonality is given by the Petrov-Galerkin condition: $L_i = K_m^T = \text{span}\{r_0, A^T r_0, \ldots, (A^T)^{i-1} r_0\}$. In this method one simultaneously looks at $Ax = b$ and $A^T x = b$. Compared to GMRES is uses much less memory, but the transpose of A is needed at each iteration, it has an irregular convergence behaviour and the method can break down before the exact solution is reached.
- The Conjugate Gradient Squared (**CGS**) method (by Sonneveld, 1989) improves the BiCG method by avoiding the multiplication by A^T.

- The BiConjugate Gradient Stabilized (**BiCGSTAB**) method (by van der Vorst, 1992) minimizes the oscillations of the CGS method by the one-dimensional minimalization of the residuum.
- The **BiCGSTAB(l)** method (by Sleijpen, Fokkema, 1993) is the variant that is extended to the l-dimensional minimalization of the residuum.
- The Quasi-Minimal Residual (**QMR**) method (by Freund and Nachtigall, 1991) minimizes the storage used by the GMRES method by quasi-minimalization.
- The Transpose Free Quasi-Minimal Residual (**TFQMR**) method (by Freund, 1993) improves the QMR method by avoiding the multiplication by A^T using ideas of the CGS method. The number of required iterations is similar to BiCGSTAB. For a well-conditioned problem, that needs only a few iterations to be solved, GMRES works better than TFQMR, but if a large number of iterations is necessary TFQMR performs better.
- The **QMRCGSTAB** method (by Chan et al., 1994) combines the ideas of the BiCGSTAB and the TFQMR methods.

Preconditioners

The convergence behaviour and speed of iterative methods, especially of the Krylov Subspace Iterative Methods, strongly depends on the spectral radius, which is the minimal radius which contains all eigenvalues, and the condition number of the matrix.

The bf condition number of the regular matrix $A \in \mathbb{C}^{nxn}$ with respect to the induced matrix norm $\|.\|_a$ is defined as:

$$\text{cond}_a(A) := \|A\|_a \, \|A^{-1}\|_a \tag{3.183}$$

If A is a normal matrix, then

$$\text{cond}_2(A) = \frac{|\lambda_n|}{|\lambda_1|} \tag{3.184}$$

where λ_n is the eigenvalue of the matrix A with the largest modulus and λ_1 the eigenvalue with the smallest modulus.
The condition number of a regular matrix is independent of the induced matrix norm limited downwards:

$$\text{cond}_a(A) \geq \text{cond}_a(I) = 1. \tag{3.185}$$

In praxis one uses the norm of the residuum vector $r_i = b - Ax_i$ to obtain the quality of the approximated numerical solution of an iterative method, since the error vector $e_i = A^{-1}b - x_i$ is unknown, because the real solution is not known. If the condition number of the solution matrix is small, it makes sense to use the convergence estimation based on the residuum. However, if the condition number is large, the error norm can increase dramatically, even if the residuum norm decreases.

So a small enough condition number of the solution matrix is mandatory for a good convergence behaviour of iterative solvers. Therefore one uses a technique called preconditioning.

To improve the convergence behaviour and rate the original linear matrix equation is transformed by so-called preconditioners into a system whose matrix possesses a better (reduced) condition number. The original system of equations:

$$Ax = b \tag{3.186}$$

is transformed into:

$$(P_L A P_R) x^P = P_L b \tag{3.187}$$
$$x = P_R x^P \tag{3.188}$$

where P_L is the left and P_R is the right preconditioner.
If setting $P_L = I$ one gets right preconditioning, which results in an unaltered residuum:

$$r \equiv b - Ax = b - AP_R P_R^{-1} x \tag{3.189}$$

But using the left preconditioning by setting $P_R = I$ can alter the residuum:

$$r_L \equiv P_L b - P_L A x = P_L (b - Ax) = P_L r \tag{3.190}$$

where, due to efficiency reasons, the matrix P_X, $X = R, L$ should be a good approximation of A^{-1}, the inverse of the original matrix A, so that $\text{cond}(P_X A) \ll \text{cond}(A)$, but still easy to construct.

Some preconditioners are:

- Jacobi,
- block Jacobi (different Krylov Subspace methods and preconditioners are applied on different blocks),
- Successive Over Relaxation (SOR),
- Incomplete Cholesky (only for symmetric matrices),
- Incomplete Lower Upper factorization (ILU),
- Additive Schwarz Method (ASM),
- Lower Upper factorization (LU),
- Cholesky method (only for symmetric and pos. definite matrices).

In praxis the Krylov Subspace Iterative methods used in combination with suitable preconditioners have showed that they are robust, stable and efficient methods for solutions of large sparse linear systems.

Darbandi et al. (2006) investigated the use of several different preconditioned Krylov subspace methods to implicitly solve the fully coupled set of incompressible Navier-Stokes equations using a finite volume discretization using PETSc (Balay et al. 2009). They found out in their extensive study that for these kind of fluid flow problems the best performance is achieved using GMRES with incomplete lower upper preconditioner.

4 Simulation Code Structure, Optimization and Parallelization of Astro-GRIPS

In this chapter the Simulation Code Structure, the basic code usage, the Optimization and the Parallelization of Astro-GRIPS, the General Relativistic Implicit Parallel Solver, is described.

4.1 Astro-GRIPS: Simulation Code Features

The simulation code Astro-GRIPS, the General Relativistic Implicit Parallel Solver, solves the Newtonian or general relativistic hydrodynamical equations on a 3D axi-symmetrical spherical grid in the background metric of a Schwarzschild or Kerr black hole using the finite volume discretization and solving the equations with implicit methods.

It is written in Fortran-90/95 and uses the message passing interface (MPI: Forum 2008, 2009; Gabriel et al. 2004) for parallelization on distributed memory machines.

Before compilation the Fortran code can be comfortably configured for different compilers, serial or parallel mode and other features (e.g. using netcdf or parallel-netcdf for data output) by a configure script using the GNU autotools system (Vaughan et al. 2000).

Initial conditions and special boundary conditions are set up in the file Setup.F90, which is usually the only source file the user has to adopt to solve a new problem.

With the use of a parameter file one can change many parameters quite easily without re-compiling the code.

By changing only the parameter SolMethod in the parameter file, one can select various degrees of implicitness, from pure explicit to fully implicit, the coupling of the equations and different iterative linear equation solvers, which are used by the implicit methods: Black-White Line-Gau"ss-Seidel, Approximate Factorization Method (which were implemented with the help of LAPACK (Anderson et al. 1999) or ScaLAPACK (Blackford et al. 1997) routines in the parallel case) and various Krylov Subspace Iterative Methods (using PETSc: Balay et al. 2009, 2008, 1997). This is a quite easy but powerful way to investigate the behaviour of different methods.

Using the Hierarchical Solution Scenario (HSS) it is possible to find quasi-stationary solutions more quickly by gradual enhancement of the coupling of the equations.

As data output and input format netCDF (Rew et al. 2009), a widely used portable binary format, is used via the netcdf library in the serial or parallel case or via the parallel-netCDF library (Li et al. 2003) to do a real parallel portable input and output of data based on MPI-I/O.
NetCDF files can be read, manipulated and visualized by many different programs, for example with ncview or with python using matplotlib (Dale et al. 2009).

One important feature of the code is the possibility to do a **restart**. Every time a netcdf data file output is done a restart can be made from it. Since there is only one netcdf data file created, and not for each process one, it is also very easy to do a restart with another number of processes and because netcdf is a portable file format, the restart can even be done easily on another architecture or computer platform (no transformation from big-endian to little-endian byte ordering or vice-versa is necessary as one might need for simple binary output).

Another interesting feature is the **prolongation**, where the grid can be refined (or made coarser) at a specified simulated time or time iteration number and the data of the previous grid level is interpolated (or extrapolated) to the new grid points.
This can be used for example to do first very coarse simulation say on a single-processor machine, and then later continue the simulation by restarting on a computer cluster and then do a refinement of the grid to get a more detailed view of e.g. a stationary solution.

Astro-GRIPS was designed in such a way, that it is a flexible, extensible, stable and robust algorithm for general relativistic (M)HD flows, but still easy to use.

Summary of simulation code features:

Astro-GRIPS is an implicit solver for modelling:

- 3D axi-symmetric,
- strongly time-dependent,
- quasi-stationary and steady state,
- compressible,
- weakly compressible,
- dissipative and diffusive,
- Newtonian, special relativistic and
- general-relativistic flows (background metric of a Schwarzschild or Kerr black hole)

A non-linear Newton-iterative numerical tool for solving:

- Newtonian Euler,
- Newtonian Navier-Stokes,
- General Relativistic Euler,
- and General Relativistic Navier-Stokes equations.

Discretization method:

- 3D axi-symmetrical spherical grid
- Finite Volume Method using a conservative formulation (but with internal energy equation)
- First and second order temporal accuracy
- up to 3rd order spatial accuracy

Properties of the solution method:

- Various degrees of implicitness:
  implicit, semi-implicit, semi-explicit and explicit methods
- Hierarchical Solution Scenario:
  enables gradual coupling of the equations

Implicit iterative solution procedures (methods to solve the large sparse linear system of equations):

- Black-White Line Gauß-Seidel method (BW-LGS)
- Approximate Factorization Method (AFM)
- Krylov Subspace Iterative Methods (KSIM,KSP): GMRES, Bi-CGSTAB, TF-QMR, ...

Basic properties of the simulation code:

- Fortran 90/95
- parallelized using MPI
- use of LAPACK or ScaLAPACK routines for BW-LGS and AFM
- PETSc library for Krylov methods
- netCDF data I/O (serial and parallel-netCDF)
- visualization scripts using python/matplotlib

Although the program itself is quite complex, it will be clearly structured and easy to use, but still very powerful and flexible.

4.2 Simulation Code Structure of Astro-GRIPS

4.2.1 Basic code structure

The simulation code Astro-GRIPS is written in Fortran-90/95 and uses the message passing interface (MPI) for parallelization on distributed memory machines.

Before compilation the Fortran code can be comfortably configured for different compilers, serial or parallel mode and other features (e.g. using netcdf or parallel-netcdf for data output) by a configure script using the GNU autotools system.

The Fortran code is distributed in several source code files and structured in the following way:

The main file is called AstroGRIPS.F90 and defines the basic flow structure of the simulation.

- (MPI): Initialization of Parallel mode (including BLACS setup for ScaLAPACK)
- Setup of basic Parameters (from parameter file)
- Dynamical memory allocation of grid and geometry related variables
- Grid generation
- (MPI): Perform domain decomposition with initialization of halo/boundary values
- Dynamical memory allocation, setup and initialization of global and local data, log and auxiliary variables
- Calculation of grid and geometry related variables
- Setup initial conditions (primitive variables) or if restart: read in data values
- Compute conservative variables from primitives
- (MPI): Halo Communication (update halo data: so all conservative variables are set right also in halo cells, before the dependent/related variables are calculated
- compute all other dependent variables
- (MPI): Halo Communication (update halo data)
- Print out Settings and Solver Info
- (MPI:PETSc): initialize PETSc (has to be called after domain decomposition is done)
- calculate explicit time step (if no restart)
- setup output data and log netcdf files and write out initial data (if no restart)
- if prolongation directly after restart is requested: do prolongation
- Time-Iteration Loop:
  - calculate explicit time step
  - determine and set time step
  - Solve equations (dependent on SolMethod setting in parameter file)
  - Print information out to Terminal
  - Data and log output
  - if prolongation is requested: do prolongation

  - if end time or maximum time step is reached: leave time-iteration loop
- Finalization: free memory: deallocate variables; (MPI): finalize PETSc and MPI

4.2.2 The SolMethod parameter

The routine Solve analysis the parameter SolMethod and sets up the corresponding parameters and auxiliary variables and calls the appropriate solver routine(s): explicit, LGSR2, AFM, KSP, ...; with the specified parameters and equations.

The parameter SolMethod, set in the parameter file, specifies the equation coupling, the order and the methods and which solution parameters to use.

Since the parameter SolMethod can be easily changed in the parameter file the code has not to be recompiled to use another equation coupling or another iterative method or another solution parameter.

The SolMethod parameter is constructed in such a way, that it really enables the user to have a large and easy to use flexibility in selecting and trying out different solvers, different orders of the equations, different parameters or even solver combinations.

The parameter SolMethod is constructed by considering following rules:

```
! SolMethod "{SolMethod}"
!
!   attention:
!     put {SolMethod} in double quotes, since elsewise it is not read in
!     as a total string, since commas separate the string
!
! with
!
! {SolMethod} = {igroup1};{igroup2}; ... ,{igroupn}
!
!   {igroup} = {giter_max(igroup)}:{imethod1},{imethod2}, ... ,{imethodn}
!
!     {imethod} = {method(igroup,imethod}({ieqns})({opts})
!
!       {ieqns} = {eqngrp1}, {eqngrp2}, ... , {eqngrpn}
!         {eqngrp} = {eqns(igroup,imethod,ieqn=1}_{eqns(igroup,imethod,ieqn=2}
!           _ ... _{eqns(igroup,imethod,ieqn=Neqns(igroup,imethod)}
!
!       {opts} = {opt1}, {opt2}, ... , {optn}
!          {opt} = i={iter_max(igroup,imethod)}
!                = mf={KSP_MATRIX_FORMAT(igroup,imethod)}
!                = ksp_rtol={ksp_rtol(igroup,imethod)}
!                = acc={accuracy(igroup,imethod)}
```

```
!
!
!    Global equation numbers:
!    1 = relativistic density
!    2 = radial momentum
!    3 = vertical/latitudinal momentum
!    4 = angular momentum
!    5 = internal energy
!
!
! attention:
! SolMethod "2:AFM<2_3,1_5>(i=2)"
! is the same as:
! SolMethod "2:AFM<2_3>(i=2),AFM<1_5>(i=2)"
!
! and gives:
! Ngroup = 1
! giter_max(igroup=1) = 2
! Nmethod(igroup=1) = 2
!
! method(igroup=1,imethod=1) = AFM
! Neqns(igroup=1,imethod=1) = 2
! eqns(igroup=1,imethod=1,ieqn=1) = 2
! eqns(igroup=1,imethod=1,ieqn=2) = 3
! opts(igroup=1,imthod=1) = i=2, so iter_max(igroup=1,imethod=1) = 2
!
! method(igroup=1, imethod=2) = AFM
! Neqns(igroup=1,imethod=2) = 2
! eqns(igroup=1,imethod=2,ieqn=1) = 1
! eqns(igroup=1,imethod=2,ieqn=2) = 5
! opts(igroup=1,imthod=2) = i=2, so iter_max(igroup=1,imethod=2) = 2
```

The global equation numbers used in the simulation code and in the construction of the SolMethod parameter are the following:

1. relativistic density D
2. radial momentum m
3. vertical/latitudinal momentum n
4. angular momentum l
5. internal energy density ϵ^d

For example:

```
SolMethod "2:AFM<2_3,1_5>(i=2)"
```

is the same as:

```
SolMethod "2:AFM<2_3>(i=2),AFM<1_5>(i=2)"
```

and means, that there is one global group iteration loop which is iterated twice. There are 2 methods used in this global group, the first method uses AFM to solve equation 2, the radial momentum *m* equation,coupled together with equation 3, the latitudinal momentum *n* equation using two 'internal' iterations, the second method couples equations 1, the relativistic density *D* equation, and 5, the internal energy ϵ^d equation together which are also solved with AFM using two 'internal' iterations.

Similar to that many different solution methods can be considered, which can be tried out. Some might give a better result other a much worse.

Here another example:

```
SolMethod "5:AFM<4>,KSP*bjacoby+gmres<2_1_5>(i=1)"
```

Here the solution method exists of one global group, which is iterated 5 times. It consists of two methods, first the equation 4, the angular momentum equation, is solved with AFM (using a default of 2 iterations), then the system of equations: the radial momentum (2), the relativistic density (1) and the internal energy (5), is solved with one iteration using GMRES with a block jacobi preconditioner, which by default uses ILU inside a block, using the PETSc library.

When using PETSc one can additionally specify many options at the command line, e.g. to view the log summary, options table and monitor the residual:

```
> ./run-intel_mpi.sh 2 AstroGRIPS.ini \
" -log_summary -options_table -ksp_monitor_true_residual"
```

or to graphically view the structure of the Matrix A, Preconditioner Matrix A_PC, RHS vector b and Solution vector x in X-Window windows and report these into files, one specifies (for a small problem):

```
> ./run-intel_mpi.sh 2 AstroGRIPS.ini " -my_ksp_draw -draw_pause -1"
```

The KSP Matrix format used by PETSc, which can have possible values of: MPIAIJ, MPIBAIJ, MPIBDiag, MPIRowbs, can be set by specifying the SolMethod option mf, e.g.:

```
SolMethod "2:AFM<4>(i=3),KSP*bjacobi+gmres<2_3_1_5>(i=2,mf=MPIBAIJ)"
```

Here the equations are solved within one group (igroup=1) with a maximum group iteration number of giter_max(igroup=1) = 4.
Within the group iteration loop:

- imethod=1:
  eqn. 4 is solved with AFM
  with iter_max(igroup=1,imethod=1) = 3 iterations

- imethod=2:
  eqns. 2,3,1 and 5 are coupled solved with KSP
  with pc_type=bjacobi and ksp_type=gmres
  with iter_max(igroup=1,imethod=2) = 2 iterations
  using KSP_Matrix_Format = MPIBAIJ.

Using the 'acc' option one can set the requested accuracy of the defect-correction iteration procedure: it is iterated until the requested accuracy is reached or until the maximum iteration number (set by the 'i' option) is reached. If *acc* is not set or set to 0, then it is iterated *i*-times.

With the 'ksp_rtol' option the relative tolerance of the KSP solution method of PETSc can be set.

4.2.3 The parameter file

To easily control the parameters of the simulation a parameter file, which should have the suffix .ini, is used. In the example directory there exists the parameter file AstroGRIPS-all.ini, which describes all possible parameters.

One can also include comments into the parameter file:
Commentary lines start with !,% or #<space>

```
! Comment
% another comment
# yet another comment
```

but

```
#nocomment
```

is no commentary line!

A typical parameter file looks like:

```
!
! parameter file: RelativisticShockTube_alfsh2.0_CNU0.6.ini
!
outfilenamebase RelativisticShockTube_alfsh2.0_CNU0.6

IStart 0
Iprolong 0

! ..... Physical Parameters .......
IGravity 0
IGR 1
ISR 1
```

```
Kerr 0
Spinh2 0.d0
IOmg_FDE 0
SpinNS 0.0

! set EoS_Type = ideal_gas (default), polytropic_gas, isothermal_gas
EoS_Type ideal_gas

! Adiabatic index
gamma 1.6666666666666666666d0

! Viscosity
alftr 0.0d0
! set Art_Visc_Type = Q_art, Q_art2, Eta_art
Art_Visc_Type Q_art
alfsh 2.0d0

! ### Time step control ###
dtmax 5.d-4
dtmin 0.9d-9
! start time step (if dtstart is zero then dtstart = dtmin)
dtstart 1.d-6
timstepcor 1.025d0
CFL 0.4d0
Timmax 0.2d0

! ### Data management ###
NNtime 1000000000

Iter_write 1000000000
Iter_show 1000000000
Iter_log 1
!
! data output in physical time intervals dtwrite
dtwrite 0.2d0
! terminal output in physical time intervals dtshow
dtshow 0.05d0
! log variable output in physical time intervals dtlog
dtlog 1.0d+12

! LogVars parameter with space separated names of logvars
! or "all" for all log variables which should be saved
LogVars "Ntime dt dtexp CFL_No MinL MaxL xxm dfm"
!LogVars "all"


! set solution method direction(s)
```

```
! for 1D (for 1 dimensional solution):
SolMethod_dir R
!SolMethod_dir Z
! for 2D:
!SolMethod_dir RZ (default, necessary for LGSR2)
!SolMethod_dir ZR


! set solution method
SolMethod "5:LGSR2<2,1,5>"


! ### Spatial and temporal accuracies ###
Iordrr 93
Iordzz 93
CNU 0.6d0


! ### Grid Distribution ###

! number of grid levels
Nlev 1

! radial direction
!
! uniform grid in R direction with NZL(lev) grid cells:
! do only edit NZL(lev), Rout(lev) and Rin(lev) here !!!
!Rin 1 1.000d3
!Rout 1 1.001d3
! for Rb-Rout and Ra-Rout in units of Rout need to set Rout first!
!Rb-Rout 1 1.00d0
!Ra-Rout 1 1.00d0
!NZEXT 0
!NZINP 1 0
!NZR 1 0
!NZRC 1 0
!NZL 1 1000
! drmin in units of (Rout(ilev)-Rin(ilev))/NZL(ilev)
!drmin-dRtNZL 1 1.d0
!alf_EXT 1.d0
! Idrmin(1) = 4
!Idrmin 1 4

Rin 1 1.000d3
Rout 1 1.001d3
! for Rb-Rout and Ra-Rout in units of Rout need to set Rout first!
Rb 1 1.00032d3
```

```
Ra 1 1.00072d3
NZEXT 0
NZINP 1 0
NZR 1 1
! equivalent to 1000 uniform:
NZRC 1 400
drmin 1 1.d-3
! equivalent to 2000 uniform:
!NZRC 1 800
!drmin 1 0.5d-3
! equivalent to 4000 uniform:
!NZRC 1 1600
!drmin 1 0.25d-3
NZL 1 1
alf_EXT 1.d0
! Idrmin(1) = 3
Idrmin 1 3


! vertical direction
!
! uniform grid in Z direction with NZD(lev) grid cells:
! do only edit NZD(lev) (, Zout(lev) and Zin(lev) ) here !!!
NZU 1 0
NZZC 1 0
NZD 1 3
! dzmin(ilev)    =  (pi/2.d0)/(NZD(ilev)-1)
! for dzmin-pit2tbNZDm1b need to set NZD first!
dzmin-pit2tbNZDm1b 1 1.d0
! Zout(ilev)     = pi/2.d0 + dzmin(ilev)/2.d0
! for Zout-pit2+dzmint2 need to set dzmin first!
Zout-pit2+dzmint2 1 1.d0
! for Za-Zout and Zb-Zout need to set Zout first!
Za-Zout 1 1.d0
Zb-Zout 1 1.d0
! Zin(ilev)      = -dzmin(ilev)/2
! for Zin-dzmin need to set dzmin first!
Zin-dzmin 1 -0.5d0
! Idzmin(1) = Idzmin(lev) = 4
Idzmin 1 4


! Setting Boundary types:
! e.g.: boundary_type_Rout reflective
! possible types: reflective, zero_gradient, default, fixed
boundary_type_Rout zero_gradient
boundary_type_Rin zero_gradient
```

```
! other parameters:
EM_S 3.d8
R_S 2.00d0
Spinh2 0.0d0

! ## scaling variables ##
R_Dim_in-R_g 2.d0    ! in units of the gravitational radius R_g (set EM_S before!)
V_Dim_in-Clight 1.d0 ! in units of the speed of light Clight
!V_Dim_in-Vs 1.d0    ! in units of the Newtonian/relativistic sound speed
!                      of an ideal gas calculated using gamma and T_Dim
Ro_Dim_in 1.d0       ! in g/(cm^3)
T_Dim_in 1.d0        ! in K (Kelvin)
```

4.2.4 The problem dependent user input file: Setup.F90

The file src/Setup.F90 in a specific problem directory contains the initial conditions. Here also the user has the possibility to setup modified restart conditions and special boundary conditions depending on the problem.

Here an example Setup.F90 file is shown:

```
#include "config.h"
#include "AstroGRIPS.h"

!    Specify the initial distributions of the variables
   subroutine setVariables
      use Parameters
      use Constants, only : one, small, zero
      use Geometry
      use Grid
      use Variables
      implicit none
      integer :: j, k, iVar
      double precision :: rmid

      rmid = (Rout(1)+Rin(1))/2.d0

      ! Set Variables
      Omg = zero
      V(:,:,2) = zero
      Ed = one
      do k = k1LT, k2LT
        do j = j1LT, j2LT
```

```
          V(j,k,1)   = 0.d0
          if(rh(j,1) > rmid) then
            Ro(j,k) = 1.d0
            Pr(j,k) = 2.0d-6/3.d0
           else
            Ro(j,k) = 10.d0
            Pr(j,k) = 40.d0/3.d0
          endif
        enddo
      enddo

      ! activate additional netcdf data output:
      do iVar=1,NumVar
        if( Var(iVar)%name == 'D1Ro' ) Var(iVar)%netcdf_out = .true.
        if( Var(iVar)%name == 'D2Ro' ) Var(iVar)%netcdf_out = .true.
        if( Var(iVar)%name == 'D2ROMR' ) Var(iVar)%netcdf_out = .true.
      enddo

      return
   end

!    sets SolveEqns(j,k):
!    variable SolveEqns(j,k) determines,
!    if equations are solved at gridpoint (j,k):
!    if SolveEqns(j,k) <= 0.d0, the equations are not solved at this gridpoint
!    whereas if SolveEqns(j,k) > 0.d0 e.g. 1.d0 the equations are solved.
!    This variable is used for example for the Forward Facing Step Problem,
!    where a cold "accretion" disc, the area where the equations are not solved,
!    is put into the computational domain to study the shock which is occurring
!    at this 'Forward Facing Step'. The size of this disc is determined by
!    the variables Rin_STEP, Rout_STEP, ThetaMin_STEP and ThetaMax_STEP.
   subroutine setSolveEqns
      use Constants
      implicit none

      SolveEqns = one

      return
   end subroutine setSolveEqns


!  while re-starting, you may still modify/change several parameters
   subroutine setVariablesOnRestart
      use Grid
      use Parameters
      use Geometry
      use Variables
```

```
      use Constants
      implicit none

      return
   end


!    sets user specified boundaries
!    if boundary_type(ind) == BC_user
   subroutine setBoundaryUser(var_ind, bc_ind, j, k)
      use Grid
      use Parameters
      use Geometry
      use Variables
      use Boundary
      use Constants
      implicit none
      integer :: j, k
      integer :: var_ind
      integer :: bc_ind
      double precision :: Temp_Dim, V_r_Rout_Dim

      Temp_Dim = 1.d9  ! Temperature in K (Kelvin)
      V_r_Rout_Dim = -1.0d+5  ! radial velocity at outer boundary in cm/s

      ! var_ind     name
      !  1          Dro
      !  2          Em
      !  3          En
      !  4          El
      !  5          Ed
      !  ...
      ! bc_ind = Rout_ind, Rin_ind, North_ind, South_ind

      select case(var_ind)
      case(1)
        ! Dro
        ! Rout:
        !           DRo(J1G-1,k) =  a_q_out(k,1)*DRo(J1G,k)                  &
        !                         + b_q_out(k,1)*DRo(J1G+1,k)                &
        !                         + c_q_out(k,1)
        ! Rin:
        !           DRo(J2G+1,k) =  a_q_in(k,1)*DRo(J2G,k)                   &
        !                         + b_q_in(k,1)*DRo(J2G-1,k)                 &
        !                         + c_q_in(k,1)
        ! DRo(j,k) = Ro(j,k)*Lorentz(j,k)
        select case(bc_ind)
```

```
        case(Rout_ind)
          ! fixed Ro(J1G-1,k) to value of initial condition
          a_q_out(k,1) = 0.d0
          b_q_out(k,1) = 0.d0
          c_q_out(k,1) = 1.d-9/Ro_Dim*Lorentz(J1G-1,k)
       case(Rin_ind)
          ! zero gradient:
          ! to get Ro(J2G+1,k)=Ro(J2G,k):
          a_q_in(k,1) = Lorentz(J2G+1,k)/Lorentz(J2G,k)
          b_q_in(k,1) = 0.d0
          c_q_in(k,1) = 0.d0
        end select
      case(2)
        ! Em
        ! Rout:
        !          Em(J1G,k) is still the boundary
        !          Em(J1G,k) = a_q_out(k,2)*Em(J1G+1,k)                    &
        !                 + b_q_out(k,2)*Em(j1G+2,k)                       &
        !                 + c_q_out(k,2)
        ! Rin:
        !          Em(J2G+1,k) =  a_q_in(k,2)*Em(J2G,k)                    &
        !                   + b_q_in(k,2)*Em(J2G-1,k)                      &
        !                   + c_q_in(k,2)
        ! Em(j,k) = V(j,k,1) * D2ROMR(j,k)*DSQRT(gdrrmr(j,k))
        select case(bc_ind)
        case(Rout_ind)
          ! fixed inflow velocity at Rout: V_r_Rout_Dim
          a_q_out(k,2) = 0.d0
          b_q_out(k,2) = 0.d0
          c_q_out(k,2) = V_r_Rout_Dim/V_Dim
        case(Rin_ind)
          ! linear extrapolation of Em:
          a_q_in(k,2) = (1.d0 + drh(J2G,1)/drh(J2G-1,1))
          b_q_in(k,2) = - drh(J2G,1)/drh(J2G-1,1)
          c_q_in(k,2) = 0.d0
        end select
      case(3)

...

      case(5)
        ! Ed
        select case(bc_ind)
        case(Rout_ind)
          ! fixed T(J1G-1,k):
          a_q_out(k,5) = 0.d0
```

```
        b_q_out(k,5) = 0.d0
        c_q_out(k,5) = Temp_Dim/T_Dim*Rgas/Emugas                              &
          * DRo(J1G-1,k)/gamam1 * T_Dim/V_Dim**2
      case(Rin_ind)
        ! zero gradient:
        ! T(J2G+1,k) = T(J2G,k)
        a_q_in(k,5) = DRo(J2G+1,k)/DRo(J2G,k)
        b_q_in(k,5) = 0.d0
        c_q_in(k,5) = 0.d0
      end select
    case default
      print *, "setBoundaryUser: varind does not exist! "
    end select

    return
  end subroutine setBoundaryUser
```

4.3 Basic Usage of Astro-GRIPS

According to the Readme.txt file in the example directory the basic usage of the code is explained here:

4.3.1 Basic usage for example problems

- Setting up the environment variable AstroGRIPS_DIR
  can be done by sourcing the setup-AstroGRIPS.sh script,
  which may setup further necessary variables
  e.g. source $HOME/Astro-GRIPS/svn/trunk/setup-AstroGRIPS.sh
  (which can be put in ~/.bashrc)
- current location should be the directory examples,
  elsewise change to it:
  > cd $AstroGRIPS_DIR/examples
- go into the specific example directory
  > cd <specific_example_dir>
  it should contain at least the following files:
  AstroGRIPS AstroGRIPS.ini configure run-intel_mpi.sh run.sh src/Setup.F90
  where AstroGRIPS is a symbolic link to src/AstroGRIPS
  (which can be generated, if it is missing, with > ln -s src/AstroGRIPS AstroGRIPS)
  Instead of or additionally to AstroGRIPS.ini there can be other parameter files ending with .ini
- one may have to update the autotools system in the main directory:
  > (cd $AstroGRIPS_DIR; autoreconf --install)
- one may have to do a make distclean in the main directory,
  if ./configure shows following error:
  configure: error: source directory already configured;
  run "make distclean" there first
  > (cd $AstroGRIPS_DIR; make distclean)
- configure program, e.g.

  for serial run optimized using default compiler:
  > ./configure

  for serial run using gfortran compiler:
  > env FC=gfortran ./configure --enable-debug

  for serial run using ifort compiler and Intel Math Kernel Library:
  > env FC=ifort ./configure --enable-imkl --enable-debug

  for parallel run using mpiifort compiler, Intel-MPI
  with Cluster Intel Math Kernel Library:
  > env FC=ifort ./configure --enable-mpi --enable-debug

  with option --disable-pnetcdf
  parallel-netcdf can be disabled and serial netcdf is used instead.

for parallel run with PETSC use:
> env FC=ifort ./configure --enable-mpi --enable-petsc --enable-debug

for production and profiling runs omit the option --enable-debug!

further configure options like setting library paths are listed by running
> ./configure --help

- compile program

  (>make clean)
  > make 2>&1 | tee make.log

- run example using default parameter file AstroGRIPS.ini:

  for serial run:
  > ./AstroGRIPS
  or using run script
  (which also saves a log file <outfilenamebase>.log
  of what is seen on the terminal):
  > ./run.sh

  for parallel run use e.g. run script:
  > ./run-intel_mpi.sh <nprocs>

- output is a netcdf file <outfilenamebase>.nc
  which can be viewed with netcdf viewers like ncview
  (http://meteora.ucsd.edu/~pierce/ncview_home_page.html)
  or read in and visualized by e.g. the python scripts found in
  $AstroGRIPS_DIR/scripts/python-scripts/

4.3.2 Modification of parameters and initial and boundary conditions

- for modifying parameters e.g. the initial time step size tmin
  change the parameter values
  in the parameter file AstroGRIPS.ini (or in a copy of it)
- for a description and other parameter file options
  look into $AstroGRIPS_DIR/examples/AstroGRIPS-all.ini
  (if changing only parameter values in the parameter file
  program recompilation is not necessary)
- to change the initial condition one has to
  change the file src/Setup.F90 and recompile the program
- run with non-default parameter file e.g. AstroGRIPS-run01.ini:

  for serial run:
  > ./AstroGRIPS -f AstroGRIPS-run01.ini
  or using run script:
  > ./run.sh AstroGRIPS-run01.ini

  for parallel run use e.g. run script:
  > ./run-intel_mpi.sh <nprocs> AstroGRIPS-run01.ini

4.4 Optimization

For implicit methods much computation time is spent in the setup of the solution matrix.

In the old code first all possible matrix entries, all Jacobian coefficients, were calculated and then copied in the solution matrix structure of the appropriate solution method (for LGSR2 and AFM the matrix structure is quite different). This copying process takes a long time, which can be avoided: With the use of a newly introduced matrix function now the necessary matrix elements are directly filled in the appropriate solution matrix structure of the selected solution method. No copying is necessary anymore!

The matrix function calculates the appropriate matrix element depending on the solution method and the boundary effects on the matrix construction by calling the Jacobian function. This split also simplifies the setup of new equations: since for a new equation in general only the new Jacobian function entries have to be specified and the matrix function with all its complicated solution method and boundary dependent settings basically remains as it is.

This restructure of matrix element calculation was also a necessary step for the MPI parallelization of the simulation code.

The recursive function

```
recursive function calcJacobian (idir, ieqG, ivarG, iblock, j, k)          &
  result(A_FCTN)
```

calculates the matrix element of the global general coefficient matrix A, the Jacobian matrix, for the ieqG (e.g. ieqG=1 is Dro) equation after the ivarG-th variable (where ivarG corresponds to the conserved variable of the ivarG-th equation).

It has following parameters:

- idir = specifies direction of direction-dependent part of Jacobian Matrix $\begin{cases} 1 & \text{for } r\text{-direction} \\ 2 & \text{for } \theta\text{-direction} \end{cases}$
- ieqG = global index of equation (eqno), ieq = index of equation
- ivarG = global index of variable (varno), ivar = index of variable
- iblock = $\left\{ \begin{array}{rl} -1 & \text{for sub-diagonal block} \\ 0 & \text{for diagonal block} \\ +1 & \text{for super-diagonal block} \end{array} \right\}$ of the Jacobian Matrix
- j = index in r-direction
- k = index in θ-direction

Recall the definition of the diagonal, sub- and super-diagonal block terms of the Jacobian in equations 3.105 and 3.106:

$$\begin{array}{lcll} \underline{S}^r & = & \underline{S}^r_{j,k} = \frac{\partial \vec{R}^i}{\partial \vec{q}^i_{j-1,k}} & : \quad \text{calcJacobian(idir=1, ieqG, ivarG, iblock=-1, j, k)} \\ \mathcal{D}^r & = & \mathcal{D}^r_{j,k} = \frac{\partial \vec{R}^{r;i}}{\partial \vec{q}_{j,k}} & : \quad \text{calcJacobian(idir=1, ieqG, ivarG, iblock=0, j, k)} \\ \overline{S}^r & = & \overline{S}^r_{j,k} = \frac{\partial \vec{R}^i}{\partial \vec{q}^i_{j+1,k}} & : \quad \text{calcJacobian(idir=1, ieqG, ivarG, iblock=+1, j, k)} \end{array} \tag{4.1}$$

$$
\begin{aligned}
\underline{S}^{\theta} &= \underline{S}^{\theta}_{j,k} = \frac{\partial \vec{R}^{i}}{\partial \vec{q}^{\,\theta}_{j,k-1}} &&: \quad \text{calcJacobian(idir=2, ieqG, ivarG, iblock=-1, j, k)} \\
\mathcal{D}^{\theta} &= \mathcal{D}^{\theta}_{j,k} = \frac{\partial \vec{R}^{\theta;i}}{\partial \vec{q}_{j,k}} &&: \quad \text{calcJacobian(idir=2, ieqG, ivarG, iblock=0, j, k)} \\
\overline{S}^{\theta} &= \overline{S}^{\theta}_{j,k} = \frac{\partial \vec{R}^{i}}{\partial \vec{q}^{\,\theta}_{j,k+1}} &&: \quad \text{calcJacobian(idir=2, ieqG, ivarG, iblock=+1, j, k)}
\end{aligned}
\tag{4.2}
$$

where $\vec{R}^{r;i}$ is the r dependent part of $\vec{R}^{i}$, whereas $\vec{R}^{\theta;i}$ is the θ dependent part of $\vec{R}^{i}$.

This Jacobian matrix element function is used inside the matrix function:

```
double precision function calcMatrix                                    &
  (qdefcorr, idir, ieqG, ivarG, iblock, j, k) result(A_FCTN)
```

which calculates the matrix elements of the matrix to be solved or the defect correction (which is for example used in the case of AFM) depending on the parameter qdefcorr: 0=matrix 1=defect correction. Depending on the solution method the appropriate matrix element construction is applied also taking into account the matrix modifications at the boundaries. This function is used to directly fill in the matrix elements into the corresponding matrix structure of the selected solution method so the calculate and copy approach of the old code is not used anymore which speeds up the matrix construction.

4.5 Parallelization

If one wants to solve a problem, where an extremely large grid is necessary to resolve the fine flow structures in space or if many time steps are necessary to follow very fast changing flows in time, like in simulating the details of turbulent flows, there is no feasible way to perform such a simulation without the help of parallel computers. Especially if the size of the problem is so large that it does not fit in the memory of one computer or one time step will take extremely long there is no way around parallelization.

In comparison to explicit Computational Fluid Dynamics (CFD) codes, which are the most other codes in Astrophysics, the parallelization of implicit methods, which are used in Astro-GRIPS, is not so easy, since it contains the solution of linear systems of equations.

The code was parallelized using the Message Passing Interface (MPI)[1], the de-facto standard for parallel computing for distributed (and shared) memory machines.

To get an optimal and very flexible solver the code was re-organized to make it possible to solve either each separate equation or one or several coupled systems of equations sequentially. For each coupled system of equations one can decide which method should be used, e.g. one could solve first the angular momentum equation with AFM (Approximate Factorization Method) and then the other equations as a coupled system (the equations appearing in the matrix in a specified order) with a Krylov Subspace Iterative Method (KSP or KSIM). This can be easily done by just changing one line in the parameter file. To optimize this process, not the whole global Jacobian matrix, but only the necessary matrix values are calculated and directly filled into the corresponding matrix array of the particular solution method.

For the parallelization the grid and therefore also the corresponding matrix is divided among the used ranks (processes). There are some overlapping cells, the so called halo cells, sometimes also called ghost cells, at the border of the grid on a particular rank. The values of these cells are updated at least once each time step by communication between the processes.

For the solution methods BW-LGS (Black-White Line Gauß-Seidel) and AFM (Approximate Factorization Method) a parallel band-matrix solver from ScaLAPACK, the parallel version of LAPACK, is used. These methods, particularly AFM, are not very well suited for parallelization, since a lot of communication between processes is necessary, which slows down the total runtime extremely, especially on distributed memory machines with slow interconnect between the nodes, like most Linux clusters.

Therefore the Krylov Subspace Iterative Methods (KSIM or KSP), e.g. GMRES or BiCG-stab, were incorporated using the PETSc library, a well tested and very portable and flexible library for solving sparse linear systems. These methods are particularly well suited for parallelization since they only use matrix-vector multiplication and minimum/maximum value determination, which result in very low communication times for sparse matrices.

Due to this large flexibility in using several available solvers one can exploit which solvers are best suited in accuracy and speed and which do not work at all, e.g. give numerical artefacts, for a particular problem.

[1] Good parallel programming workshops are held at the HLRS, the Hochleistungsrechenzentrum Stuttgart (Rabenseifner (ed.) 2007).

To further speed-up the code the data in- and output is now also done in parallel using parallel-netCDF. The netCDF data format (originally only used by the weather and climate forecast community) is a widely used platform independent binary data format, that also allows a short description of the dataset inside the netCDF-file.

Due to this re-organisation and these modifications, especially the parallelization, of the general relativistic implicit code and the usage of High-Performance Computers it is now possible to explore a wide range of new exciting problems in Astrophysics and other related research areas, which were not possible to solve before.

Summary of the parallelization of Astro-GRIPS:

- Computational Domain/Matrix - Decomposition
- Halo Cells and Halo Communication
- BW-LGSR2 and AFM
  implemented using parallelized BLACS/ScaLAPACK routines
- KSIMs/KSP are implemented by the PETSc library,
  which is very flexible and allows to use
  different Krylov Subspace iterative methods: GMRES, BiCGStab, TF-QMR, ...
  with a variety of preconditioners: Jacobi, Block Jacobi, ILU, ...

Runtime, speed up and scalability of Astro-GRIPS

To investigate the parallel performance of Astro-GRIPS a Taylor Couette problem (3 dimensional axisymmetric) was executed repeatedly in parallel using different number of processes. The grid size was chosen to be 144x1152, that is 144 cells in radial direction with radii from 1.0 to 1.2 and 1152 cells in latitudinal direction ranging from the midplane to the polar axis. The simulation was run up to a physical time of 0.5 which corresponds to 272 time steps using either the Krylov Subspace Iterative Method (KSP), GMRES with Block Jacobi as preconditioner implemented by the PETSc library, or the Approximate Factorization Method (AFM).

The test runs were performed on the Helics II cluster of the Interdisciplinary Center for Scientific Computing (IWR) at the University of Heidelberg consisting of 160 nodes of 2 CPUs Dual Core AMD Opterons with 10 Gbit Myrinet interconnect switch. Equivalent runs were also performed on our local Dual Quad Core Intel Xeon compute server. Due to the presence of nodes with multiple multi-core CPUs on this system it is possible e.g. to run a job with 16 processes on 16 nodes using only one core on each node (processes per node number: ppn=1), on 4 nodes using all available cores on the nodes (ppn=4) or on 8 nodes using 2 cores on each node (ppn=2). Since the communication times for the processes in one CPU/one node itself are smaller than between CPUs/nodes, using a different ppn number will lead to a different performance result.

The runtime plot shows the runtime t_N (excluding initialization and terminal and data I/O times, but could be also the total runtime in this particular test case, since the time needed for these tasks is minimal) of the parallel job executed on N processes divided by the runtime t_1 on only one process. In the speed up plot the reciprocal of the runtime ($\frac{1}{\text{runtime}} = \frac{t_1}{t_N}$) is plotted which gives a straight line if one would have ideal speed up. The scalability of a parallel job is defined as $\frac{t_1}{(t_N N)}$ and should always stay at one if one could have ideal performance. For efficient runs the scalability is at least

70-80%. Runs with lower scalability, but larger than 50%, still provide some increase in speed, but very disproportional with respect to the number of used nodes. Scalability of under 50% means hat the runs are slower than runs on less processes.

Amdahl's law states that the increase of computational speed is most likely not proportional to the increase of used processes. So a point, the scalability limit, can be reached, where the increase of processes does not bring any speed increase but stagnation or even decrease. This has to do with the fact, that there is a parallel overhead due to interprocess communication and/or double calculation of halo cell values or there is only a very tiny part in the code that is not (or cannot) be parallelized. The scalability of parallel jobs depends on various parameters, like the computer system (CPUs, memory, interconnect), compilers and used compiler flags, but also on the problem size. The chosen problem size is actually too small for running the problem on too many processes since the ratio of halo cells to physical cells increases with the number of processes and reaches very large values for the used problem size on parallel runs on too many processes (e.g. say maybe larger than 32). So for a larger problem one would expect the scalability limit should actually be reached only for a much larger number of used processes.

Halo to data cell ratio for a problem size of 1152 grid cells in latitudinal direction

number of processes	number of radial halo cells	number of radial halo cells per process	number of radial data cells per process	ratio of halo to data cells
1	0	0	1152	0.00%
2	4	2	576	0.35%
4	12	3	288	1.04%
8	28	3.5	144	2.43%
16	60	3.75	72	5.21%
32	124	3.875	36	10.76%
64	252	3.9375	18	21.88%
96	380	3.9583	12	32.99%

Table 4.1: Halo to data cell ratio for a problem size of 1152 grid cells in latitudinal direction. The cells are divided among the processes and at each process boundary (except for the real boundary) there is a layer of 2 halo cells, which are communicated to the neighbour at least once at every time-step iteration. From the table can be seen that the halo to data cell ratio is increasing with the number of used processes. If this ratio is too large, too much communication in comparison to the calculation has to be done and therefor the simulation slows down. This is the reason why this problem shows only good scalability up to about 32 processes, where the halo to data cell ratio is already about 10%.

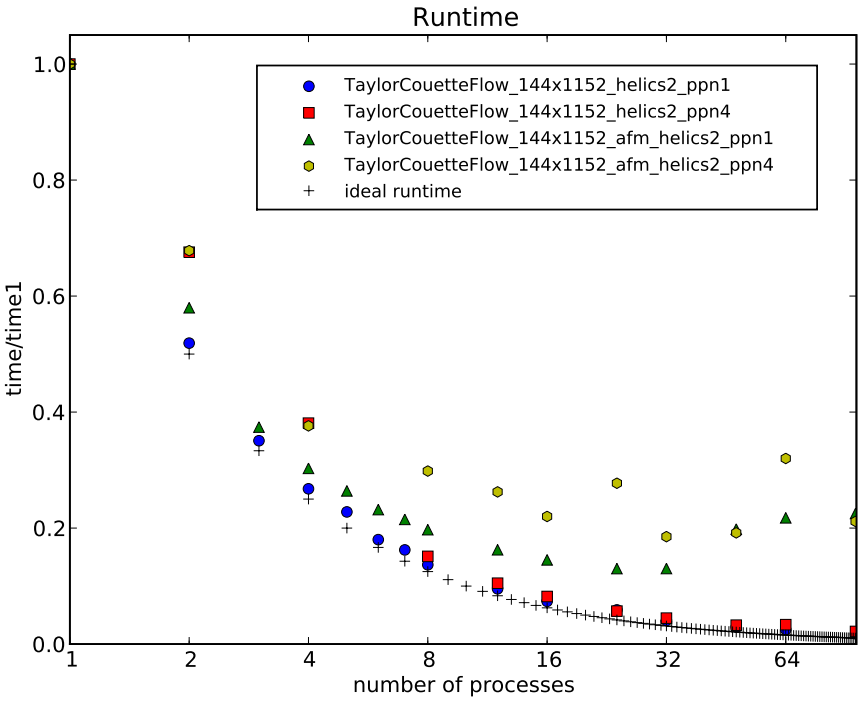


Figure 4.1: *Taylor-Couette Flow parallel Astro-GRIPS runs: execution time on Helics II*

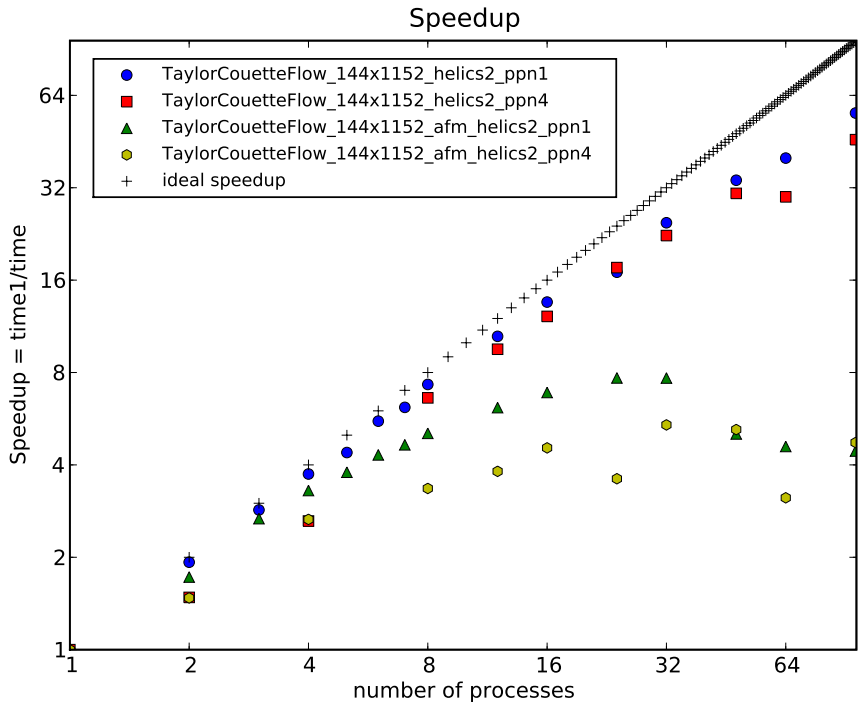


Figure 4.2: *Taylor-Couette Flow parallel Astro-GRIPS runs: Speedup on Helics II*

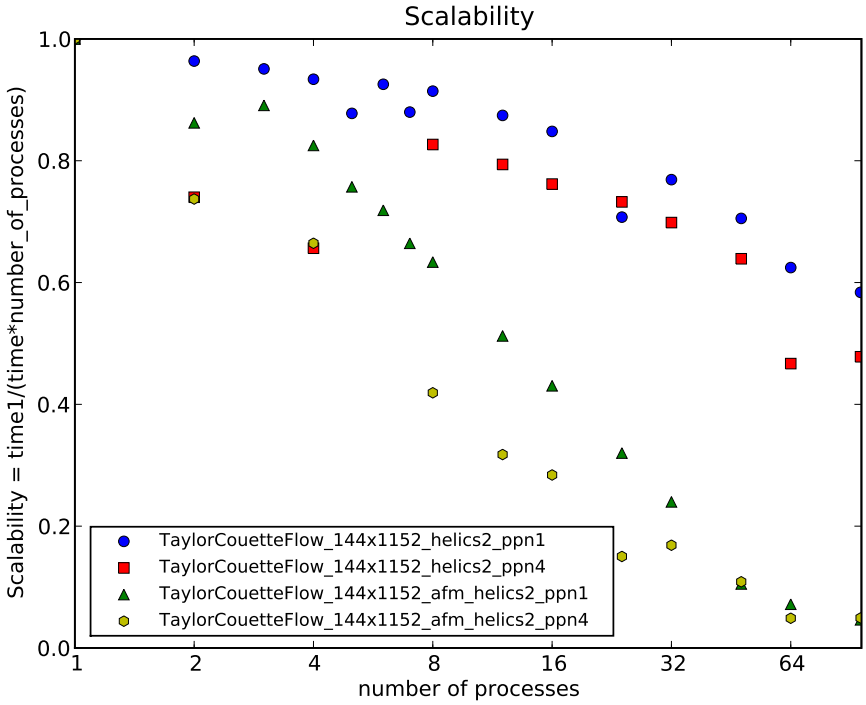


Figure 4.3: *Taylor-Couette Flow parallel Astro-GRIPS runs: Scalability* $\frac{t_1}{(t_N N)}$ *on Helics II: For efficient runs the scalability is at least 70-80%. Runs with lower scalability, but larger than 50%, still provide some increase in speed, but very disproportional with respect to the number of used nodes. Scalability of under 50% means hat the runs are slower than runs on less processes. In this plot one clearly sees that the Approximate Factorization Method (AFM) is not so well suited for parallelization (since a lot of data communication between the processes is necessary) whereas Krylov Subspace Iterative Methods are very well suited, because they basically only consist of matrix-vector multiplication and maximum/minimum determination, which can be implemented very efficiently in a parallel way. The downturn of the curves for more than about 32 processes is caused by the bad ratio between halo cells and real data cells on a single process which is a result of the too small problem size of 144x1152 and increases the halo communication time. So that means that for small problems there exists an optimal number of processes to run on and running on more processes may not help to increase the speed effectively.*

5 Test Problems and Applications

5.1 One-Dimensional Problems

In this section several one-dimensional test problems are presented, which in most cases are compared with their analytical solution.
First on the Burgers' equation is shown the shock capturing, the effect of artificial viscosity.
Then standard shock tube test problems are solved: the standard Newtonian Sod Shock Tube problem is followed by its special relativistic counterpart.
To test the effect of radial pressure terms and gravity, first the Newtonian and then the general relativistic spherical Bondi accretion onto a central object is presented.

5.1.1 Burgers' Equation

With the solution of the Burgers' equation it is shown the technique of shock capturing, the effect of artificial viscosity to correct for the loss of energy at shock fronts due to the use of the internal energy instead of the total energy equation.

The Burgers' Equation is:

$$\frac{\partial \mathrm{u}}{\partial \mathrm{t}} + \mathrm{u}\frac{\partial \mathrm{u}}{\partial \mathrm{x}} = \nu \frac{\partial^2 \mathrm{u}}{\partial \mathrm{x}^2} \text{ with } \nu = 10^{-3} \tag{5.1}$$

In figure 5.1 starting from the initial condition $t = 0$ (dark blue) a time series with times $t = 0.2$ (green), 0.4 (red) and 0.6 (light blue) can be seen.
On the left the velocity of the Burgers' equation solved with the explicit method without shock capturing ($\alpha_{sh} = 0$) is plotted. One can clearly see the velocity oscillations at the shock front.
The middle plot shows the same solution now with applied shock capturing with $\alpha_{sh} = 1$. The artificial viscosity can be seen in the right plot, which shows, that the artificial viscosity is only applied at the shock front and smoothes out the occurring oscillations.
When using implicit methods with shock capturing which can be seen in figure 5.2, one gets even better results for the velocity distribution, since there are no oscillations at the shock front seen at all.

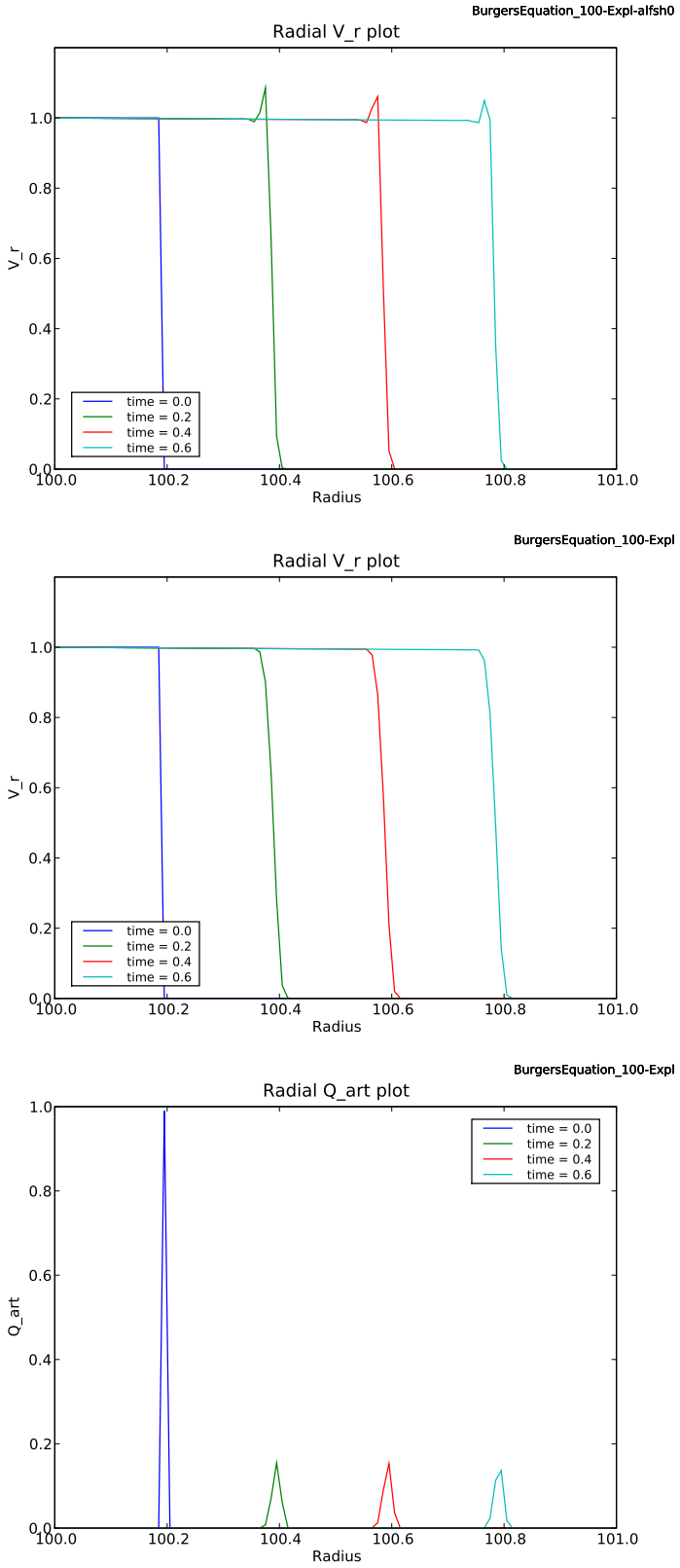


Figure 5.1: *Burgers' equation solved with the explicit method using a CFL number of 0.45 without (top) and with shock capturing with alfsh=1 (middle) with corresponding artificial viscosity (bottom).*

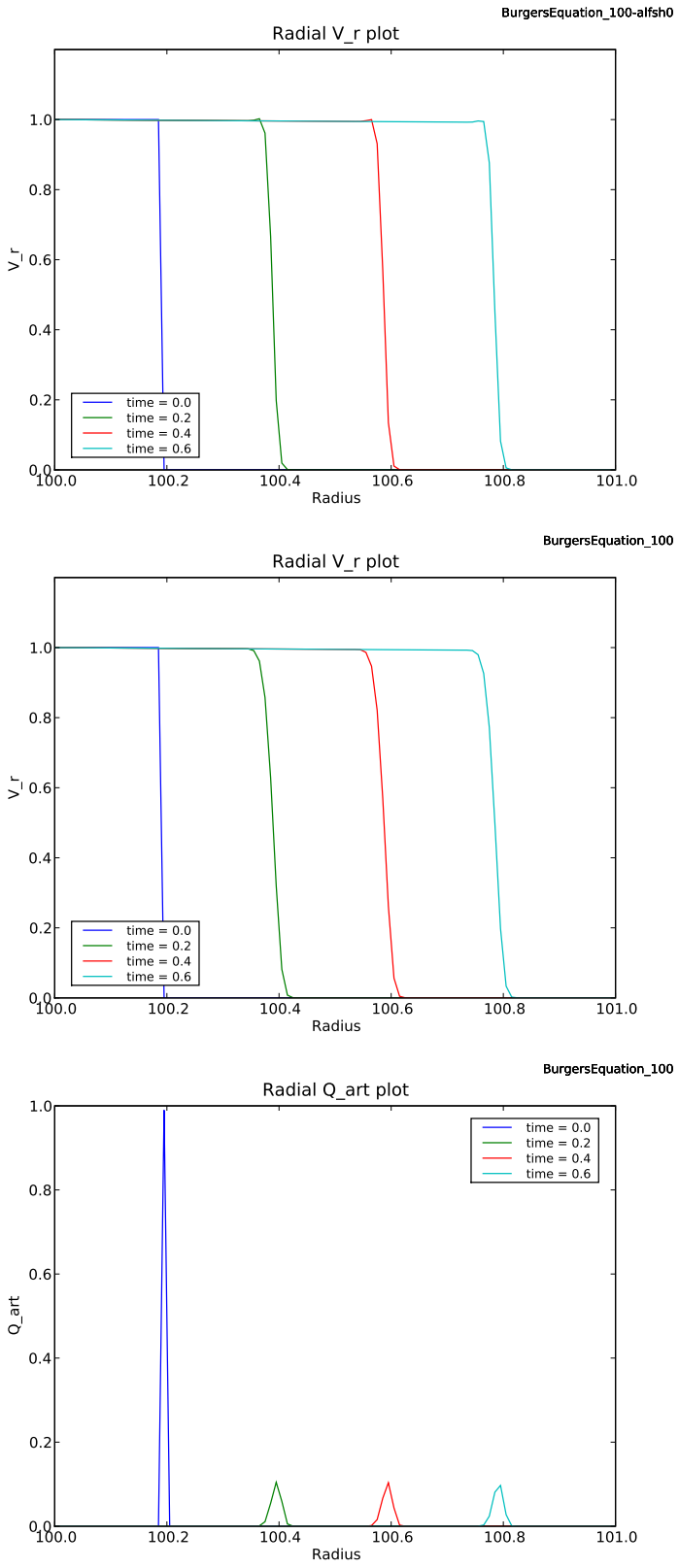


Figure 5.2: *Burgers' equation solved with the implicit method with a CFL number of 0.45 without (top) and with shock capturing with alfsh=1: velocity (top) and corresponding artificial viscosity (bottom).*

5.1.2 Shock Tube Problem

The Shock Tube problem describes a one-dimensional flow in a tube. As initial conditions a fluid is separated by a diaphragm dividing it in left and right homogeneous states with different density and pressure values. At the start time of the simulation the diaphragm is removed in no-time. Usually the fluid is initially at rest in both parts, but due to the step in density and pressure the fluid begins to move and tries to cancel out the differences on the left and the right side. This problem, where one has initially step functions of the state variables, is called Riemann Problem.

The Riemann Problem

The standard form of a conservation law is:

$$q_t + f(q)_x = b \tag{5.2}$$

with vector of conservative states $q = q(x,t) \in \mathbb{R}^m$ and flux-function $f(q)$ and b denotes the source terms, which should usually be zero for real conservation laws (but there can also occur geometrical source terms due to a curved coordinate system).

Linearizing the flux function $f(q) = Aq$, the corresponding linearized hyperbolic system is:

$$q_t + Aq_x = b \tag{5.3}$$

with a $m \times m$-matrix A, which is the (constant) Jacobian matrix of the flux function, and has m right eigenvectors r_p and m different real eigenvalues λ_p; $p = 1, 2, \ldots, m$ with $\lambda_1 \leq \lambda_2 \leq \ldots \lambda_m$, that means the system is hyperbolic, and if the eigenvalues are all distinct one speaks of a strictly hyperbolic system.

The initial data is $q(x,0) = q_0(x)$.

Now remember the three-step principle of a Godunov type scheme (see page 74): reconstruct-solve-average.

First from cell-averaged values at each grid point one constructs the state (or flux) values at the cell interfaces, and then at each interface local Riemann problems are solved. After that an average is applied to get the new cell-averaged values at each grid point. Then one can continue with the next time-step.

The Riemann problem is described by the above linearized hyperbolic system with following initial conditions:

$$q(x,0) = q_0(x) = \begin{cases} q_l & : \quad x < 0 \\ q_r & : \quad x \geq 0 \end{cases} \tag{5.4}$$

The left and the right state can be written in the basis of the eigenvalues of A:

$$q_l = \sum_{p=1}^{m} v_p^l r_p \tag{5.5}$$

$$q_r = \sum_{p=1}^{m} v_p^r r_p \tag{5.6}$$

The solution can be constructed by the method of characteristics:
the system of coupled equations can be transformed in a system with equations that are decoupled, which is done by determining the eigenvalues and the eigenvectors of the Jacobian matrix of the flux function. For each equation of that decoupled system the solution can be found very easily, since each equation is a simple advection equation which advances the solution along the characteristics, in the case of a linear system the solution is shifted with a constant speed.

The decomposed state vector is:

$$q(x,t) = \sum_{p=1}^{m} v_p(x - \lambda_p t) r_p \tag{5.7}$$

Then from the initial conditions:

$$v_p(x,0) = \begin{cases} v_p^l & : \quad x < 0 \\ v_p^r & : \quad x \geq 0 \end{cases} \tag{5.8}$$

the decoupled advection equations are advanced (with the speed λ_p):

$$v_p(x,t) = \begin{cases} v_p^l & : \quad x - \lambda_p t < 0 \\ v_p^r & : \quad x - \lambda_p t \geq 0 \end{cases} \tag{5.9}$$

$P(x,t)$ should be the largest value of p for which $x - \lambda_p t > 0$, then the solution is:

$$q(x,t) = \sum_{p=1}^{P(x,t)} v_p^r r_p + \sum_{p=P(x,t)+1}^{m} v_p^l r_p \tag{5.10}$$

In each wedge of the $x-t$ plane the solution is constant. At the p-th characteristics (the lines separating the wedges), the solution jumps with a step of:

$$[q] = (q_r - q_l) = (v_p^r - v_p^l) r_p \tag{5.11}$$

For the flux function one gets:

$$[f] = (f(q_t) - f(q_l)) = (v_p^r - v_p^l) A r_p = A[q] = \lambda_p [q] \tag{5.12}$$

where λ_p is the speed with which the p-th jump is moving (the p-th discontinuity is propagating). This condition is called the Rankine-Hugoniot jump condition.

Finally one gets for the solution of the Riemann problem:

$$q(x,t) = q_l + \sum_{\lambda_p < \frac{x}{t}} (v_p^r - v_p^l) r_p \tag{5.13}$$

$$= q_r - \sum_{\lambda_p > \frac{x}{t}} (v_p^r - v_p^l) r_p \tag{5.14}$$

For the Godunov type schemes now one has to apply an average to get the cell-averaged values at each grid point. Due to the fact, that it is not allowed that the local Riemann solution overlaps with the neighbouring Riemann solution, the time-step has to be limited:

$$\Delta t \leq \frac{\Delta x}{2 \max(|\lambda_p|)} \tag{5.15}$$

where $\max(|\lambda_p|)$ is the maximum of the largest modulus of the eigenvalues (characteristic speeds) at each grid point.
This is a general constraint for Riemann solver based methods and is consistent with the CFL-condition. It shows, that Riemann solvers are not suitable for implicit methods, which usually use large timesteps, therefore other advection methods are used for implicit methods.

Shock Tube problem

This procedure to solve the Riemann problem can now be applied to the hydrodynamic Euler equations to get the solution of the shock tube problem (actually it has to be slightly modified since one solves now a non-linear system of equation):

The eigenvalues of the 1D Euler equations with equation of state of an ideal gas $P = (\gamma - 1)\rho\epsilon$ are:

$$\lambda_1 = u - a \tag{5.16}$$
$$\lambda_2 = u \tag{5.17}$$
$$\lambda_3 = u + a \tag{5.18}$$

which correspond to the flow speed u and two acoustic waves travelling with sound speed $\pm a$ relative to the flow. The sound speed can be calculated by $a = \sqrt{\frac{\gamma P}{\rho}}$.

The solution can possess three different kind of waves, depending on the difference of the eigenvalues of the left and the right state.
The characteristic front S_p which separates q_l from q_r is

- a contact discontinuity, if $\lambda_p(q_l) = \lambda_p(q_r)$
- a shock wave, if $\lambda_p(q_l) > S_p > \lambda_p(q_r)$
- a rarefaction wave, if $\lambda_p(q_l) < \lambda_p(q_r)$

For the 1D Euler equation one can get, depending on the initial conditions, four possible solutions:

1. a rarefaction wave moving to the left and a shock travelling to the right.
2. a shock travelling to the left and a rarefaction wave moving to the right.
3. two rarefaction waves, one moving to the left and one to the right.
4. two shocks, one propagating to the left and one to the right.

All left and right states are separated by a contact surface.

In the case of non-linear systems, which for example the Euler or Navier-Stokes equations are, the construction of the solution is a little bit more complicated because additionally one has to fulfil the so-called entropy condition across the discontinuities to get the right physical weak solution (see for example: Leveque 1998).

Sod Shock Tube Problem

In 1978 Sod (Sod 1978) compared different numerical methods applied to the shock tube problem. Since then this problem with his initial conditions is used as a general test problem for hydrodynamical simulation codes.

The Sod Shock Tube Problem is a Riemann problem with following initial conditions:

$$\begin{pmatrix} \rho_L \\ P_L \\ u_L \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \text{ for } x < 0.5, \quad \begin{pmatrix} \rho_R \\ P_R \\ u_R \end{pmatrix} = \begin{pmatrix} \frac{1}{8} \\ \frac{1}{10} \\ 0 \end{pmatrix} \text{ for } x \geq 0.5. \tag{5.19}$$

Using Sod's initial conditions for the Shock Tube Problem one gets following results (see figure 5.3):

- a rarefaction wave moving to the left,
- a contact discontinuity propagating slowly to the right,
- a shock wave travelling fast to the right.

Simulation results:

For all simulations a domain of [0,1] (or a domain of [1000,1001] in radial direction to minimize curvature effects due to the spherical grid) and 400 grid cells were used. As equation of state (EoS) an ideal gas with $\gamma = 1.4$ was selected. The final integration time was set to 0.2.

In figures 5.4 and 5.5 the Sod Shock Tube Problem is solved with PLUTO (Mignone et al. 2007), a very flexible, well-documented and user-friendly explicit solver for Newtonian and special relativistic Astrophysical (magneto-) hydrodynamic flows. In figure 5.4 one sees the optimal solution, whereas in 5.5 the instability problems of some explicit methods (Euler type time-stepping, linear interpolation) even for CFL numbers smaller than one is demonstrated.

In figure 5.6 the Sod Shock Tube problem is solved with Astro-GRIPS using the explicit method. Since the internal energy equation is used, one has to tune the shock capturing parameter α_{sh} to an appropriate value.

Figure 5.7 shows the result obtained with Astro-GRIPS using an implicit method using the third order spatial van Leer advection scheme, a Crank-Nicolson number of $\vartheta_{CN} = 0.75$ and a CFL number of 0.8, where the artificial viscosity parameter α_{sh} was set to a suitable value to account for the correction of the energy loss due to the internal energy formulation. Note, that since this is a highly time-dependent problem a higher CFL number was not used here due to physical accuracy, although it would not result into numerical instabilities.

For better comparison the analytic solution for all variables is plotted together with the simulation results in one plot. This shows that Astro-GRIPS can reproduce Sod's solution with high accuracy.

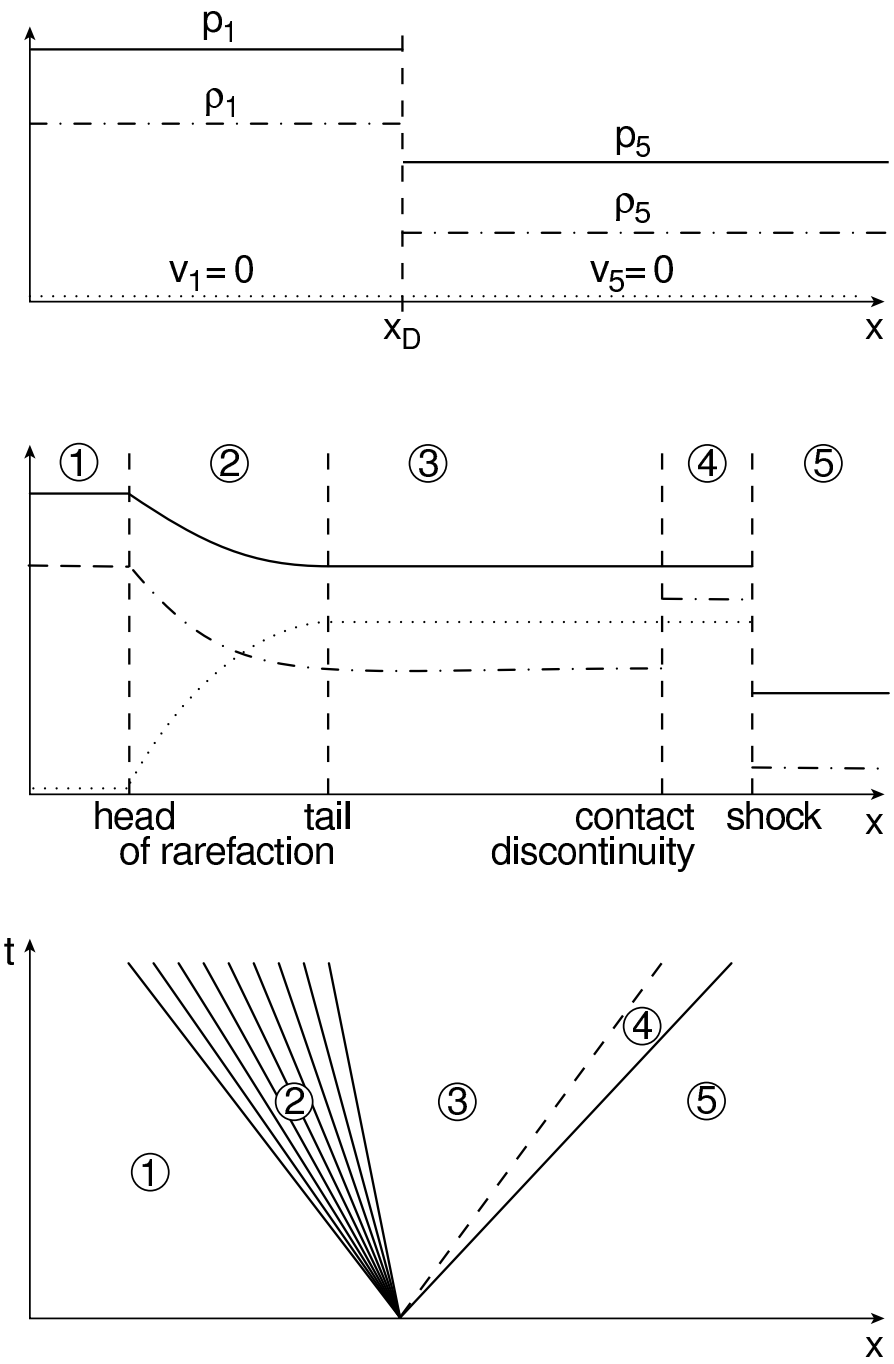


Figure 5.3: *Riemann problem: In the top plot the initial condition, a step in density and pressure at point x_D usually with zero velocity, in the middle plot the solution and in the bottom plot the Riemann fan in the $x-t$ plane, which describes the time-development of the solution, is shown. The density (dashed-dotted line), the pressure (solid line) and the velocity (dotted line) are shown. There are 5 regions of the solution: region 1 contains the unperturbed left state, region 2 consists of the rarefaction wave travelling to the left, region 3 and 4 are separated by a jump in the density, the contact discontinuity, but have constant pressure and velocity, The front of the shock separates region 4 and the unperturbed right state in region 5. Note that this plot is for the solution of the Special Relativistic Riemann Problem, in the Newtonian case the shape of the rarefaction wave is not very curved, but almost linear instead! (plot for the Special Relativistic Riemann Problem from Martí and Müller (2003))*

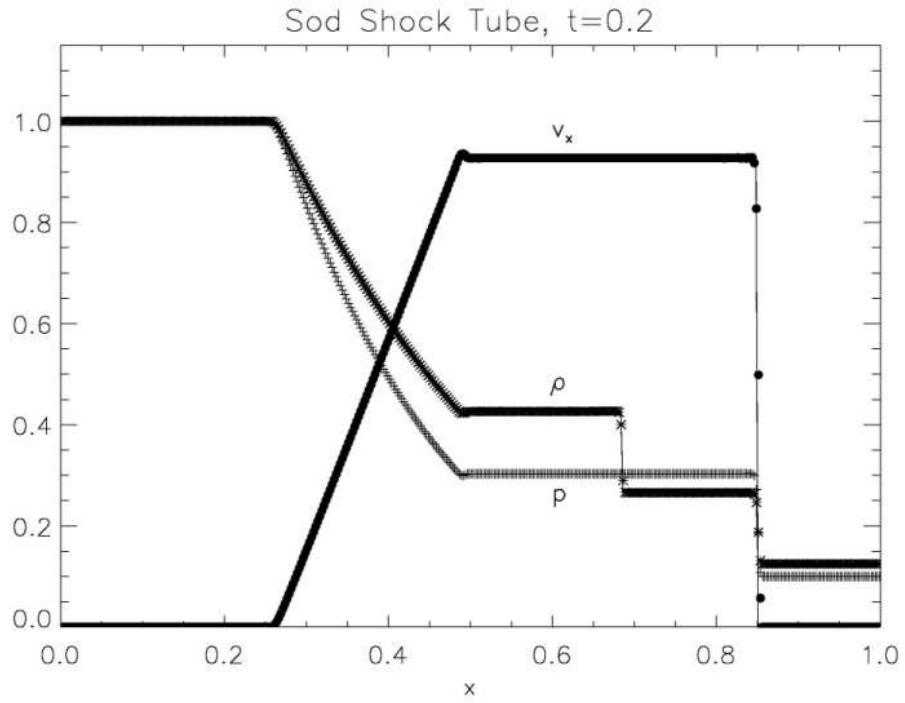


Figure 5.4: *Sod Shock Tube from Pluto test gallery: Time stepping with charact. tracing, Interpol.: parabolic on primitive variables, Riemann Solver: two-shock, CFL=0.8*

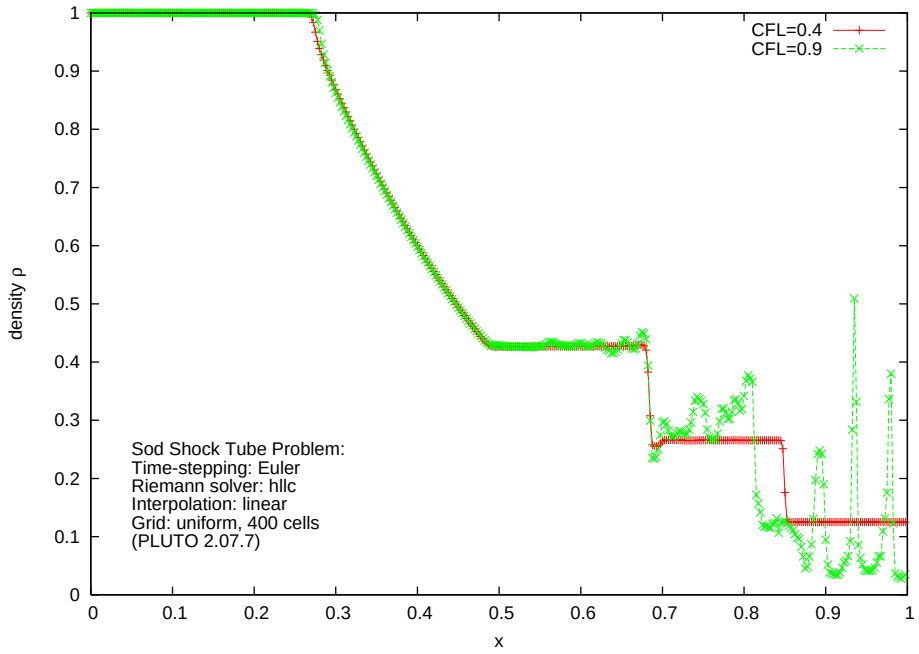


Figure 5.5: *Shock tube problem with CFL=0.4 and 0.9 with PLUTO. Although both CFL-numbers are smaller than unity the numerical solution procedure does not appear to be stable even with CFL=0.9 (Hujeirat, Keil and Heitsch 2007 (arXiv:0712.3674v1)).*

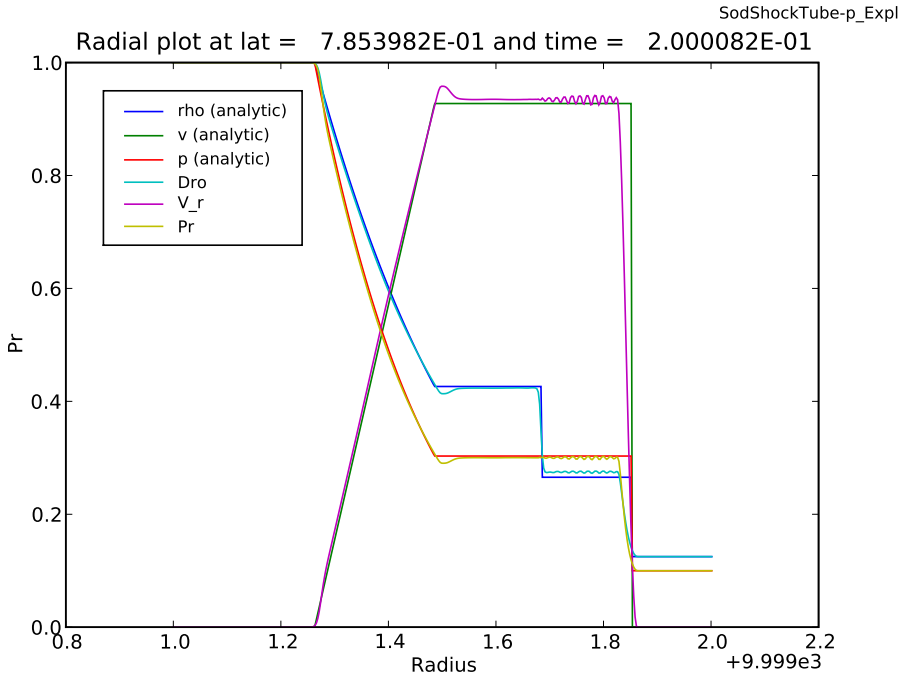


Figure 5.6: *Sod Shock Tube problem solved with Astro-GRIPS using the explicit method, third order in space and second temporal order with shock capturing α_{sh}=32 and CFL=0.4 (for CFL > 0.55 the solution is oscillating very much or the code is aborted due to negative pressure)*

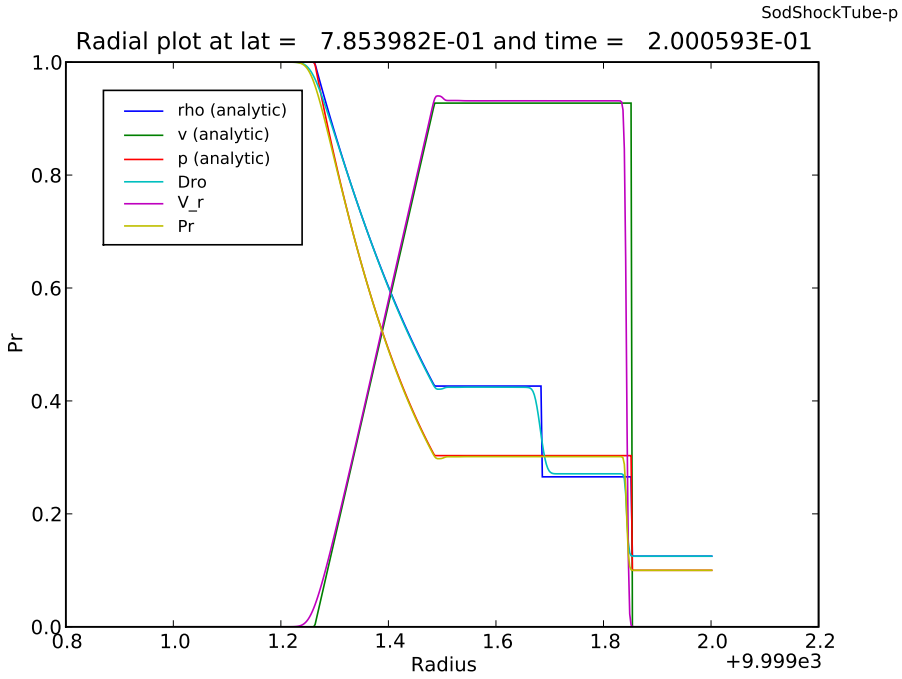
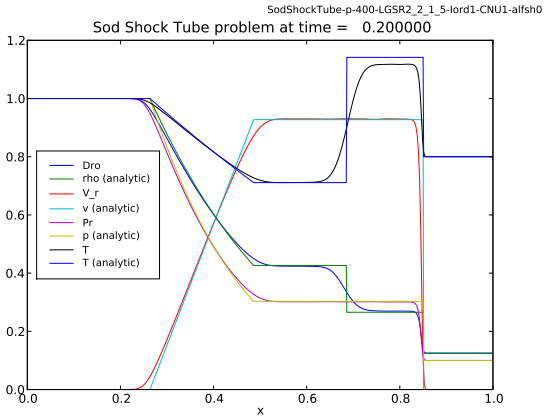
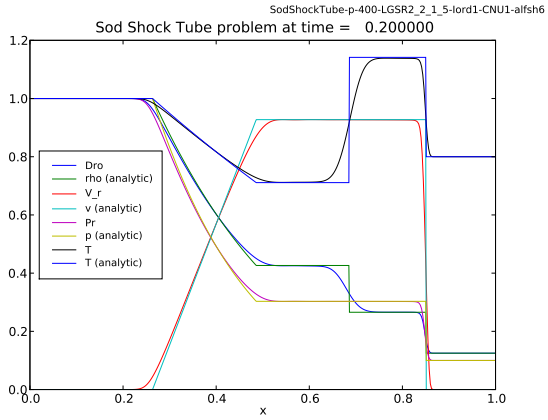


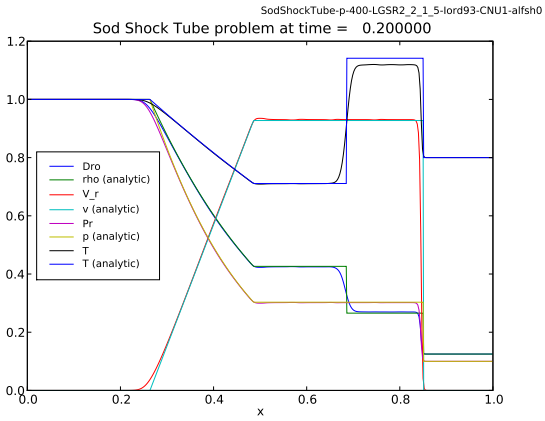
Figure 5.7: *Sod Shock Tube problem solved with Astro-GRIPS using the implicit method, third order in space and second temporal order with shock capturing α_{sh}=1 and CFL=0.8.*



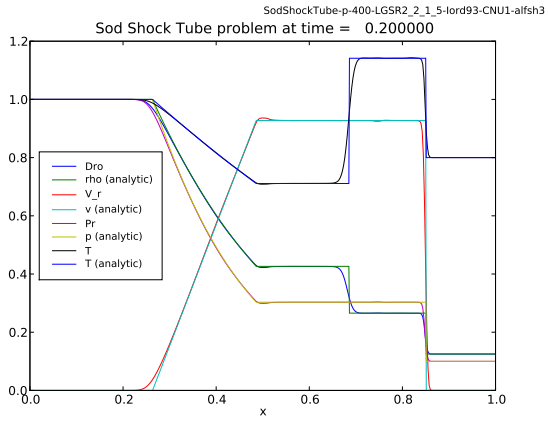
(a) *1st order; no artificial viscosity; 400 grid points*



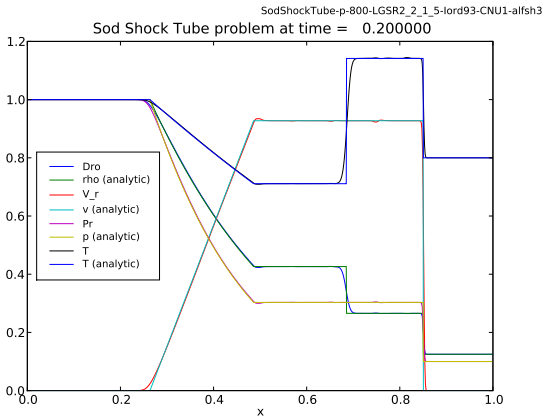
(b) *1st order; artificial viscosity* $\alpha_{sh} = 6.0$*; 400 grid points*



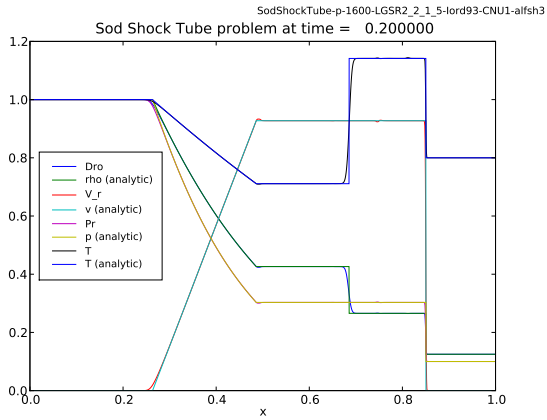
(c) *3rd order van Leer; no artificial viscosity; 400 grid points*



(d) *3rd order van Leer; artificial viscosity* $\alpha_{sh} = 3.0$*; 400 grid points*



(e) *3rd order van Leer; artificial viscosity* $\alpha_{sh} = 3.0$*; 800 grid points*



(f) *3rd order van Leer; artificial viscosity* $\alpha_{sh} = 3.0$*; 1600 grid points*

Figure 5.8: *Sod Shock Tube Problem simulations performed with Astro-GRIPS using different spatial orders, different artificial viscosity and different number of grid cells.*

5.1.3 Relativistic Shock Tube Problem

With the Relativistic Shock Tube Problem the non-linear wave advection and the conservation laws of one-dimensional special relativity (without gravity) can be tested.

For the Relativistic Shock Tube Problem now the specific internal energy ϵ can be large, and does not have to be small as in the Newtonian case: $\epsilon \ll c^2$. The speed of the shocks in the Newtonian Shock Tube Problem can be made arbitrary large, depending on the initial ratio of the pressure in the tube. While non-relativistic solvers may produce propagating velocities that exceed the speed of light, a conservative and accurate relativistic solver produces velocities that can be extremely close to but never exceed the speed of light and therefore the flow can have a very high Lorentz number Γ.

The initial conditions of the simulations in figure 5.9 and following figures are:

$$\begin{pmatrix} \rho_L \\ P_L \\ u_L \end{pmatrix} = \begin{pmatrix} 10 \\ f\,\frac{40}{3} \\ 0 \end{pmatrix} \text{ for } x < 0.5, \quad \begin{pmatrix} \rho_R \\ P_R \\ u_R \end{pmatrix} = \begin{pmatrix} 1 \\ \frac{2\,10^{-6}}{3} \\ 0 \end{pmatrix} \text{ for } x \geq 0.5. \tag{5.20}$$

where the factor f is changed to get different maximum Lorentz factors:
for figure 5.9: $f = 1$, which results in a maximum Lorentz factor $\Gamma_{max} \approx 1.4$,
for figure 5.10: $f = 10, \Gamma_{max} \approx 2.3$, figure 5.11: $f = 100, \Gamma_{max} \approx 3.8$,
figures 5.12, 5.13, 5.14 and 5.15: $f = 1000, \Gamma_{max} \approx 6.8$ and for figure 5.16: $f = 100000, \Gamma_{max} \approx 21$.
In the simulations an equation of state of an ideal gas with adiabatic index $\gamma = \frac{5}{3}$ was used.

The simulations were run up to a time of 0.2 and, since this is a very time-dependent problem and to be sure, that there are no numerical instabilities due to the CFL condition are occurring, a maximum CFL time step of 0.4 was used.
The equations were solved with an implicit method using the Black-White Line-GaußSeidel method for the solution of the linear matrix equations with 5 global iterations using the implicit operator splitting (IOS) technique with the equation order 2, 1, 5: first radial momentum m equation, then the continuity equation and finally the energy equation are solved in each global iteration step.
The spatially third order van Leer advection scheme ($Iordrr = 93$) and a non-unity Crank-Nicolson number $\vartheta_{CN} \neq 1$ was used, so explicit and implicit methods were mixed. The artificial viscosity parameter α_{sh} and the Crank-Nicolson number ϑ_{CN} were varied to find the best solutions which are presented here.

For one-dimensional special relativity (ISR=1 and IGravity=0) in Astro-GRIPS the metric entries are set to: $g_{tt} = -1, g_{t\phi} = g_{\phi t} = 0, g_{rr} = g_{\theta\theta} = g_{\phi\phi} = 1$
and to get a 1D radial solution one sets in the parameter file:
SolMethod_dir R

Since up to now Astro-GRIPS uses only spherical shaped cells in the calculation of the volumes/areas and interface areas/lines, one has to use a large radius (e.g. 1000 or larger) together with a small radial computational domain (e.g. 1) to approximate a Cartesian behaviour in the one-dimensional simulations.

For the calculation of the Lorentz factor Γ in these simulations a mixed scheme is applied where conservative and primitive variables are used. Another scheme uses only the conservative variables and can be easily switched on by a flag in the parameter file (calcLorentz_Type Lorentz_mixed or calcLorentz_Type Lorentz_cons).

The analytic solution is constructed using the Fortran program riemann.f by Martí and Müller (2003) where also the derivation of the analytic solution is described.

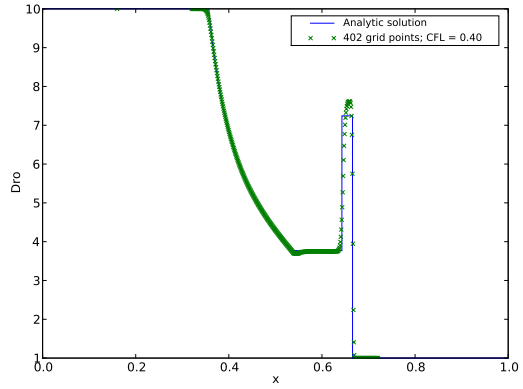
Compared to the Newtonian case the maximum compression ratio of the gas in not limited anymore and thus in the relativistic case it is much more difficult to numerically determine the solution due to the appearance of stronger shock discontinuities.

Note that in these simulations the pressure and energy density is not artificially limited to floor values at each cycle to ensure only positive values, as for example is done in Anninos and Fragile (2003). For the right artificial viscosity and Crank-Nicolson parameters there is no need to limit any variables to floor values.

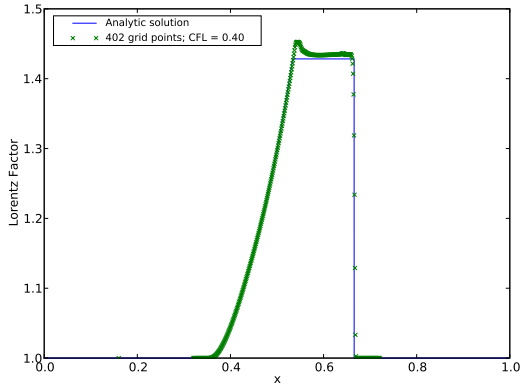
The higher the initial pressure ratio is, the thinner the density shock feature and the higher the maximum Lorentz factor gets. So for very large initial pressure ratios one has to use an enormous fine grid, so that the extremely thin density shock feature is resolved well. The lack of spatial resolution is the cause for the density undershoot in the simulations and therefore also the Lorentz factor might not be calculated well enough.

That is probably the reason, why Anninos and Fragile (2003) have only performed simulations of the relativistic shock tube problem for maximum Lorentz factors of about 1.43 and 3.59. They used another problem to test their explicit Cosmos code for large Lorentz factor flows: the Relativistic Wall Shock Problem.

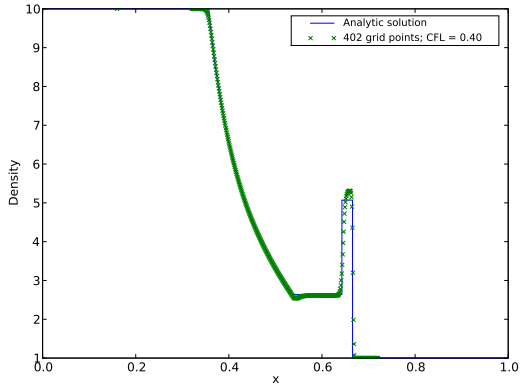
Explicit methods are not able to handle these type of flows with strong shocks and very high Lorentz factors, due to the strong nonlinearity of the equations. There are various explicit codes, which will break down for high Lorentz factors, e.g. PLUTO (`http://plutocode.to.astro.it`), which gives usually very good results, breaks down for Lorentz factors that are larger than about 15 (Hilscher 2009).



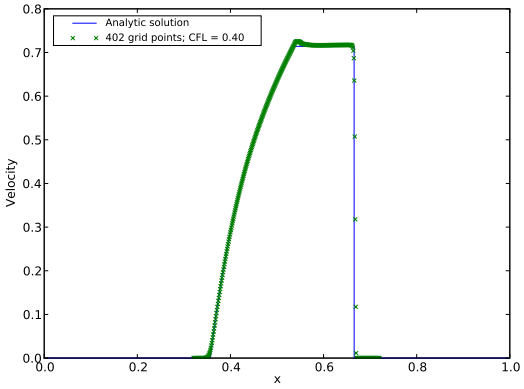
(a) *Relativistic Density* $D = \rho\Gamma$



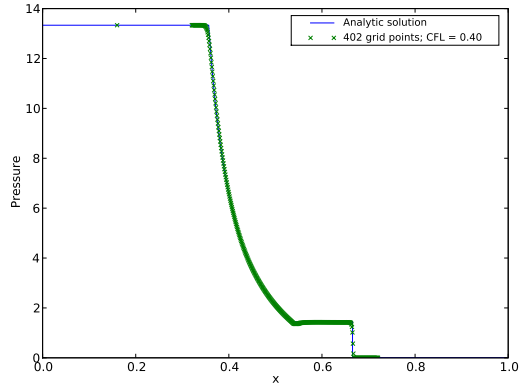
(b) *Lorentz Factor* $\Gamma = \frac{u^t}{c}$



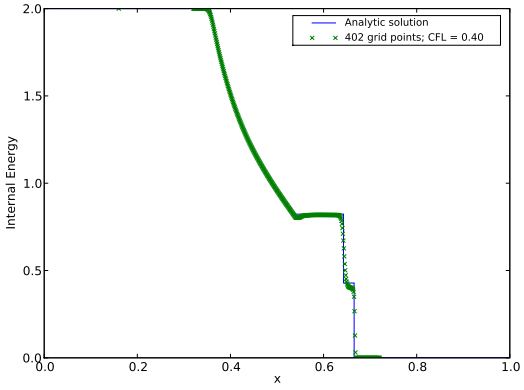
(c) *Density* ρ



(d) *Velocity* V

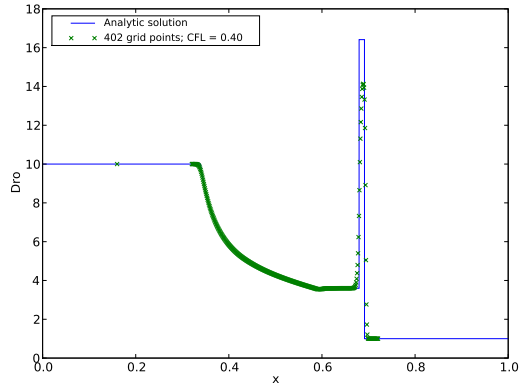


(e) *Pressure* P

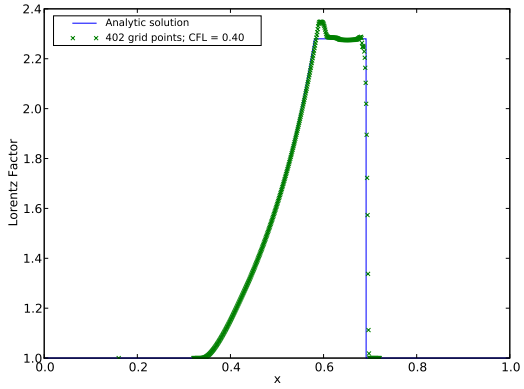


(f) *Internal Energy* $\epsilon = \frac{1}{\gamma-1}\frac{P}{\rho}$

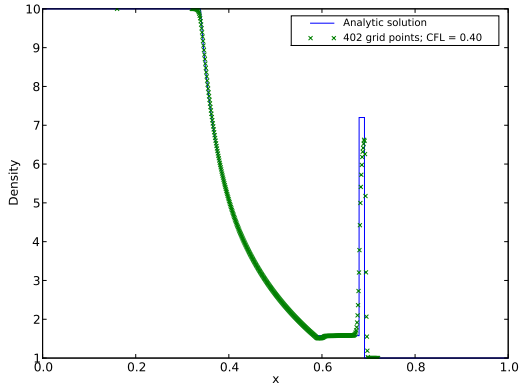
Figure 5.9: *Relativistic Shock Tube Problem with maximum Lorentz factor of approx. 1.4; non-uniform grid distribution with 402 grid points corresponding in the relevant region to a uniform grid of 1000 cells between 0 and 1; the optimal solution is obtained for an artificial viscosity parameter* $\alpha_{sh} = 2.0$ *and a Crank-Nicolson factor* $\vartheta_{CN} = 0.6$



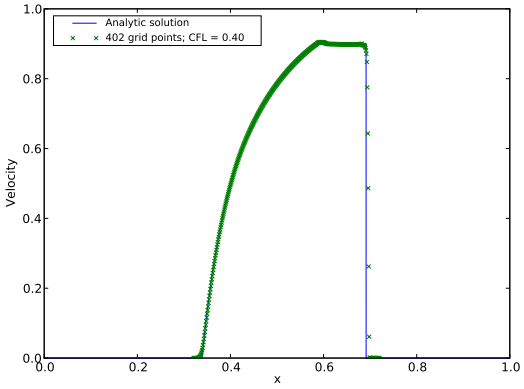
(a) *Relativistic Density* $D = \rho\Gamma$



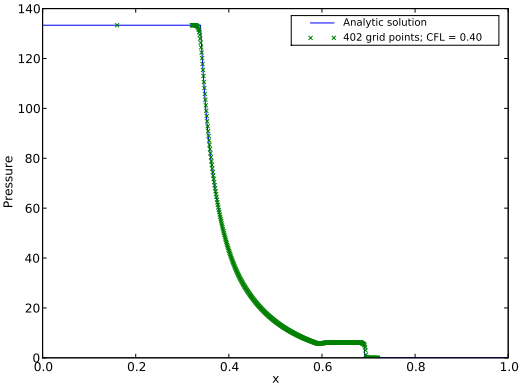
(b) *Lorentz Factor* $\Gamma = \frac{u^t}{c}$



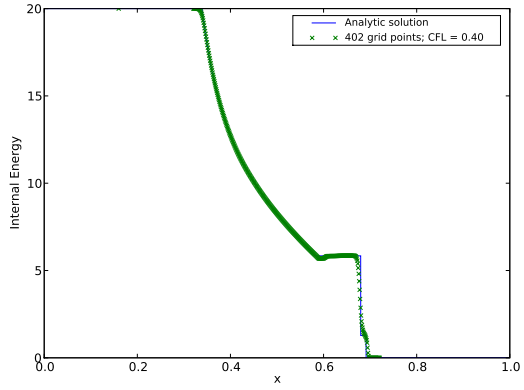
(c) *Density* ρ



(d) *Velocity* V

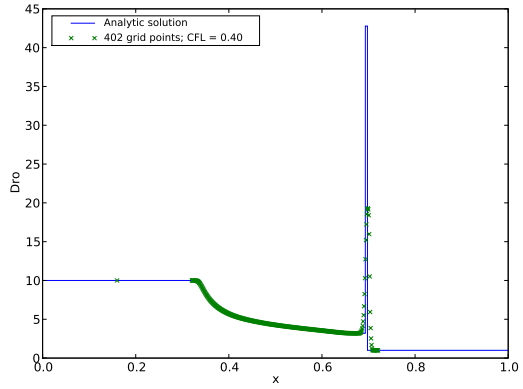


(e) *Pressure* P

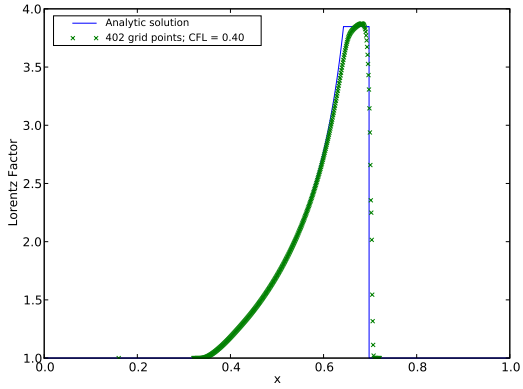


(f) *Internal Energy* $\epsilon = \frac{1}{\gamma-1}\frac{P}{\rho}$

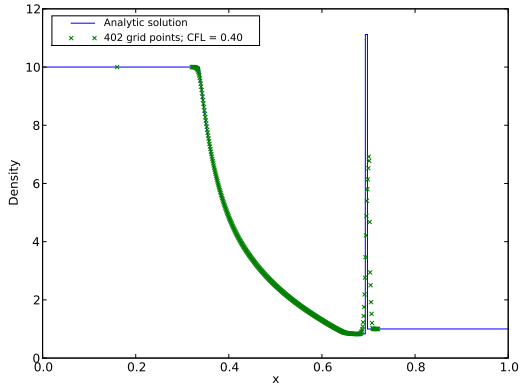
Figure 5.10: *Relativistic Shock Tube Problem with maximum Lorentz factor of approx. 2.3; non-uniform grid distribution with 402 grid points corresponding in the relevant region to a uniform grid of 1000 cells between 0 and 1; the optimal solution is obtained for an artificial viscosity parameter* $\alpha_{sh} = 2.0$ *and a Crank-Nicolson factor* $\vartheta_{CN} = 0.55$



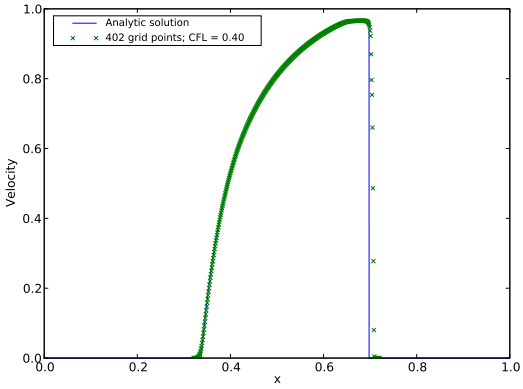
(a) *Relativistic Density* $D = \rho\Gamma$



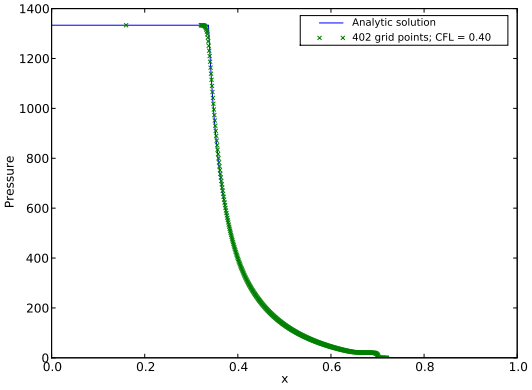
(b) *Lorentz Factor* $\Gamma = \frac{u^t}{c}$



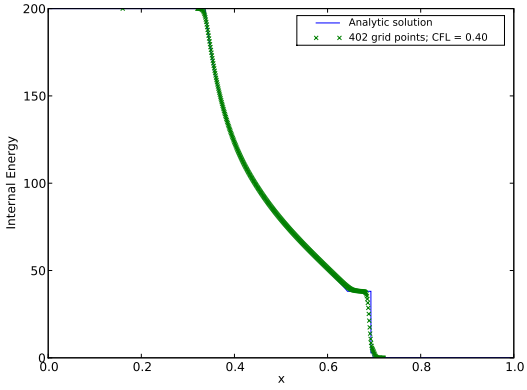
(c) *Density* ρ



(d) *Velocity* V

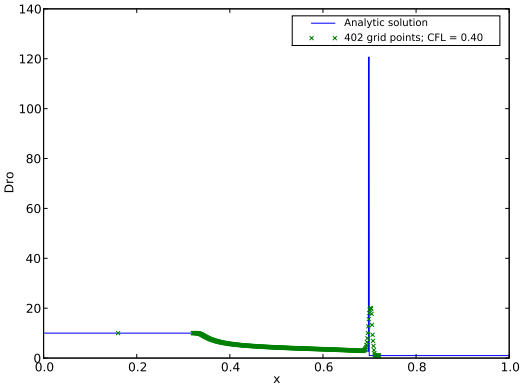


(e) *Pressure* P

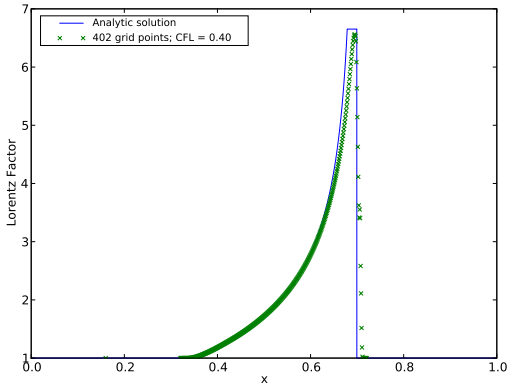


(f) *Internal Energy* $\epsilon = \frac{1}{\gamma-1}\frac{P}{\rho}$

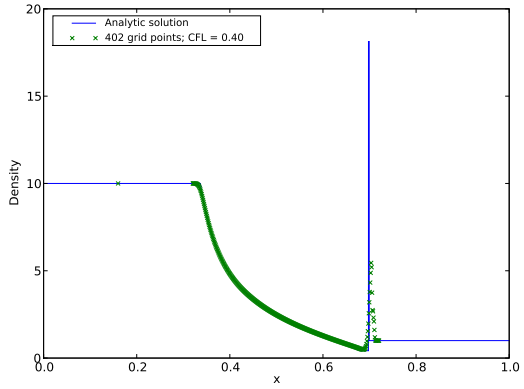
Figure 5.11: *Relativistic Shock Tube Problem with maximum Lorentz factor of approx. 3.8; non-uniform grid distribution with 402 grid points corresponding in the relevant region to a uniform grid of 1000 cells between 0 and 1; the optimal solution is obtained for an artificial viscosity parameter* $\alpha_{sh} = 2.5$ *and a Crank-Nicolson factor* $\vartheta_{CN} = 0.6$



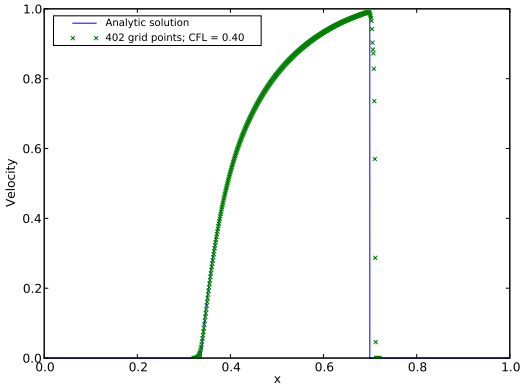
(a) *Relativistic Density* $D = \rho\Gamma$



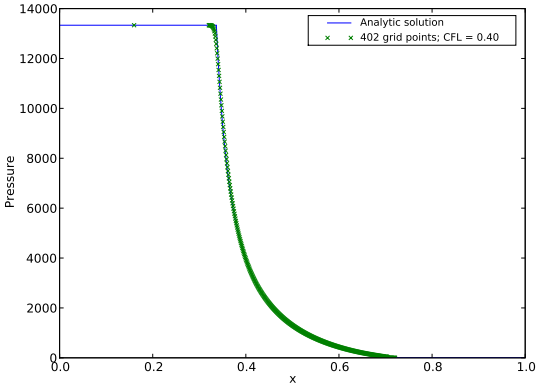
(b) *Lorentz Factor* $\Gamma = \frac{u^t}{c}$



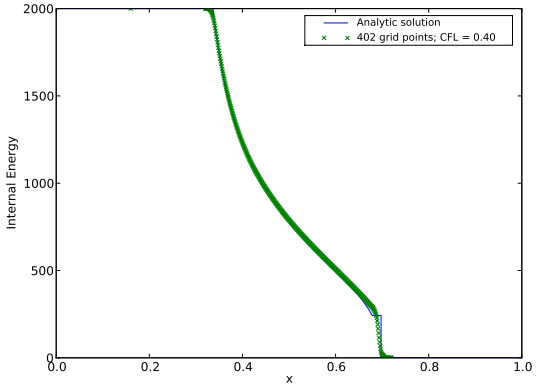
(c) *Density* ρ



(d) *Velocity* V

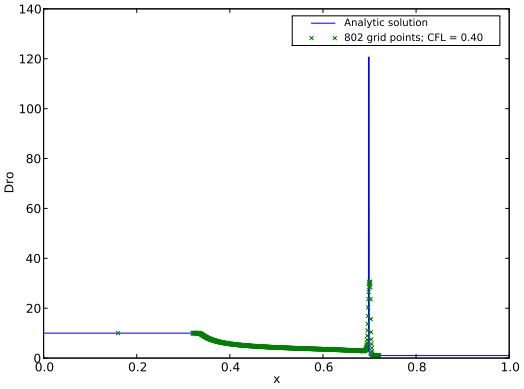


(e) *Pressure* P

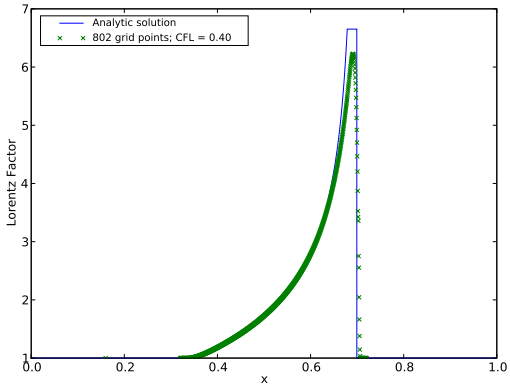


(f) *Internal Energy* $\epsilon = \frac{1}{\gamma-1}\frac{P}{\rho}$

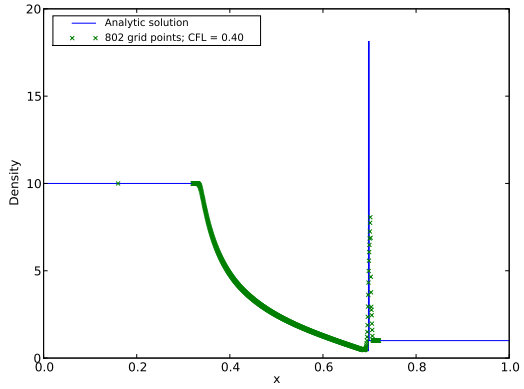
Figure 5.12: *Relativistic Shock Tube Problem with maximum Lorentz factor of approx. 6.8; non-uniform grid distribution with 402 grid points corresponding in the relevant region to a uniform grid of 1000 cells between 0 and 1; the optimal solution is obtained for an artificial viscosity parameter* $\alpha_{sh} = 1.0$ *and a Crank-Nicolson factor* $\vartheta_{CN} = 0.65$



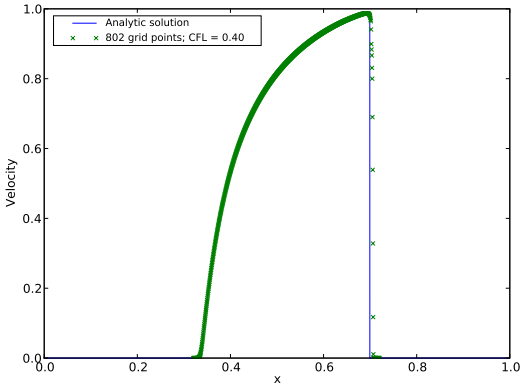
(a) *Relativistic Density* $D = \rho\Gamma$



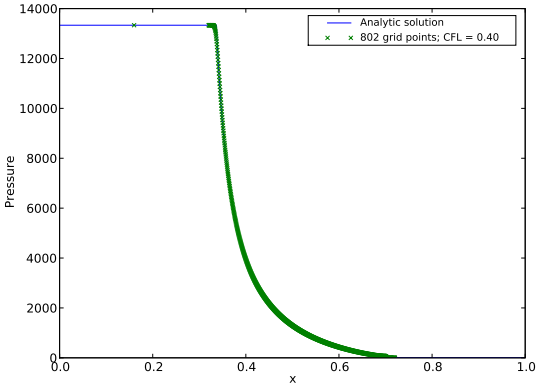
(b) *Lorentz Factor* $\Gamma = \frac{u^t}{c}$



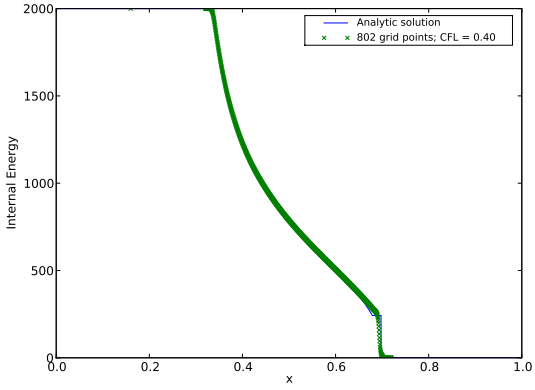
(c) *Density* ρ



(d) *Velocity* V

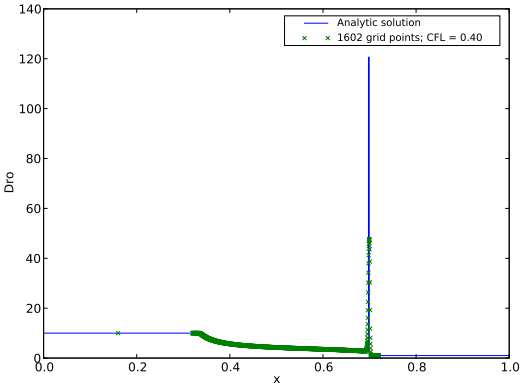


(e) *Pressure* P

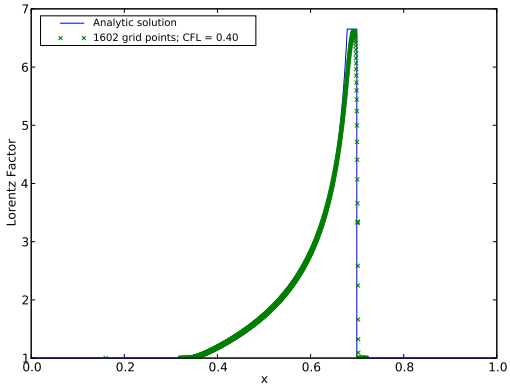


(f) *Internal Energy* $\epsilon = \frac{1}{\gamma-1}\frac{P}{\rho}$

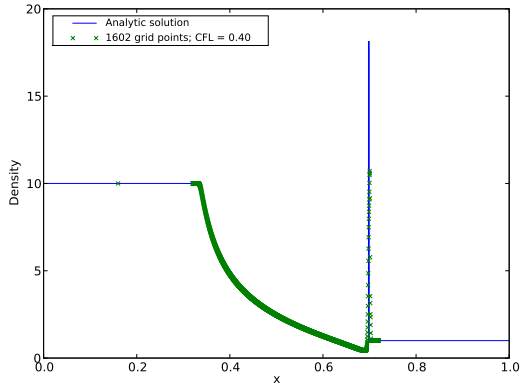
Figure 5.13: *Relativistic Shock Tube Problem with maximum Lorentz factor of approx. 6.8; non-uniform grid distribution with 802 grid points corresponding in the relevant region to a uniform grid of 2000 cells between 0 and 1; the optimal solution is obtained for an artificial viscosity parameter* $\alpha_{sh} = 2.5$ *and a Crank-Nicolson factor* $\vartheta_{CN} = 0.6$



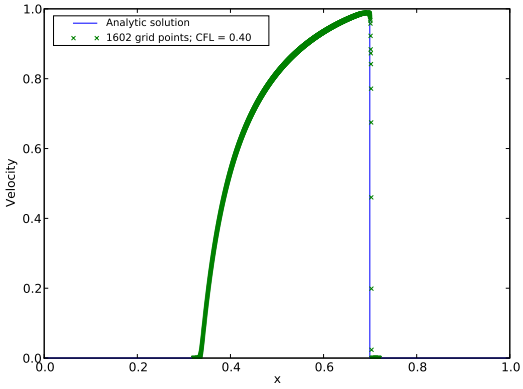
(a) *Relativistic Density* $D = \rho\Gamma$



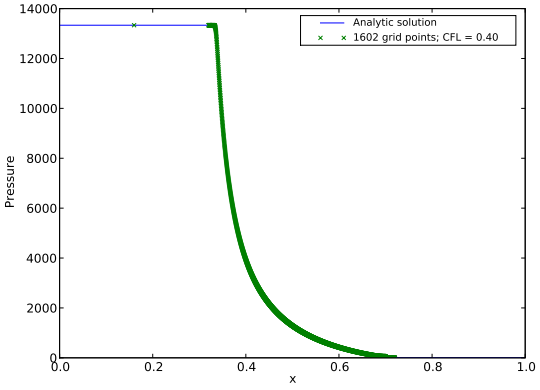
(b) *Lorentz Factor* $\Gamma = \frac{u^t}{c}$



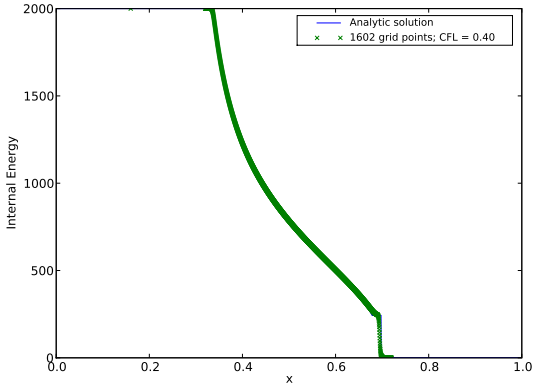
(c) *Density* ρ



(d) *Velocity* V

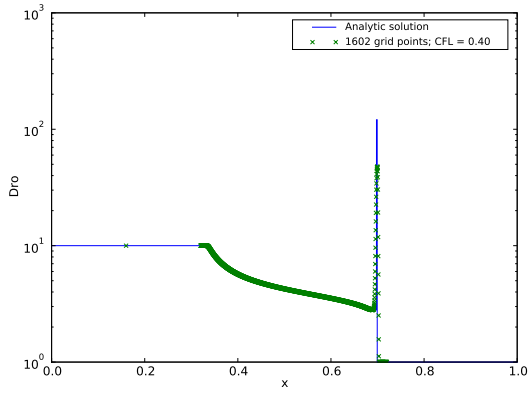


(e) *Pressure* P

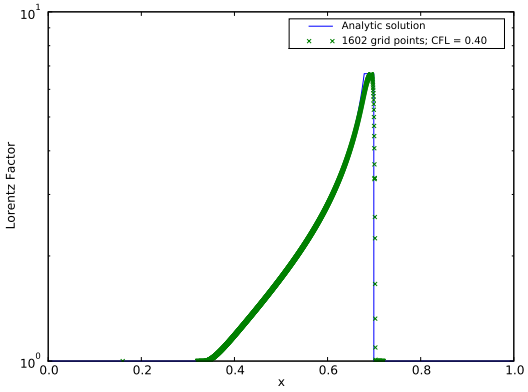


(f) *Internal Energy* $\epsilon = \frac{1}{\gamma-1}\frac{P}{\rho}$

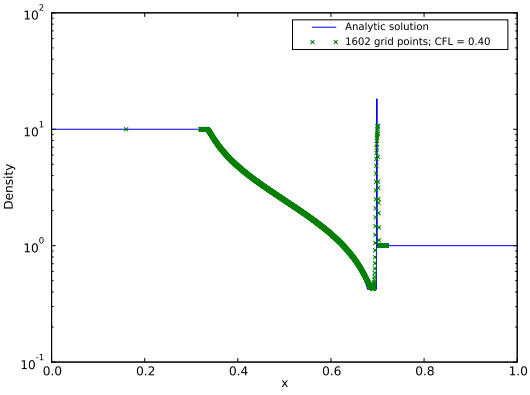
Figure 5.14: *Relativistic Shock Tube Problem with maximum Lorentz factor of approx. 6.8; non-uniform grid distribution with 1602 grid points corresponding in the relevant region to a uniform grid of 4000 cells between 0 and 1; the optimal solution is obtained for an artificial viscosity parameter* $\alpha_{sh} = 1.5$ *and a Crank-Nicolson factor* $\vartheta_{CN} = 0.6$



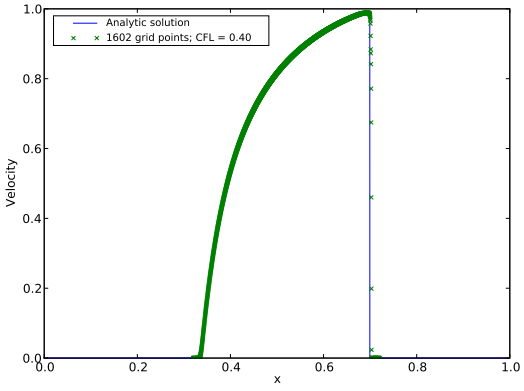
(a) *Relativistic Density* $D = \rho\Gamma$



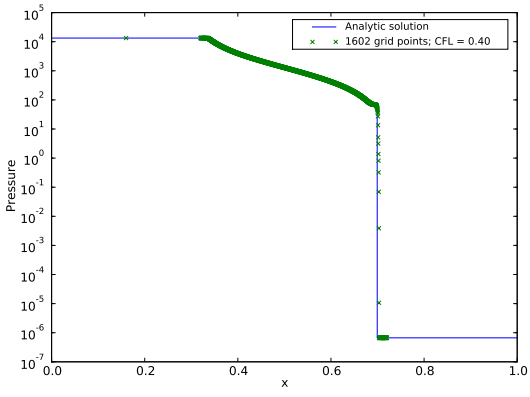
(b) *Lorentz Factor* $\Gamma = \frac{u^t}{c}$



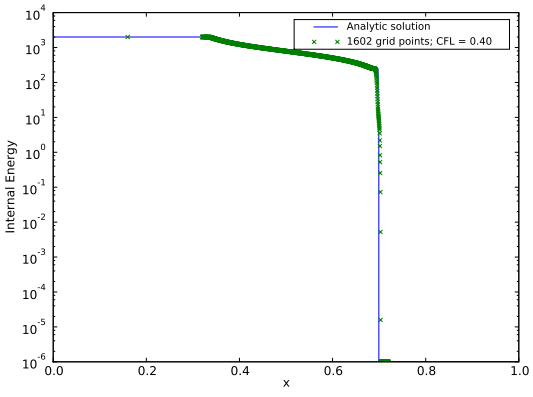
(c) *Density* ρ



(d) *Velocity V*

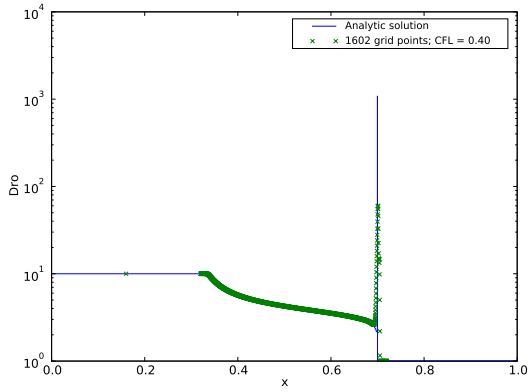


(e) *Pressure P*

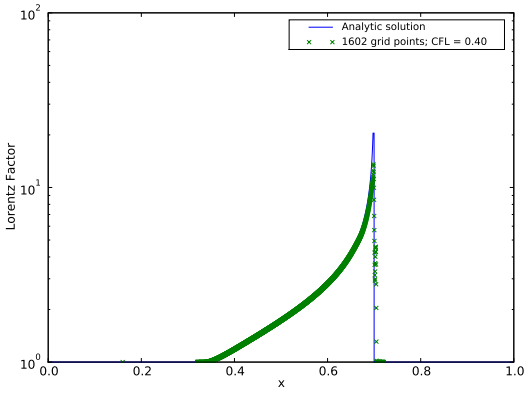


(f) *Internal Energy* $\epsilon = \frac{1}{\gamma-1}\frac{P}{\rho}$

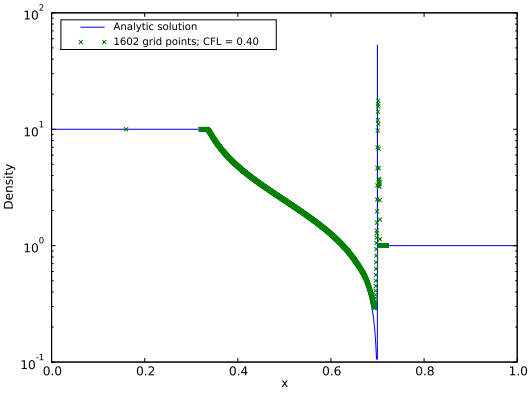
Figure 5.15: *Relativistic Shock Tube Problem with maximum Lorentz factor of approx. 6.8;* **logarithmic plots (except for velocity)***; non-uniform grid distribution with 1602 grid points corresponding in the relevant region to a uniform grid of 4000 cells between 0 and 1; the optimal solution is obtained for an artificial viscosity parameter* $\alpha_{sh} = 1.5$ *and a Crank-Nicolson factor* $\vartheta_{CN} = 0.6$



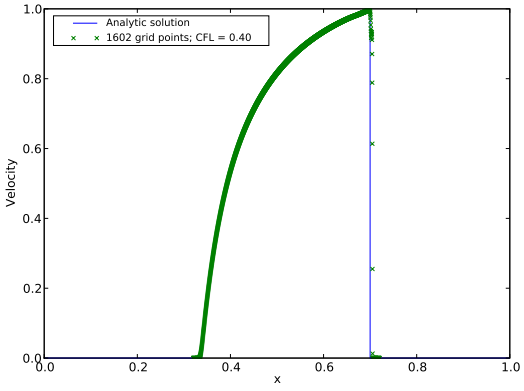
(a) *Relativistic Density* $D = \rho\Gamma$



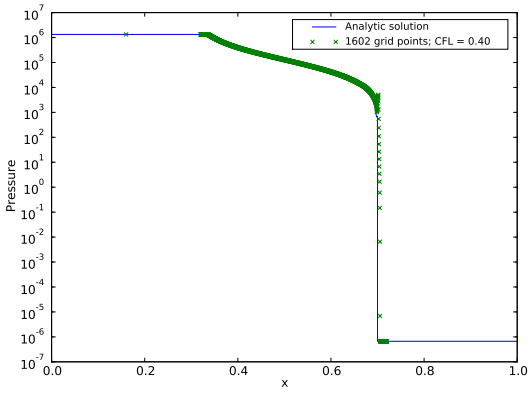
(b) *Lorentz Factor* $\Gamma = \frac{u^t}{c}$



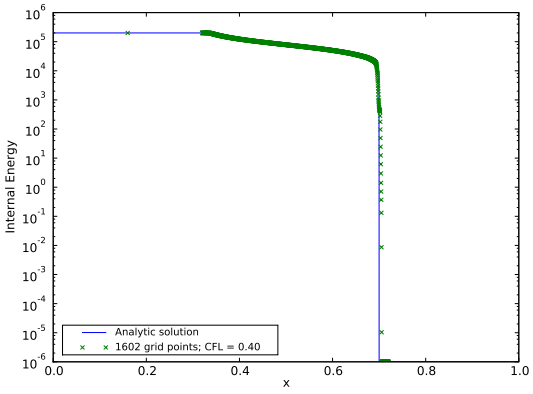
(c) *Density* ρ



(d) *Velocity V*



(e) *Pressure P*



(f) *Internal Energy* $\epsilon = \frac{1}{\gamma-1}\frac{P}{\rho}$

Figure 5.16: *Relativistic Shock Tube Problem with maximum Lorentz factor of approx. 21;* **logarithmic plots (except for velocity)***; non-uniform grid distribution with 1602 grid points corresponding in the relevant region to a uniform grid of 4000 cells between 0 and 1; the optimal solution is obtained for an artificial viscosity parameter* $\alpha_{sh} = 0.125$ *and a Crank-Nicolson factor* $\vartheta_{CN} = 0.625$

5.1.4 General Relativistic Spherical Accretion

The general relativistic steady-state spherical symmetric accretion onto a Schwarzschild black hole or neutron star is essentially a one dimensional radial problem to test the right implementation of the gravity terms in the general relativistic solver.

Bondi (1952) described the analytical solution to the Newtonian spherical accretion and Michel (1972) extended this to the general relativistic spherical accretion onto a Schwarzschild black hole. The solution is similar to that of Parker's solar wind model as it is also a sonic point flow, but here in the opposite direction: from outwards to inwards. From the outer subsonic inflow there is a transition to a supersonic inflow at a critical radius, the so-called critical or sonic point.

In figure 5.17c the analytic solution of fig. 1 of Michel (1972) is reproduced (using a self-written python-script).

For the simulation the same parameters as in Michel (1972): fig. 1 are used: The velocity at the critical point $v_c = 4.6 \cdot 10^{-5}\, c$, which is located at the radius $r_c = 2\, AU \approx 2.29 \cdot 10^8\, r_g$. The mass of the central object is $M_{BH} \approx 0.884\, M_\odot$. The adiabatic (polytropic) index is that of an ultra-relativistic plasma $\gamma = \frac{4}{3}$ ($n = \frac{1}{\gamma - 1} = 3$). This gives an asymptotic gas temperature at infinity of $10^{-9}\, m_p\, c^2 / k_B \approx 10^{-9} \cdot 10^{13}\,\mathrm{K} = 10^4\,\mathrm{K}$

For the simulations the innermost part, from $r_{in} = 2.2\, r_g$ to $r_{out} = 10.0\, r_g$, of this solution is simulated using the polytropic equation of state as in the analytic solution only solving the continuity equation and the radial momentum equation coupled.

Some other simulations were performed using the ideal gas law and therefore additionally solving the internal energy equation.

At the outer radial boundary at $r = r_{out} = 10.0\, r_g$ the analytic values of the conservative variables were set (taking into account the staggered grid structure) for a uniform grid of 64 cells in code units:

$$D_{Rout} = 10.240 \tag{5.21}$$
$$m_{Rout} = -5.785 \tag{5.22}$$
$$\epsilon^d_{Rout} = 2.470 \cdot 10^{-4} \tag{5.23}$$

As initial conditions a homogeneous density of $\rho \approx 8.2 \cdot 10^{-9}\, \frac{\mathrm{g}}{\mathrm{cm}^3}$ and a temperature of $T \approx 4.355 \cdot 10^7 \mathrm{K}$, which gives a pressure of $P \approx 6.0 \cdot 10^7\, \mathrm{g}\, \frac{\mathrm{cm}^2}{\mathrm{s}^2} = 6.0 \cdot 10^7\, \frac{\mathrm{dyn}}{\mathrm{cm}^2} = 60\, bar = 6.0 \cdot 10^6\,\mathrm{Pa}$, corresponding to the analytic outer boundary values at $10\, r_g$ and zero velocity was used. Here the pressure was calculated from the temperature using the relation $P = \frac{\mathcal{R}_{\mathrm{gas}}}{\mu_{\mathrm{gas}}} \rho T$ with the molecular mass (molecular weight) $\mu_{\mathrm{gas}} = 0.5 \cdot 10^{-3}\mathrm{kg/mol} = 0.5\mathrm{g/mol}$ (for ionized hydrogen) and the gas constant $\mathcal{R}_{\mathrm{gas}} = 8.34472\mathrm{J/(Kmol)} = 8.314472 \cdot 10^7\mathrm{erg/(Kmol)}$.

Whereas at the inner boundary at $r = 2.2\, r_g = 1.1\, r_{SS}$ zero gradient of the primitive variables ρ, V_{Er}, P were used (zero gradient of the conservative variables gives not so good results).

The used scaling variables are:

$$R_{Dim} = 130566.75\,\text{cm} = r_g \tag{5.24}$$

$$\rho_{Dim} = 1.0 \cdot 10^{-9}\,\frac{\text{g}}{\text{cm}^3} \tag{5.25}$$

$$V_{Dim} = 29979245800.0\,\frac{\text{cm}}{\text{s}} = c \tag{5.26}$$

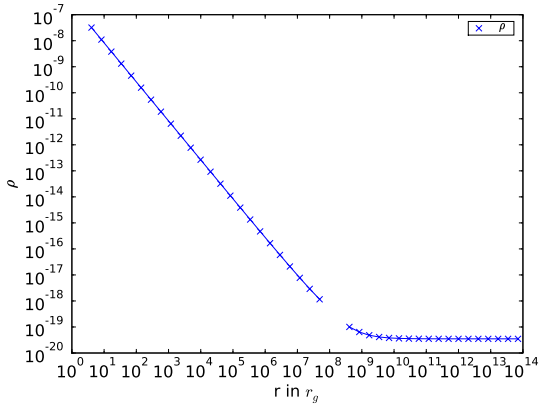
$$T_{Dim} = 1.0 \cdot 10^{9}\,\text{K} \tag{5.27}$$

Time is then scaled by $time_{Dim} = R_{Dim}/V_{Dim} = \frac{r_g}{c} \approx 4.355 \cdot 10^{-6}\text{s} = 4.355\ \mu s$, pressure and energy densities by $\rho_{Dim} \cdot V_{Dim}^2 = \rho_{Dim} \cdot c^2 \approx 8.988 \cdot 10^{11}\frac{\text{g}}{\text{cms}^2} = 8.988 \cdot 10^{11}\frac{\text{erg}}{\text{cm}^3} = 8.988 \cdot 10^{10}\frac{\text{J}}{m^3}$.

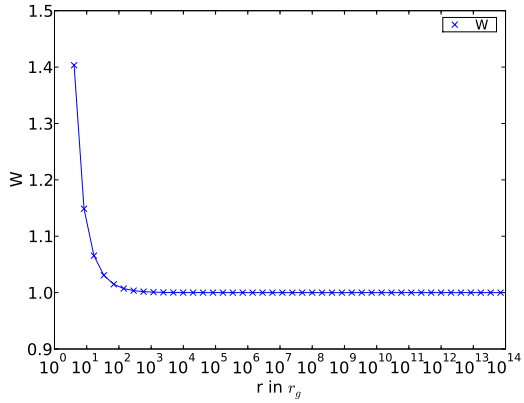
Figures 5.18, 5.19 and 5.20 show the excellent numerical results in comparison with the analytic solution for 64, 128 and 256 uniform grid cells for the polytropic equation of state and figure 5.21 the results for 256 uniform grid cells in the case of an ideal gas law, where additionally the internal energy equation was solved.

In figure 5.22 one sees the huge difference in the vicinity of the black hole, where the redshift factor α plays a big role, between the Boyer-Lindquist coordinate observer located at infinity fixed to the grid and the local Euler frame observer: The pseudo Lorentz factor of the Boyer-Lindquist observer $\Gamma = \frac{u^t}{c}$ and the corresponding velocity V_r and the local Lorentz factor $W = \alpha\frac{u^t}{c}$ in the Euler frame with the velocity V_{Er}, which gives from the point of the fluid a more natural and intuitive description: the radial velocity in the local Euler frame v_{Er} will approach the speed of light c at the event horizon, which is located at the Schwarzschild radius $r_{SS} = 2\,r_g$; so there the local Lorentz factor W will be infinity.

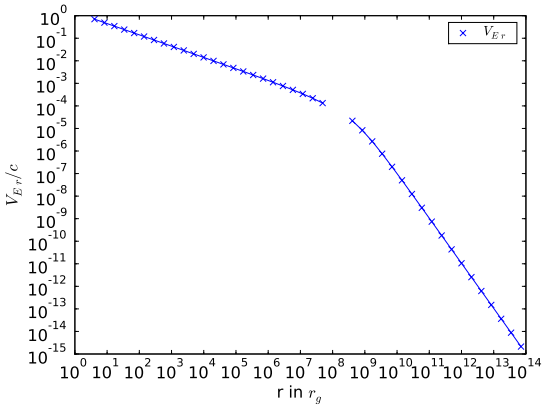
Some log variables are plotted in figure 5.23: In figure 5.23a the maximum of the moduli of the residuum of m is plotted, which is a measure of how good the linear equation system is solved. Note that here the values are always below 10^{-5}, the requested accuracy set by the the 'acc' option in the SolMethod parameter in the parameter file. In figure 5.23b the maximum of the moduli of the RHS of the radial momentum equation is plotted versus the simulated physical time, which is a measure of how far away the solution is from the steady-state. If this parameter is tiny the steady-state solution is reached.



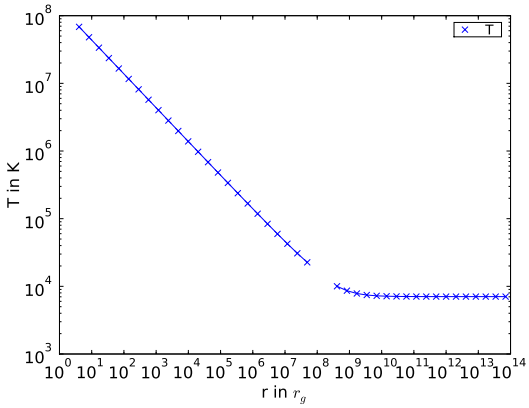
(a) *Density* ρ



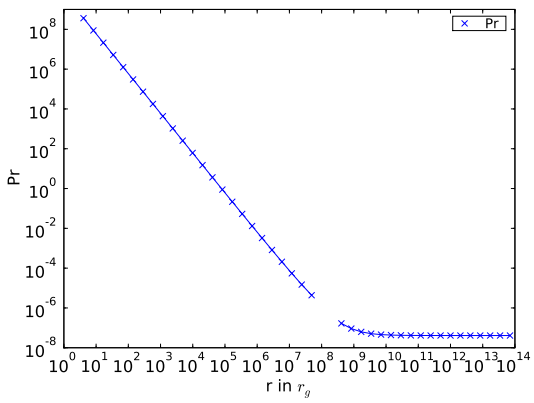
(b) *Lorentz Factor* $W = \alpha\Gamma = \alpha\frac{u^t}{c}$



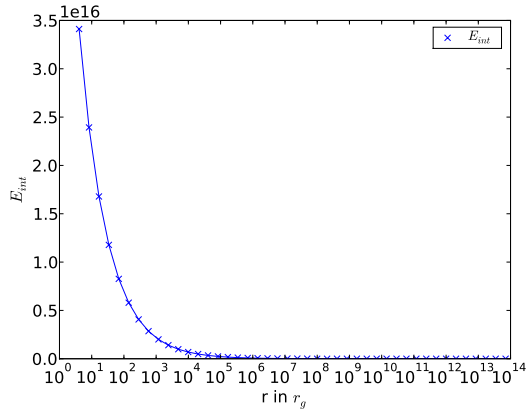
(c) *Velocity* V_{Er}



(d) *Temperature* T

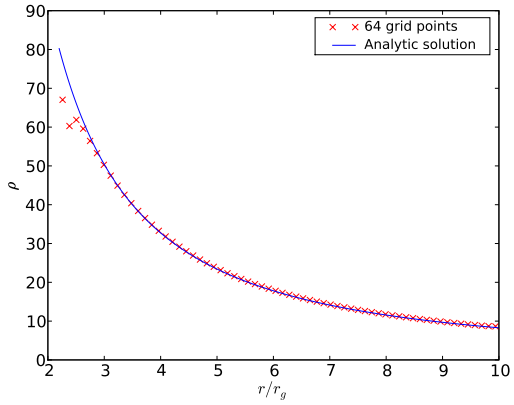


(e) *Pressure* P

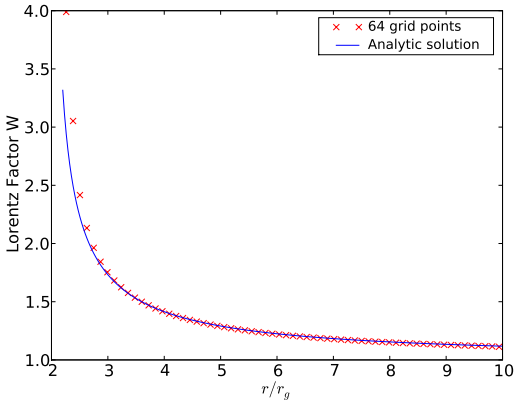


(f) *Internal Energy* $\epsilon = \frac{1}{\gamma-1}\frac{P}{\rho}$

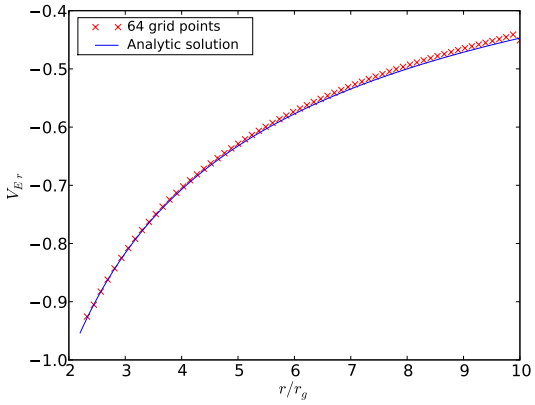
Figure 5.17: *General Relativistic Spherical Accretion analytic solution of a own written python script reproducing the results of Michel (1972): fig. 1*



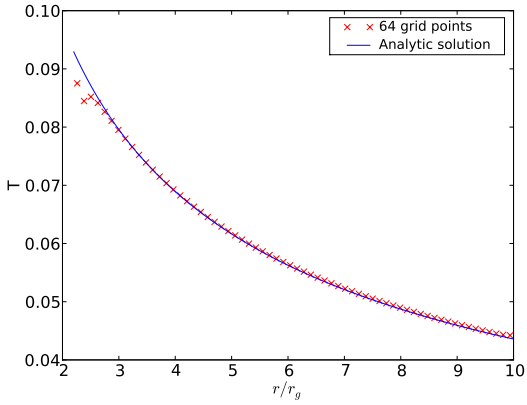
(a) *Density* ρ



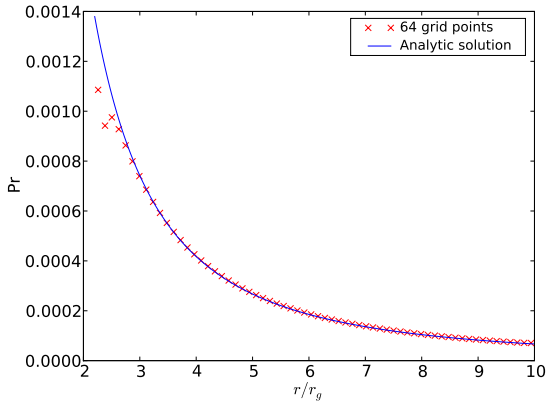
(b) *Lorentz Factor* $W = \alpha\Gamma = \alpha\frac{u^t}{c}$



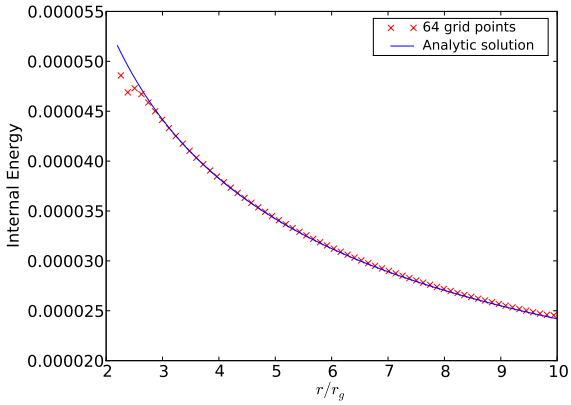
(c) *Velocity* V_{Er}



(d) *Temperature* T

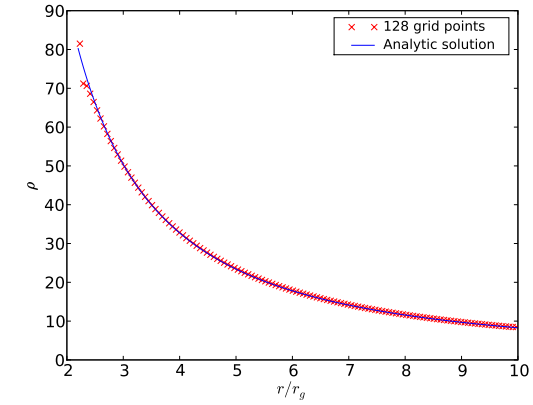


(e) *Pressure* P

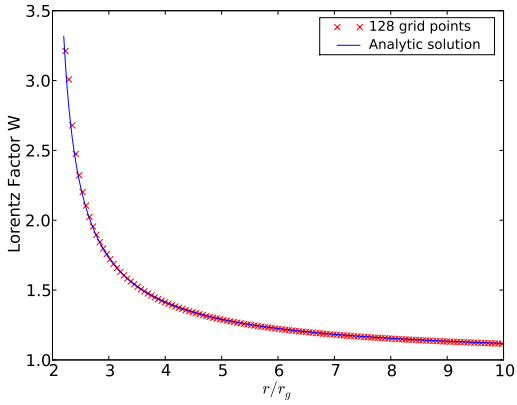


(f) *Internal Energy* $\epsilon = \frac{1}{\gamma-1}\frac{P}{\rho}$

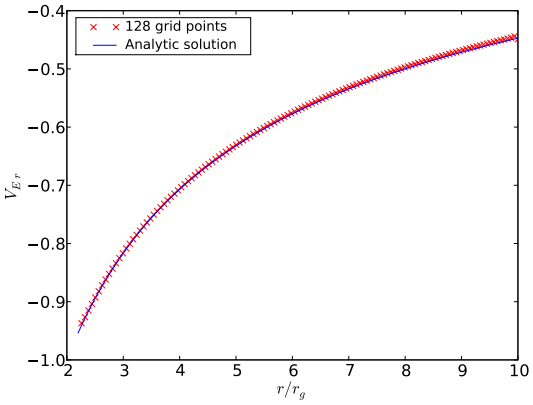
Figure 5.18: *General Relativistic Spherical Accretion: analytic and numerical solution of Astro-GRIPS using the polytropic equation of state and* **64 uniform grid cells**



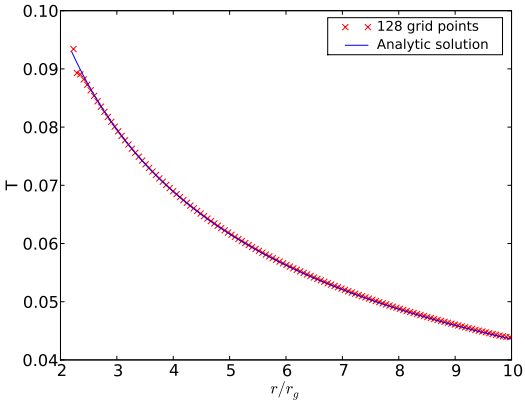
(a) *Density* ρ



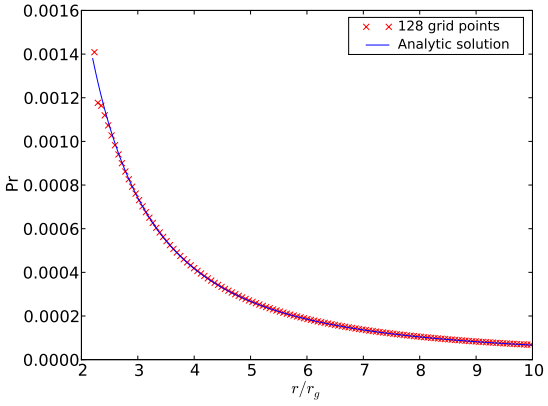
(b) *Lorentz Factor* $W = \alpha\Gamma = \alpha\frac{u^t}{c}$



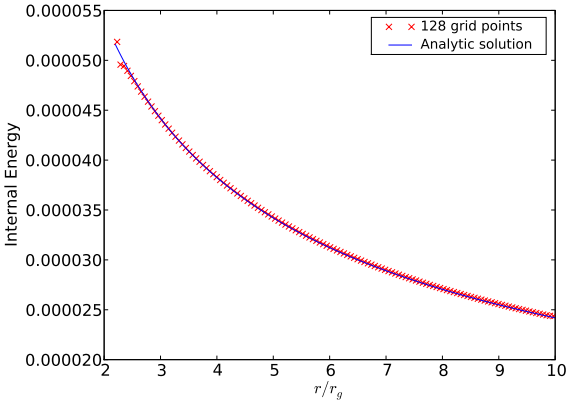
(c) *Velocity* V_{Er}



(d) *Temperature* T

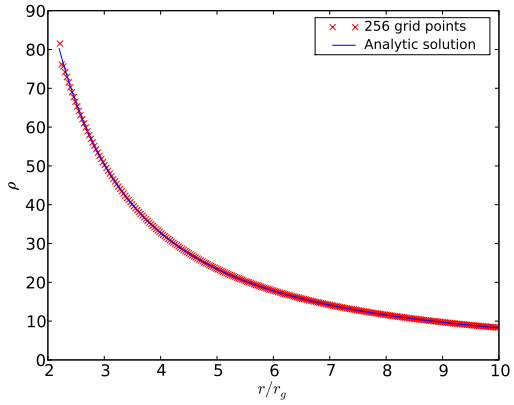


(e) *Pressure* P

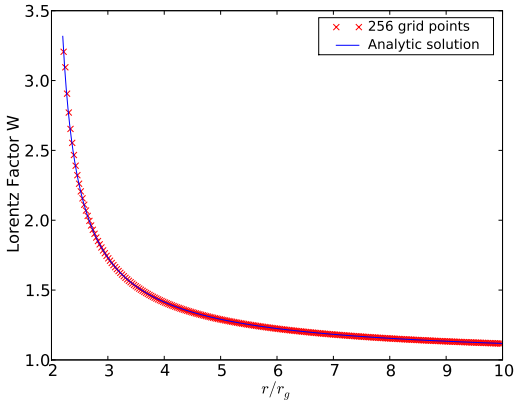


(f) *Internal Energy* $\epsilon = \frac{1}{\gamma-1}\frac{P}{\rho}$

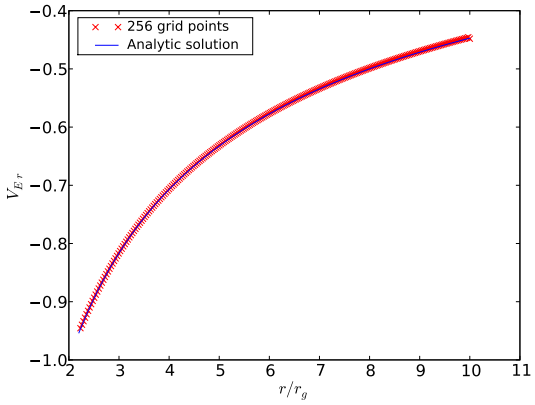
Figure 5.19: *General Relativistic Spherical Accretion: analytic and numerical solution of Astro-GRIPS using the polytropic equation of state and* **128 uniform grid cells**



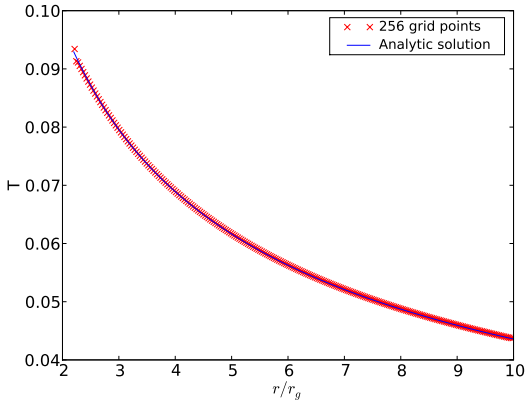
(a) *Density* ρ



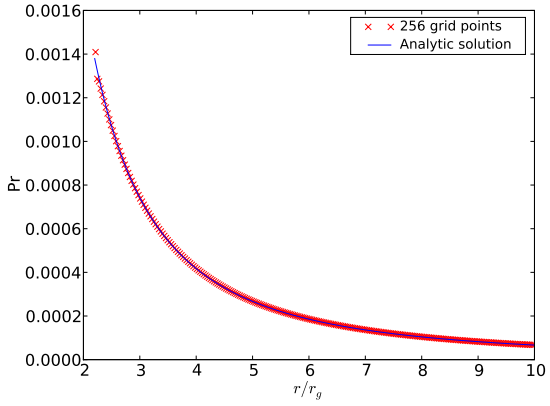
(b) *Lorentz Factor* $W = \alpha\Gamma = \alpha\frac{u^t}{c}$



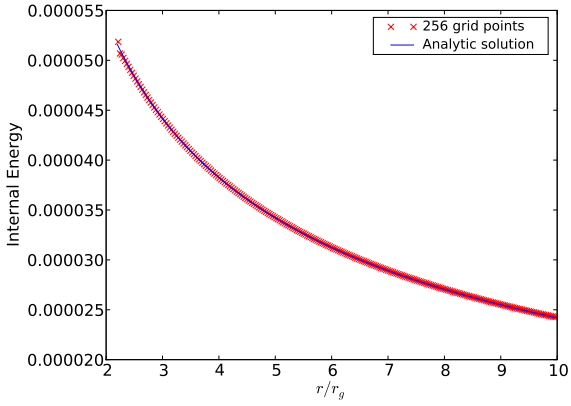
(c) *Velocity* V_{Er}



(d) *Temperature* T

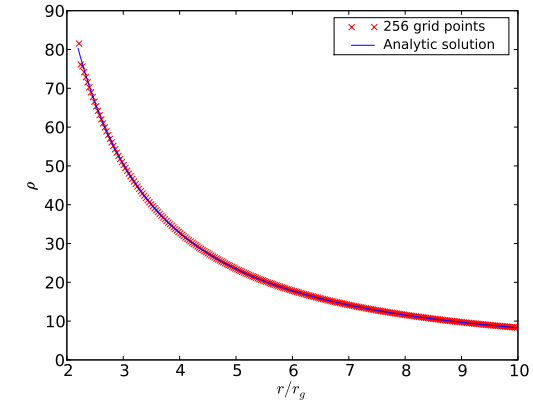


(e) *Pressure* P

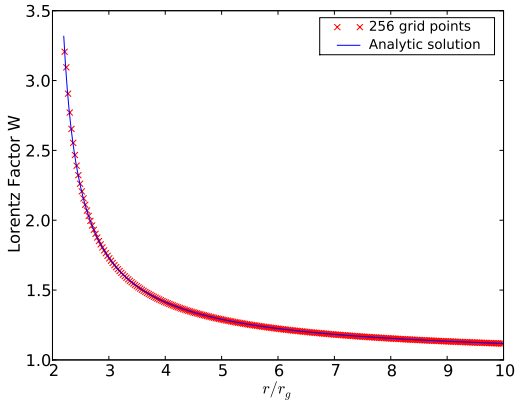


(f) *Internal Energy* $\epsilon = \frac{1}{\gamma-1}\frac{P}{\rho}$

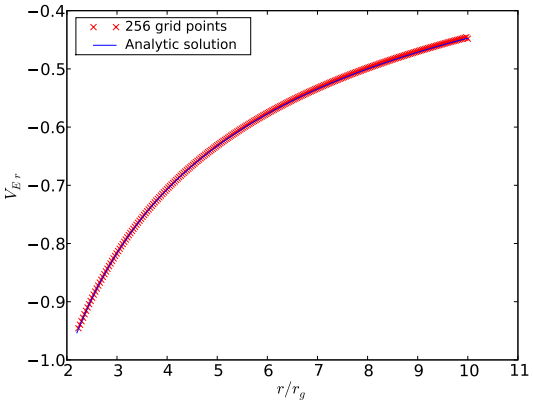
Figure 5.20: *General Relativistic Spherical Accretion: analytic and numerical solution of Astro-GRIPS using the polytropic equation of state and* **256 uniform grid cells**



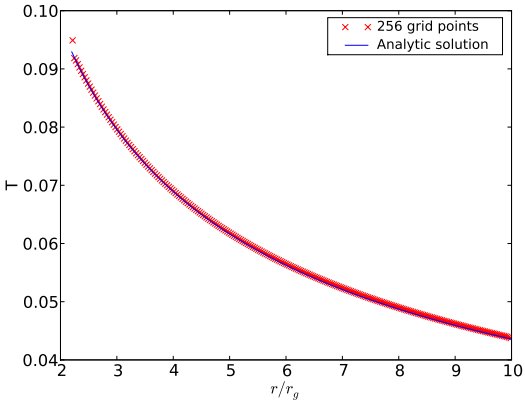
(a) *Density* ρ



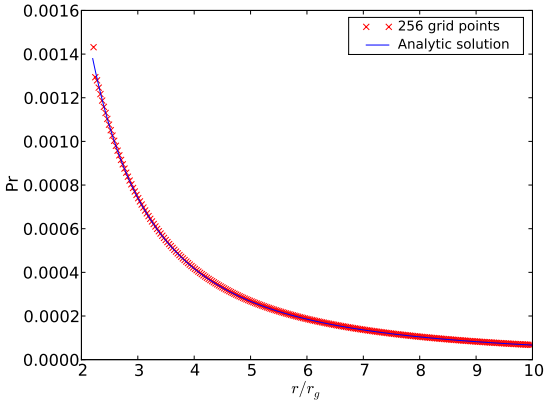
(b) *Lorentz Factor* $W = \alpha\Gamma = \alpha\frac{u^t}{c}$



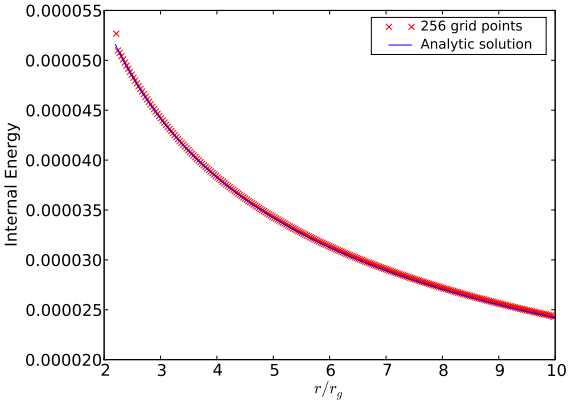
(c) *Velocity* V_{Er}



(d) *Temperature* T

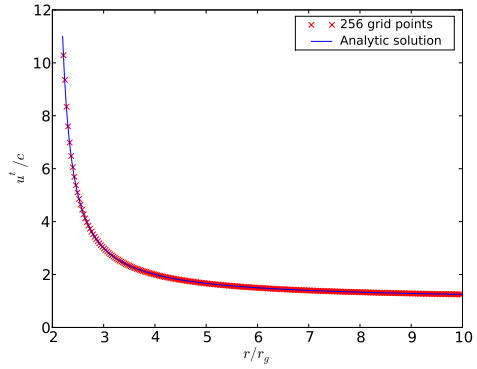


(e) *Pressure* P

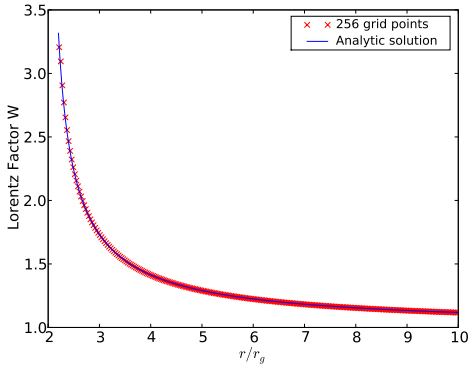


(f) *Internal Energy* $\epsilon = \frac{1}{\gamma-1}\frac{P}{\rho}$

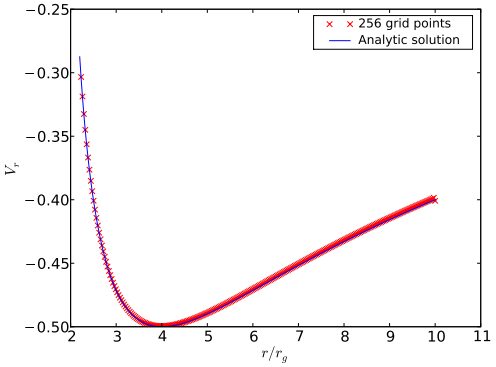
Figure 5.21: *General Relativistic Spherical Accretion: analytic and numerical solution of Astro-GRIPS using the* **ideal gas law** *and the internal energy equation and* **256 uniform grid cells**



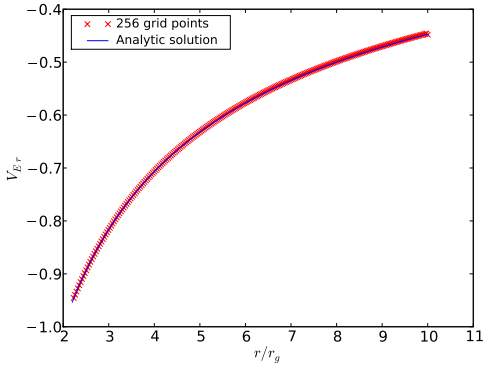
(a) *Pseudo Lorentz factor* $\Gamma = \frac{u^t}{c}$.



(b) *Lorentz factor* $W = \alpha \frac{u^t}{c}$.

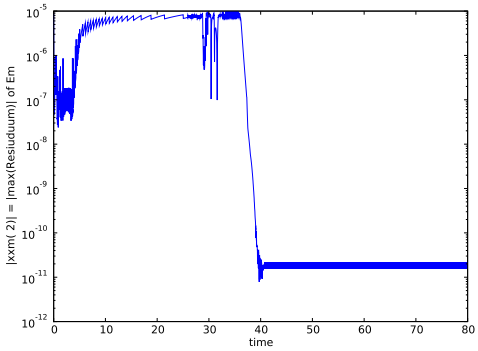


(c) *Velocity* V_r.

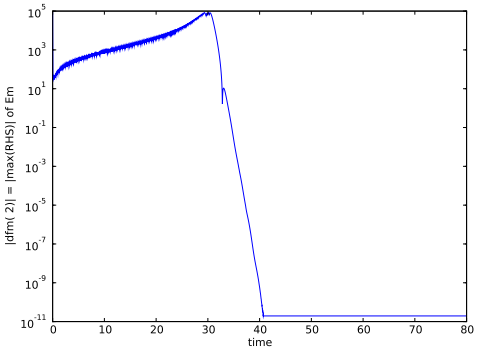


(d) *Velocity* V_{Er} *of the local Euler frame.*

Figure 5.22: *The differences between the Boyer-Lindquist coordinate observer frame, an observer at infinity fixed to the grid, (plots on the left) and the local Euler fluid frame (plots on the right).*



(a) *The maximum of the moduli of the residuum (of m): this is a measure of the accuracy the linear equation system is solved.*



(b) *The maximum of the moduli of the RHS of m: a measure of how far away the solution is from the steady-state.*

Figure 5.23: *Some log variables for the simulation run of the general relativistic spherical accretion with polytropic equation of state and 256 grid cells.*

5.2 More-Dimensional Problems

5.2.1 Taylor Couette Flow

The Taylor-Couette Flow problem is a well known more-dimensional problem in computational fluid dynamics to test the (Newtonian) Navier-Stokes equations (without gravity).

In the usual Taylor–Couette flow a viscous fluid is moving between two rotating cylinders. At low angular velocities, where the fluid shows a circular Couette flow, this setup was used by Couette to determine the viscosity of fluids.

Taylor (1923) described the instabilities, which occur for higher angular velocities of the inner cylinder in the circular Couette flow. First one gets Taylor vortices, then more complicated wavy vortex flows and at a certain high enough Reynolds number there is the onset of turbulence.

Here a Taylor-Couette Flow is simulated in 3D axi-symmetry between two concentric spheres instead of the usual cylinders. The inner radius is $R_{in} = 1$ and the outer radius is $R_{out} = 1.2$ and the latitude θ ranges only from 0 to $\frac{\pi}{2}$. For the simulation 144x1152 grid cells, a 3rd order spatial and 1st order temporal accurate scheme, an adiabatic index of $\gamma = \frac{5}{3}$ and at the polar axis and the equator reflecting boundaries are used and the inner and outer sphere is handled as a rigid wall, which means that there are also reflecting boundaries applied. The outer sphere is left static $\Omega_{out} = 0$, whereas the inner sphere is rotating with an angular velocity of $\Omega_{in} = 5$. The simulation is started with an initially linear Ω-distribution. The viscous fluid has an initial density of one and a temperature of 10, the turbulent viscosity parameter was set to $\alpha_{tr} = 0.05$ or 0.01. The scale of the turbulent velocity was set to be equal to the sound speed ($V_{tur} = V_s$) and the turbulent length scale to half the gap space: $L_{tur} = 0.1$.

The turbulent viscosity is defined as:

$$\nu = \alpha_{tr} V_{tur} L_{tur} \tag{5.28}$$

and the Reynolds number for this flow can be calculated as:

$$Re = \frac{|\Delta\Omega| R_{in} |\Delta R|}{\nu} = \frac{|\Omega_{out} - \Omega_{in}| R_{in} |R_{out} - R_{in}|}{\nu} \tag{5.29}$$

which for the aforementioned parameters is ≈63 for $\alpha_{tr} = 0.05$ for the non-turbulent case in fig. 5.24 and ≈316 for $\alpha_{tr} = 0.01$ for the turbulent case in fig. 5.25.

This problem was also used to test the parallelization of Astro-GRIPS, which is described in the section 4.5 about parallelization on page 123.

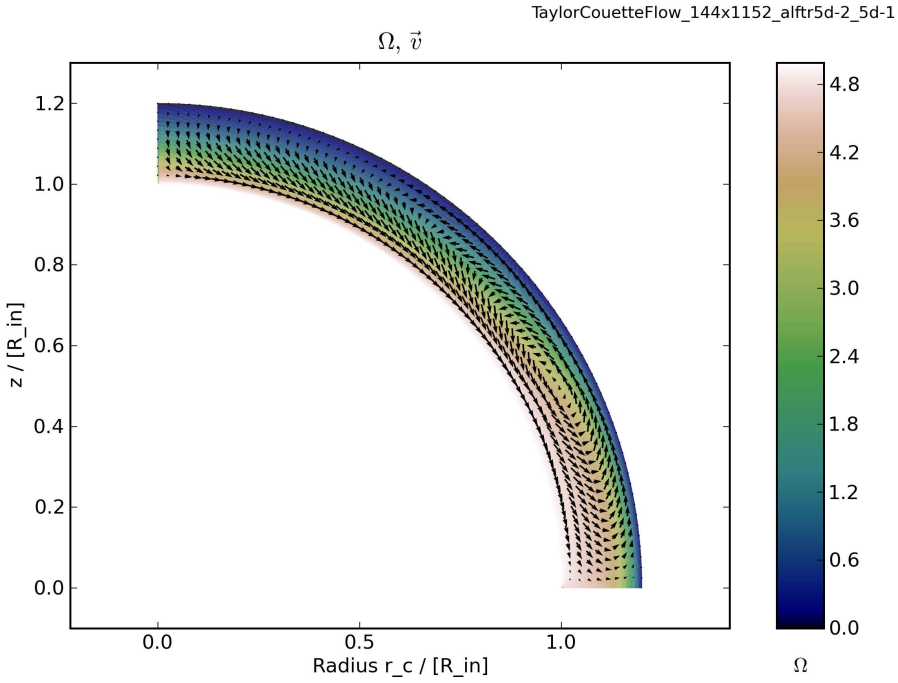


Figure 5.24: *Laminar Taylor-Couette Flow with $\alpha_{tr} = 0.05$ (Re≈63): Ω-distribution and velocity arrows (arctan scaling)*

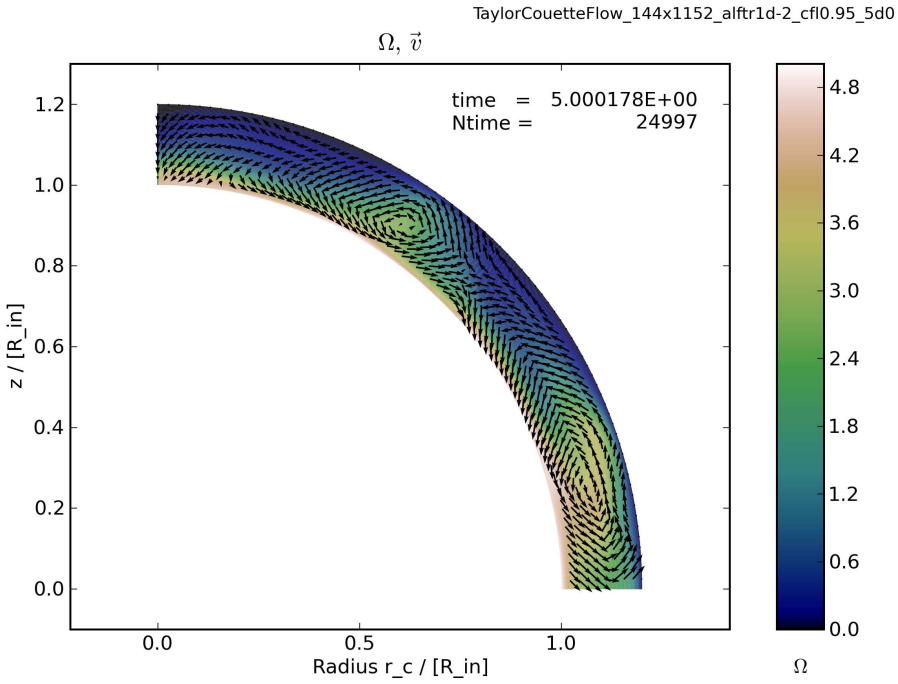


Figure 5.25: *Turbulent Taylor-Couette Flow with $\alpha_{tr} = 0.01$ (Re≈316): Ω-distribution and velocity arrows (arctan scaling)*

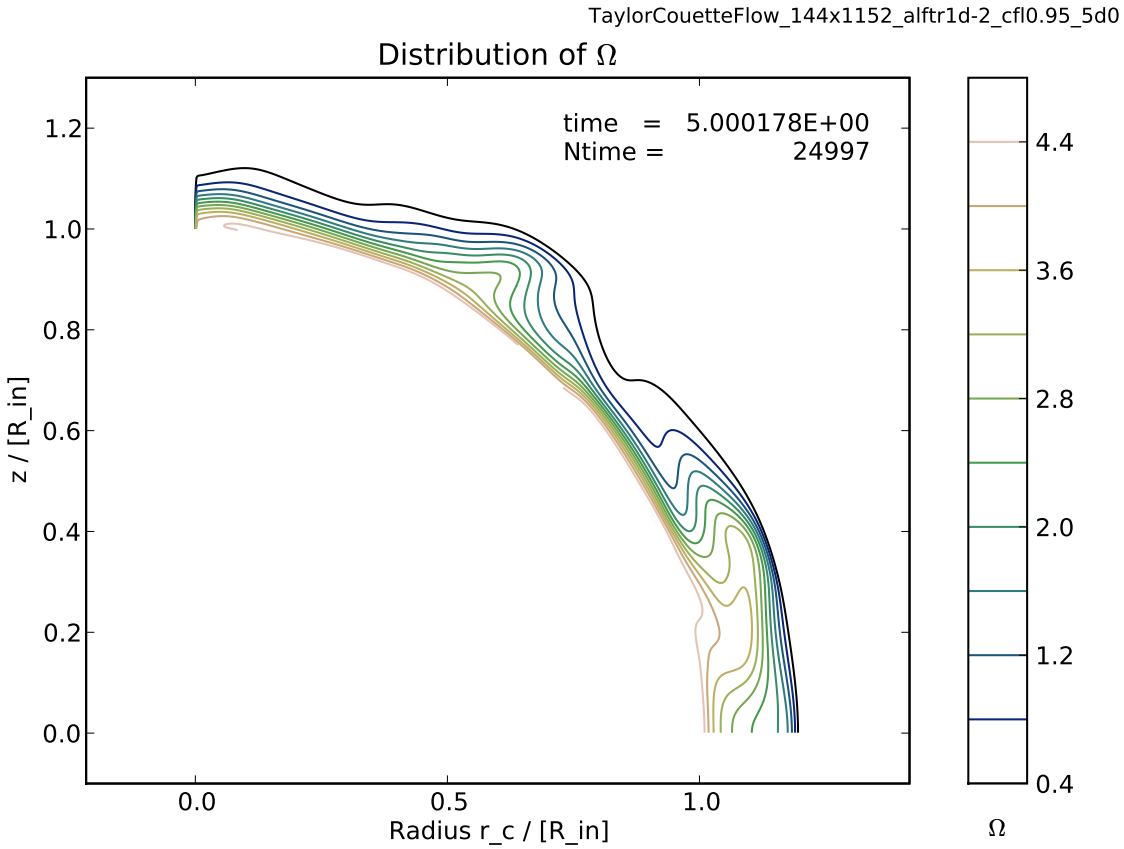


Figure 5.26: *Turbulent Taylor-Couette Flow between two concentric spheres with* $\alpha_{tr} = 0.01$ *(Re≈316):* Ω*-Contours*

5.2.2 General Relativistic Standing Shocks at Cold Discs around Black Holes

In this two dimensional problem a elsewise spherically infalling flow forms a standing shock at a cold disc around a Schwarzschild black hole. The plasma surrounding the disc is taken to be inviscid, thin, hot and non-rotating. This flow configuration is similar to the forward facing step problem in computational fluid dynamics.

For this simulation a cold and dense disc has been placed in the innermost equatorial region. It is a region, where no equations are solved and which acts like a reflecting barrier for the elsewise radial inflow, so a curved standing shock is formed.

The computational domain reaches from an inner radius at $2.2\,r_g = 1.1\,r_{SS}$ to an outer radius of $10.0\,r_g = 5.0\,r_{SS}$, where $R_g = \frac{GM}{c^2}$ is the gravitational radius and $R_{SS} = 2R_g$ the Schwarzschild radius, and in θ direction from zero (midplane) to 90 degrees (polar axis).

The same parameters as for the general relativistic spherical accretion (see 5.1.4) were used.

The cold disc, the region where nothing is calculated, extends out to a radius of $6.0\,r_g = 3.0\,r_{SS}$ and has an angle above the midplane of 10 degrees.

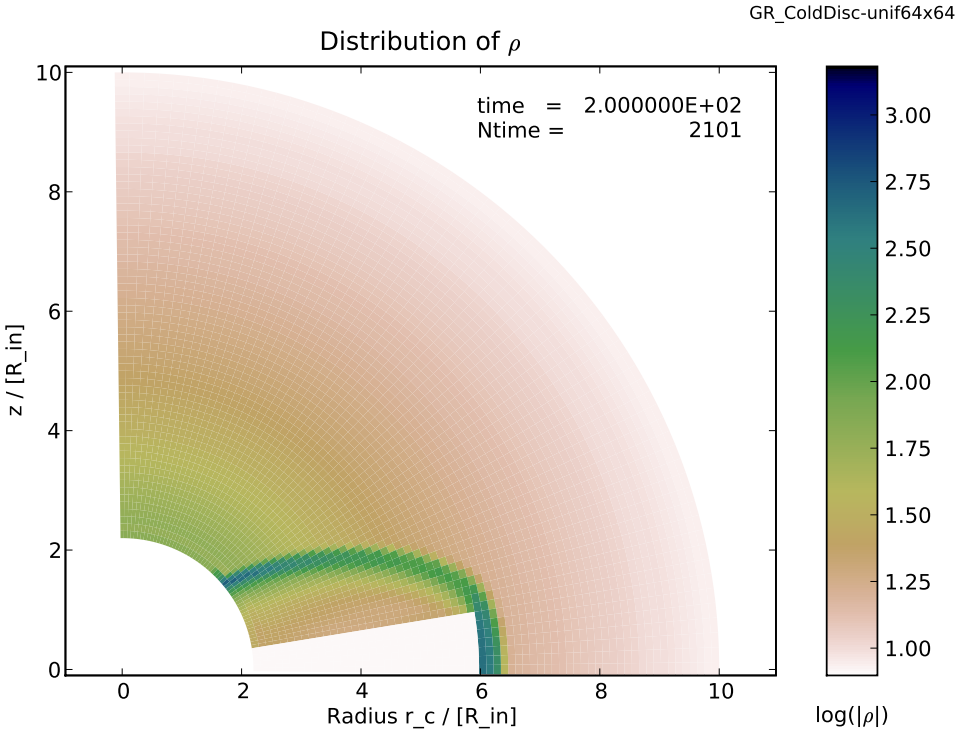
At the outer radial boundary there are set fixed boundary conditions as in the case of the general relativistic spherical accretion and the latitudinal velocity is set to zero there. At the inner radial boundary there are zero gradient boundary conditions for the conservative variables except for the latitudinal velocity which was fixed to zero. In the midplane there are set symmetric and at the polar axis are set axi-symmetric boundary conditions.

As equation of state an ideal gas law with an adiabatic coefficient of $\gamma = \frac{4}{3}$ for an ultra-relativistic plasma was used.

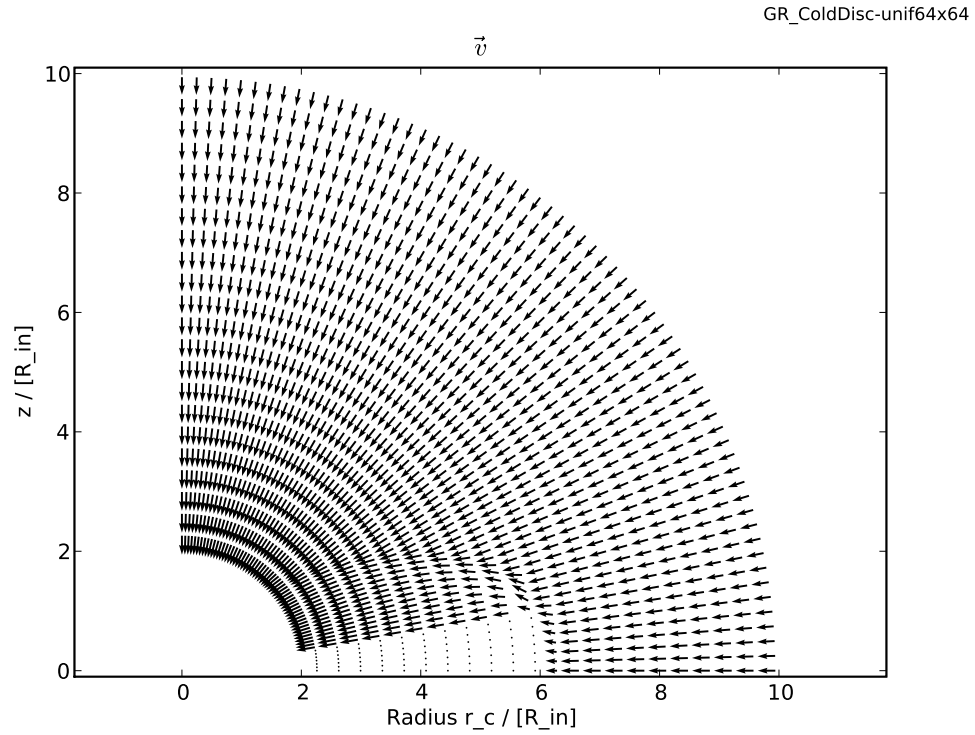
The steady state solution was obtained with a fully coupled system solved with the AFM method using a 3rd order spatial van Leer advection scheme and a Crank-Nicolson number $\vartheta_{CN} = 0.5$, which results in second order temporal accuracy.

Figures 5.27, 5.28 and 5.29 show the density, velocity arrows, velocity components of the local Euler frame, the pressure and the temperature distribution of the steady state solution on a uniform 64x64 grid. One clearly sees the standing shock around the cold disc.

These numerical results show that the simulation code is also capable of simulating 3D axi-symmetric general relativistic flows.

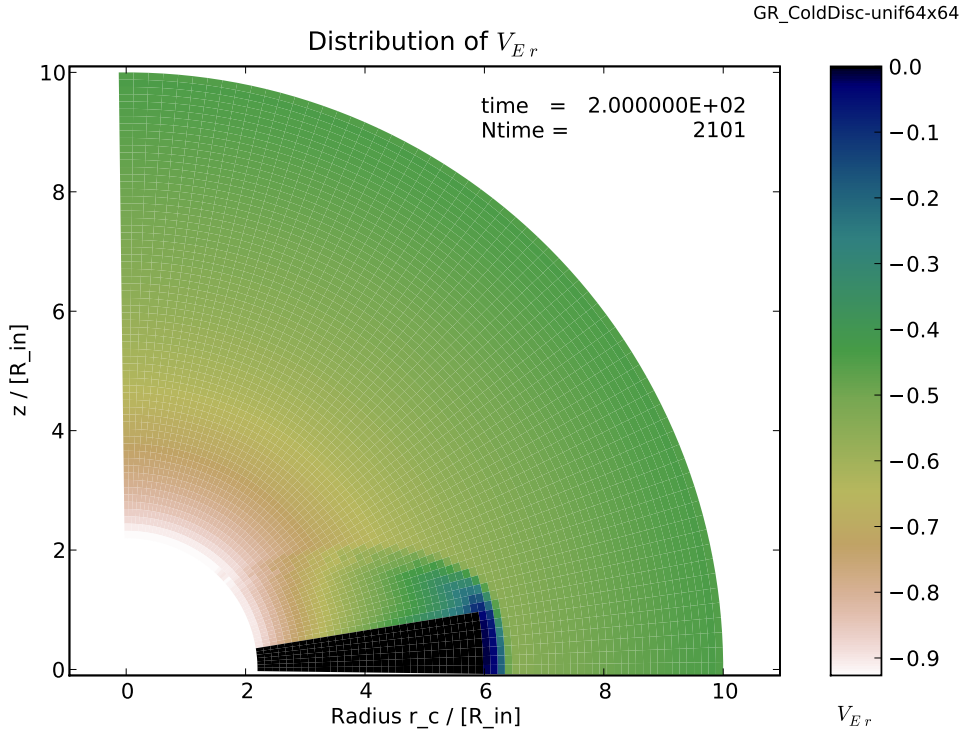


(a) *Density ρ in* $10^{-9}\ \frac{\mathrm{g}}{\mathrm{cm}^3}$

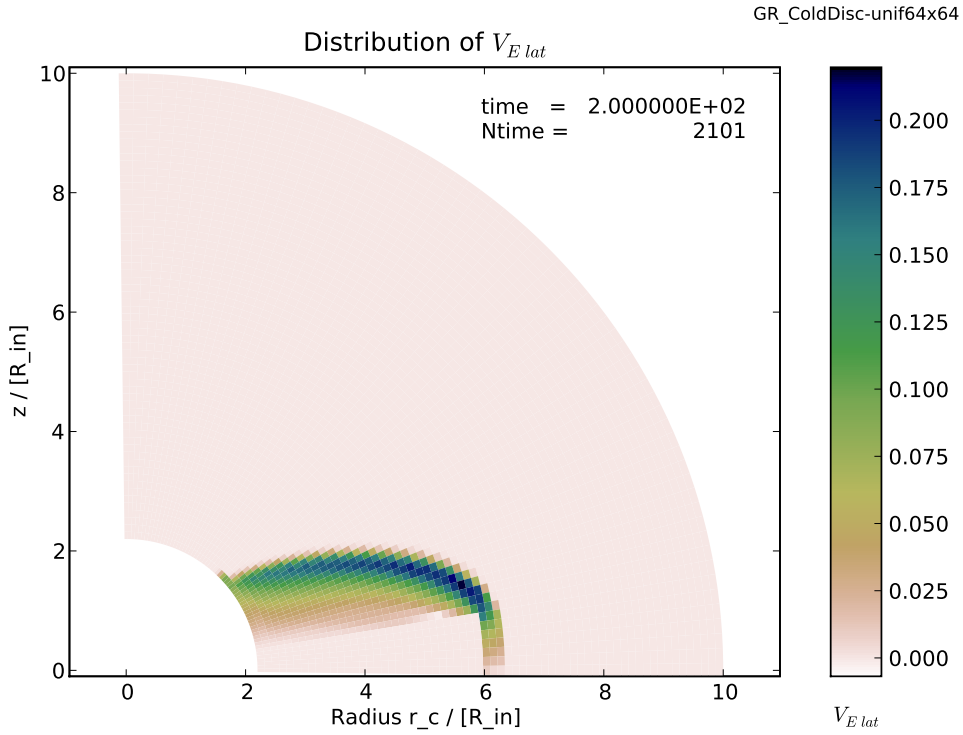


(b) *Velocity $\vec{V}_E$ (arctan scaling)*

Figure 5.27: *General Relativistic Standing Shock at a Cold Disc around a Schwarzschild Black Hole: Astro-GRIPS simulation with a uniform grid size of 64x64 cells: density in code units and velocity arrows (arctan scaling)*

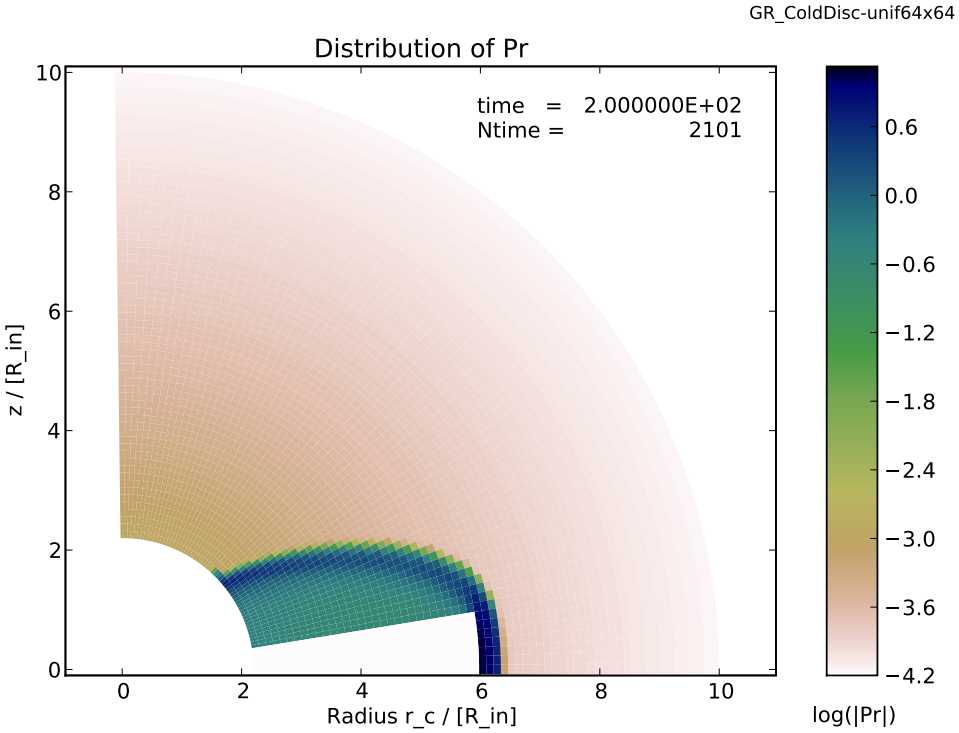


(a) *Radial Velocity V_{Er} in the local Euler frame*

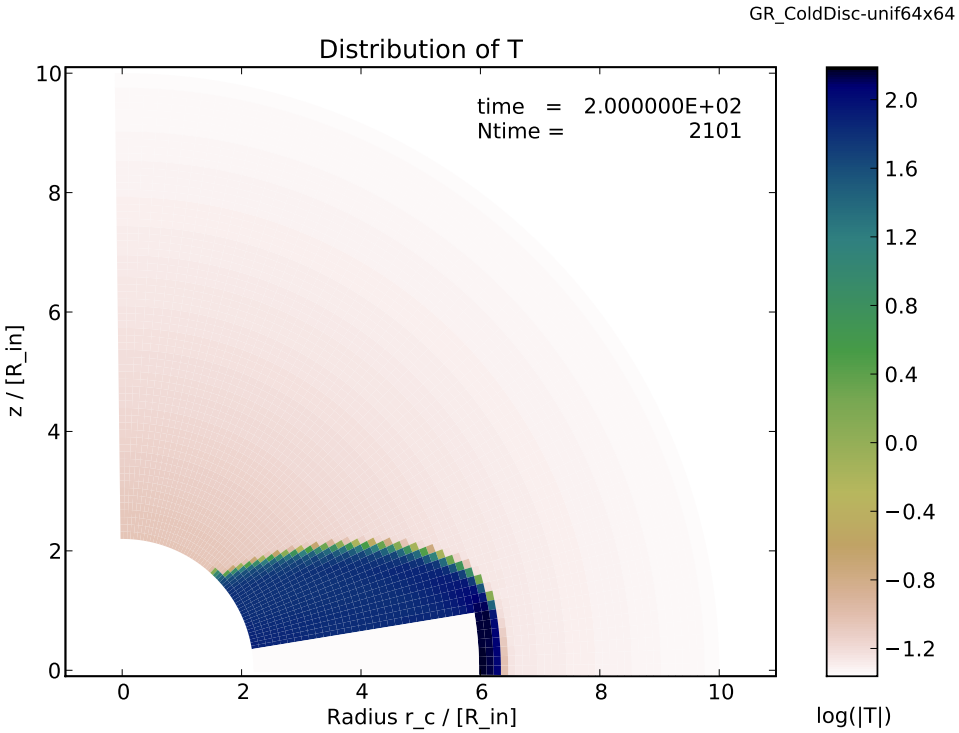


(b) *Latitudinal Velocity $V_{E\,lat}$ in the local Euler frame*

Figure 5.28: *General Relativistic Standing Shock at a Cold Disc around a Schwarzschild Black Hole: Astro-GRIPS simulation with a uniform grid size of 64x64 cells: radial and latitudinal velocity components in the local Euler frame in units of the speed of light.*



(a) *Pressure Pr in* $8.988 \cdot 10^{11} \frac{\text{dyn}}{\text{cm}^2} = 8.988 \cdot 10^{10}$ Pa.



(b) *Temperature T in* 10^9 K.

Figure 5.29: *General Relativistic Standing Shock at a Cold Disc around a Schwarzschild Black Hole: AstroGRIPS simulation with a uniform grid size of 64x64 cells: pressure and temperature in code units.*

6 Summary and Conclusion

In this thesis the importance of general relativistic effects and the necessity for implicit methods especially in the simulation of accretion flows in the vicinity of compact objects and the jet launching process is pointed out.

For that purpose — as a first step — the user-friendly implicit simulation code Astro-GRIPS, the General Relativistic Implicit Parallel Solver, was developed, which solves the three-dimensional axi-symmetric general-relativistic hydrodynamic Euler- or Navier-Stokes equations under the assumption of a fixed background metric of a Schwarzschild or Kerr black hole using time-implicit methods. It is an almost total re-write of an old spaghetti-code like serial Fortran 77 simulation program. By modernisation and optimization it is now a modern, well structured, user-friendly, flexible and extensible simulation program written in Fortran 90/95.

The general relativistic hydrodynamic Euler- and Navier-Stokes equations were derived under the assumption of a fixed background metric of a Schwarzschild or Kerr black hole using Boyer-Lindquist coordinates. According to the finite volume method a 3D axi-symmetric staggered grid discretization was performed using the internal energy formulation. This was done in such a way that it is easy to reduce the system to the usual Newtonian equations. Different equations of state can be used to close the system: ideal gas, polytropic, isothermal and an approximation of the Synge equation of state, a generalization of the ideal gas law to correctly describe the region between the Newtonian ($\gamma = 5/3$) and the ultra-relativistic ($\gamma = 4/3$) flow states. Also a tabulated equation of state could be easily included.

For the solution of the hydrodynamic Euler equations without diffusion and without sophisticated radiative effects for time-dependent compressible flows explicit methods are very well suited. But to simulate a very complex flow with magnetic fields, diffusive and viscous effects (Navier-Stokes equations) and with atomic and chemical reactions with radiative transfer taken into account, one has to use the numerically unconditionally stable implicit methods: In contrast to explicit methods, implicit methods are not numerically limited by the Courant-Friedrichs-Lewy (CFL) time step size, and so simulations with physical phenomena, that posses various different time scales, are possible without stagnation. In the future implicit methods can also help to overcome the so-called "time-step crisis" in gravitational collapse simulations (but therefor one has to implement a Poisson solver in the Newtonian case or solve the Einstein's field equations in the general relativistic case). Another advantage of implicit methods is that there is a tighter coupling of the equations, which is very important for an accurate description of non-linear effects.

The system of large sparse linear equations, which is constructed for the implicit method, is solved with the defect-correction iteration procedure using the Black-White Line-Gauß-Seidel (BW-LGS), the Approximate Factorization Method (AFM) or the Krylov Subspace Iterative Methods (KSIMs) like GMRES, BiCGSTAB and TFQMR. The equations can be solved sequentially (the implicit operator splitting approach) or solved together as one large coupled system. It is also possible to group some equations together and solve them with one method and the other equations with another method.

Here a superior feature of Astro-GRIPS is the large flexibility in choosing the solution coupling and method by only changing one line in the parameter file, the SolMethod parameter. The hierarchical solution scenario (HSS), the gradual enhancement of the equation coupling, can be used together with the restart possibility and the prolongation to a finer grid, to find a stationary or a quasi-stationary state more quickly.

The optimization of the matrix construction by function calls, and separating the pure Jacobian element calculation from the calculation of the matrix elements, where the boundary conditions are additionally be considered, makes the algorithm better maintainable and extendable. Now the matrix elements are directly calculated during the fill-in of the matrix structure of the selected method. This optimization results in a faster matrix construction compared to the before used calculate and copy approach and was a necessary step for the parallelization.

Large amount of time was spent to perform the MPI-parallelization for distributed (and shared) memory machines. Here a domain- and matrix-decomposition with halo cell communication was implemented and the ScaLAPACK band matrix solver is used in the parallel implementation of the BW-LGS and AFM methods. When the program is compiled with PETSc support, then the interface to use the PETSc-library is compiled in and the full range of Krylov Subspace Iterative Methods (KSIMs, KSP) with preconditioners from PETSc can be used, but if one has no PETSc library one can already use BW-LGS and AFM. The efficient scaling of the algorithm was shown. As expected, the KSIM show the best scaling, BW-LGS is not suitable for all problems and AFM cannot be parallelized very efficiently due to its two-step algorithm in the two different directions, where the matrix has to be re-ordered in-between, which results in large communication costs.

Many improvements were applied to the simulation program to try to make the code usage very easy and comfortable:
After setting up some environment variables using a small shell script (this step is usually only done once), one can easily configure the code using a configure script, which has several options e.g. to change the used compiler and library paths, to request a debug compilation or MPI parallel mode and PETSc support. For each problem there is a separate directory with a problem dependent src/Setup.F90 file, which contains the initial conditions (and special boundary conditions when necessary) and one or more parameter files, where one can specify almost all parameters used for a simulation run. This has the advantage that one can change many parameters of a simulation without re-compiling the code.
For serial and parallel data input and output the NetCDF data format, a portable standard binary data format, is used. This format is better than ASCII-data output, which is large, and better than direct binary-data, which is not portable. There are many data manipulation and analysing programs and visualization tools available which can handle NetCDF data. To have an easy start in visualization one can either use netcdf viewers like ncview or the supplied python scripts, which use matplotlib for the visualization.

It was shown that Astro-GRIPS was able to solve special relativistic shock tube problems for Lorentz factors up to at least about 21 although the numerical algorithm solving the internal energy equation with the use of artificial viscosity to replace the lost energy at shock fronts is not so well suited for shock problems than the high resolution shock capturing (HRSC) schemes, which are solving Riemann problems and are therefore especially adapted for shock problems by construction. The general relativistic spherical accretion shows that the general relativistic source terms are implemented right and that the code is capable of finding a general relativistic steady state. The two-dimensional Taylor-Couette problem, where here one looks at a viscous flow between two concentric spheres

instead of the usual cylinders, where one is static and one is rigidly rotating, shows the laminar and turbulent behaviour of such systems and is an application of the Newtonian Navier-Stokes equations. The 2D simulation of the elsewise spherical accretion flow around a static cold disc in the case of a Schwarzschild black hole is shown. There one gets a shock around the disc, like in the forward facing step model in computational fluid dynamics.
These test problems show that the simulation code is working right and that it can now be applied to more complicated problems.

Possible further developments of Astro-GRIPS in the future may include:

- use of Kerr-Schild instead of Boyer-Lindquist coordinates
  to overcome the potential problems of Boyer-Lindquist coordinates near the black hole, especially in the ergosphere.
- use of the total energy formulation (might be better for some problems, but it might be more difficult to construct the Jacobian for the implicit method)
- further test problems, e.g. for dissipative general relativistic flows (Navier-Stokes equations)
- general relativistic magneto-hydrodynamics (MHD)
  with several methods to ensure the $\mathrm{div}\vec{B} = 0$ constraint.
- non-ideal general relativistic MHD: reconnection, ohmic heating, Hall term
- include different cooling and heating processes or approximations thereof, for example relativistic bremsstrahlung, heat conduction or synchrotron emission, which should be easy to include since the internal energy equation is used. These thermodynamical relevant processes are considered to be very relevant e.g. in polar accretion onto neutron stars, X-ray bursts and GRBs. They are mostly not considered up to now, since most of the Astrophysical solvers are only time-explicit and therefore have problems with the different timescales of such problems compared to the hydrodynamical timescale. Here implicit methods are best suited to overcome this time-step problem.
- add Newtonian and general relativistic radiative transfer equations (gray approximation and frequency dependent)
- extend the system of equations to multi-component plasmas:
  In situations where the cooling and heating processes around compact objects like black holes or neutron stars posses timescales that are longer than the corresponding dynamical time scale, electrons and ions of the plasma may decouple thermally and therefore creation and annihilation processes must be taken into account more frequently. Such situations can occur for example in the ergosphere of a Kerr black hole or in boundary layers and at the polar caps of neutron stars. Astro-GRIPS is designed in such a way, that it is relatively easy to add further equations to it. Therefore the implementation of such multi-component equations should be straightforward, if one has calculated the Jacobian entries needed for the implicit solver.
- include atomic and chemical networks
- extend to fully 3D (or 4D in the case of frequency dependent radiative transfer)
- non-fixed spacetimes (coupling with solution of Einstein's field equations or with some approximations thereof)

Astro-GRIPS, the **G**eneral **R**elativistic **I**mplicit **P**arallel **S**olver, the numerical simulation code described here, is a fully implicit solver for flows in general relativistic Astrophysics, that is parallelized, can be used very flexible and runs on various computer platforms. Due to the implicit method the code is unconditionally stable and takes into account the coupling of equations and therefore the non-linear behaviour of the fluid flows.

With the Hierarchical Solution Scenario (HSS) which consists of the gradual coupling of the equations, one has a powerful method for quickly finding quasi-stationary solutions.

The development of such an implicit simulation code is important to advance the numerical simulation techniques in Astrophysical Fluid Dynamics, so that one has the basis to include and study the influence of interesting physical aspects: magnetic fields, radiative processes, heating and cooling and atomic and chemical reactions (all in Newtonian as well as in a general relativistic formulation).

The aim is to understand the nature of the flows around compact objects, especially the formation and acceleration of ultra-relativistic multi-component plasma MHD-jets around spinning black holes and other ultra-compact objects and finally compare the numerical results with Astrophysical observations.

Acknowledgement

Thanks goes to my referees Prof. Dr. Max Camenzind and Priv.-Doz. Christian Fendt and the other examiners Prof. Dr. Matthias Bartelmann and Prof. Dr. Immo Appenzeller for their prompt agreements to be my referees and examiners.

I especially say thanks to my supervisor Prof. Dr. Max Camenzind for the possibility to do this thesis and the continuous support, his excellent lectures, the research seminars and the opportunity to take part of his very broad knowledge about relativity and compact objects.

Special Thanks goes to:

Ahmad A. Hujeirat, the General Relativistic - Implicit -Radiative Magneto-HydroDynamics (GR-I-RMHD) project initiator, who invested a lot of time to explain me implicit methods and his implicit simulation program in great detail and with whom I had lots of interesting discussions.

Paul P. Hilscher doing his excellent Diplomarbeit (diploma thesis) in the GR-I-RMHD group for helping me in the programming, the innovative discussions and the nice time we spent together.

I would like to also thank all the other current and former members of the theory group as well as our guests for all the discussions and the good atmosphere at the Nordinstitut.

Further thanks goes to all other students and members of the Landessternwarte Heidelberg at the Königstuhl as well as to all other people also from the other Astronomical institutes who made my doctoral stay in Heidelberg a great experience.

Thanks to the Klaus-Tschira-Stiftung for the partial financial support in giving me a research position for 10 hours per week as well as for some travel money and the dual quad core compute server for the GR-I-RMHD project (project number: 00.099.2006).

Thanks to the people of the Helics group and for the possibility to perform large parallel runs on the Helics II cluster of the Interdisciplinary Centre for Scientific Computing (IWR) at the University of Heidelberg.

Thanks to the Hochleistungsrechenzentrum Stuttgart (HLRS) for the excellent courses about parallelization and implicit solvers and the nice time I spent there.

I would also thank all other people who helped and supported me during the last years.

Last but not least I especially thank my parents and my brother
for their continuous support and love.

Bibliography

F. Aharonian, A. G. Akhperjanian, A. R. Bazer-Bachi, M. Beilicke, W. Benbow, D. Berge, K. Bernlöhr, C. Boisson, O. Bolz, V. Borrel, I. Braun, A. M. Brown, R. Bühler, I. Büsching, S. Carrigan, P. M. Chadwick, L.-M. Chounet, G. Coignet, R. Cornils, L. Costamante, B. Degrange, H. J. Dickinson, A. Djannati-Ataï, L. O. Drury, G. Dubus, K. Egberts, D. Emmanoulopoulos, P. Espigat, F. Feinstein, E. Ferrero, A. Fiasson, G. Fontaine, S. Funk, S. Funk, M. Füßling, Y. A. Gallant, B. Giebels, J. F. Glicenstein, P. Goret, C. Hadjichristidis, D. Hauser, M. Hauser, G. Heinzelmann, G. Henri, G. Hermann, J. A. Hinton, A. Hoffmann, W. Hofmann, M. Holleran, S. Hoppe, D. Horns, A. Jacholkowska, O. C. de Jager, E. Kendziorra, M. Kerschhaggl, B. Khélifi, N. Komin, A. Konopelko, K. Kosack, G. Lamanna, I. J. Latham, R. Le Gallou, A. Lemière, M. Lemoine-Goumard, J.-P. Lenain, T. Lohse, J. M. Martin, O. Martineau-Huynh, A. Marcowith, C. Masterson, G. Maurin, T. J. L. McComb, E. Moulin, M. de Naurois, D. Nedbal, S. J. Nolan, A. Noutsos, K. J. Orford, J. L. Osborne, M. Ouchrif, M. Panter, G. Pelletier, S. Pita, G. Pühlhofer, M. Punch, S. Ranchon, B. C. Raubenheimer, M. Raue, S. M. Rayner, A. Reimer, J. Ripken, L. Rob, L. Rolland, S. Rosier-Lees, G. Rowell, V. Sahakian, A. Santangelo, L. Saugé, S. Schlenker, R. Schlickeiser, R. Schröder, U. Schwanke, S. Schwarzburg, S. Schwemmer, A. Shalchi, H. Sol, D. Spangler, F. Spanier, R. Steenkamp, C. Stegmann, G. Superina, P. H. Tam, J.-P. Tavernet, R. Terrier, M. Tluczykont, C. van Eldik, G. Vasileiadis, C. Venter, J. P. Vialle, P. Vincent, H. J. Völk, S. J. Wagner, and M. Ward. Fast Variability of Tera–Electron Volt γ Rays from the Radio Galaxy M87. *Science*, 314:1424–1427, December 2006. doi: 10.1126/science.1134408.

M. A. Aloy, J. M. Ibáñez, J. M. Martí, and E. Müller. GENESIS: A High-Resolution Code for Three-dimensional Relativistic Hydrodynamics. ApJS, 122:151–166, May 1999. doi: 10.1086/313214.

E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, third edition, 1999. ISBN 0-89871-447-8 (paperback). URL http://www.netlib.org/lapack/lug/.

P. Anninos and P. C. Fragile. Nonoscillatory Central Difference and Artificial Viscosity Schemes for Relativistic Hydrodynamics. ApJS, 144:243–257, February 2003. doi: 10.1086/344723.

I. Appenzeller, R. Bender, A. Böhm, N. Drory, K. Fricke, R. Häfner, J. Heidt, U. Hopp, K. Jäger, M. Kümmel, D. Mehlert, C. Möllenhoff, A. Moorwood, H. Nicklas, S. Noll, R. Saglia, W. Seifert, S. Seitz, O. Stahl, E. Sutorius, T. Szeifert, S. Wagner, and B. Ziegler. The FORS Deep Field. *The Messenger*, 100:44–48, June 2000.

I. Appenzeller, R. Bender, A. Böhm, S. Frank, K. Fricke, A. Gabasch, J. Heidt, U. Hopp, K. Jäger, D. Mehlert, S. Noll, R. Saglia, S. Seitz, C. Tapken, and B. Ziegler. Exploring Cosmic Evolution with the FORS Deep Field. *The Messenger*, 116:18–24, June 2004.

R. L. Arnowitt, S. Deser, and C. W. Misner. *Canonical analysis of general relativity*, pages 127–+. 1962.

L. Baiotti, I. Hawke, P. J. Montero, and L. Rezzolla. A new three-dimensional general-relativistic hydrodynamics code. *Memorie della Societa Astronomica Italiana Supplement*, 1:210–+, 2003.

S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith. Efficient management of parallelism in object oriented numerical software libraries. In E. Arge, A. M. Bruaset, and H. P. Langtangen, editors, *Modern Software Tools in Scientific Computing*, pages 163–202. Birkhäuser Press, 1997.

S. Balay, K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang. PETSc users manual. Technical Report ANL-95/11 - Revision 3.0.0, Argonne National Laboratory, 2008.

S. Balay, K. Buschelman, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang. PETSc Web page, 2009. http://www.mcs.anl.gov/petsc.

S. A. Balbus and J. F. Hawley. A powerful local shear instability in weakly magnetized disks. I - Linear analysis. II - Nonlinear evolution. ApJ, 376:214–233, July 1991. doi: 10.1086/170270.

S. A. Balbus, J. F. Hawley, and J. M. Stone. Nonlinear Stability, Hydrodynamical Turbulence, and Transport in Disks. ApJ, 467:76–+, August 1996. doi: 10.1086/177585.

L. S. Blackford, J. Choi, A. Cleary, E. D'Azevedo, J. Demmel, I. Dhillon, J. Dongarra, S. Hammarling, G. Henry, A. Petitet, K. Stanley, D. Walker, and R. C. Whaley. *ScaLAPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1997. ISBN 0-89871-397-8 (paperback). URL http://www.netlib.org/scalapack/slug/.

R. D. Blandford and R. L. Znajek. Electromagnetic extraction of energy from Kerr black holes. MNRAS, 179:433–456, May 1977.

P. Bodenheimer and S. E. Woosley. A two-dimensional supernova model with rotation and nuclear burning. ApJ, 269:281–291, June 1983. doi: 10.1086/161040.

P. Bodenheimer, J. E. Tohline, and D. C. Black. Three-dimensional Numerical Calculations of Collapsing Protostellar Clouds. In *Bulletin of the American Astronomical Society*, volume 10 of *Bulletin of the American Astronomical Society*, pages 655–+, September 1978.

H. Bondi. On spherically symmetrical accretion. MNRAS, 112:195–+, 1952.

Robert H. Boyer and Richard W. Lindquist. Maximal analytic extension of the kerr metric. *Journal of Mathematical Physics*, 8(2):265–281, 1967. doi: 10.1063/1.1705193.

G. E. Brown, C.-H. Lee, R. A. M. J. Wijers, H. K. Lee, G. Israelian, and H. A. Bethe. A theory of gamma-ray bursts. *New Astronomy*, 5:191–210, July 2000. doi: 10.1016/S1384-1076(00)00026-9.

M. Camenzind. *Compact Objects in Astrophysics : white dwarfs, neutron stars, and black holes*. Springer, 2007. ISBN 9-783540-257707.

A. R. Choudhuri. *The physics of fluids and plasmas : an introduction for astrophysicists /*. December 1998.

D. A. Clarke. A Consistent Method of Characteristics for Multidimensional Magnetohydrodynamics. ApJ, 457:291–+, January 1996. doi: 10.1086/176730.

P. Colella and P.R. Woodward. The piecewise parabolic method (ppm) for gas-dynamical simulations. *J. Comput. Phys.*, 54:174–201, 1984.

C. Cutler and K. S. Thorne. An Overview of Gravitational-Wave Sources. *ArXiv General Relativity and Quantum Cosmology e-prints*, April 2002.

D. Dale, M. Droettboom, E. Firing, and J. Hunter. *Matplotlib Release 0.99.0*, 2009. URL http://matplotlib.sourceforge.net/.

M. Darbandi, G. E. Schneider, and S. Vakili. Using different preconditioned krylov subspace methods to solve coupled fluid flow equations. *Computational Fluid Dynamics*, 15(1), 2006.

J.-P. De Villiers and J. F. Hawley. A Numerical Method for General Relativistic Magnetohydrodynamics. ApJ, 589:458–480, May 2003. doi: 10.1086/373949.

A. Einstein. Die Feldgleichungen der Gravitation. *Sitzungsberichte der Königlich Preußischen Akademie der Wis senschaften (Berlin), Seite 844-847.*, pages 844–847, 1915.

S. A. E. G. Falle. A numerical scheme for multifluid magnetohydrodynamics. MNRAS, 344:1210–1218, October 2003. doi: 10.1046/j.1365-8711.2003.06908.x.

J. A. Font. Numerical hydrodynamics and magnetohydrodynamics in general relativity. *Living Reviews in Relativity*, 11(7), 2008. URL http://www.livingreviews.org/lrr-2008-7.

Message Passing Interface Forum. *MPI: A Message-Passing Interface Standard Version 1.3*. Knoxville, TN, USA, 2008.

Message Passing Interface Forum. *MPI: A Message-Passing Interface Standard Version 2.2*. Knoxville, TN, USA, 2009.

J. Frank, A. King, and D. J. Raine. *Accretion Power in Astrophysics: Third Edition*. February 2002.

B. Fryxell, K. Olson, P. Ricker, F. X. Timmes, M. Zingale, D. Q. Lamb, P. MacNeice, R. Rosner, J. W. Truran, and H. Tufo. FLASH: An Adaptive Mesh Hydrodynamics Code for Modeling Astrophysical Thermonuclear Flashes. ApJS, 131:273–334, November 2000. doi: 10.1086/317361.

E. Gabriel, G. E. Fagg, G. Bosilca, T. Angskun, J. J. Dongarra, J. M. Squyres, V. Sahay, P. Kambadur, B. Barrett, A. Lumsdaine, R. H. Castain, D. J. Daniel, R. L. Graham, and T. S. Woodall. Open MPI: Goals, concept, and design of a next generation MPI implementation. In *Proceedings, 11th European PVM/MPI Users' Group Meeting*, pages 97–104, Budapest, Hungary, September 2004.

C. F. Gammie, J. C. McKinney, and G. Tóth. HARM: A Numerical Scheme for General Relativistic Magnetohydrodynamics. ApJ, 589:444–457, May 2003. doi: 10.1086/374594.

T. A. Gardiner and J. M. Stone. Multidimensional MHD Algorithms in Athena. In G. P. Zank and N. V. Pogorelov, editors, *Numerical Modeling of Space Plasma Flows*, volume 359 of *Astronomical Society of the Pacific Conference Series*, pages 143–+, December 2006.

R. Geroch. Relativistic theories of dissipative fluids. *Journal of Mathematical Physics*, 36:4226–4241, August 1995. doi: 10.1063/1.530958.

A. M. Ghez, B. L. Klein, M. Morris, and E. E. Becklin. High proper-motion stars in the vicinity of sagittarius a*: Evidence for a supermassive black hole at the center of our galaxy. *The Astrophysical Journal*, 509(2):678–686, 1998. doi: 10.1086/306528.

S.K. Godunov. A finite difference method for the numerical computation and discontinuous solutions of the equations of fluid dynamics. *Mat. Sb.*, 47:271–306, 1959.

E. Gourgoulhon. 3+1 Formalism and Bases of Numerical Relativity. *ArXiv General Relativity and Quantum Cosmology e-prints*, March 2007.

A. Harten, P.D. Lax, and B. van Leer. On upstream differencing and godunov-type schemes for hyperbolic conservation laws. *SIAM Rev.*, 25:35–61, 1983.

J. F. Hawley, L. L. Smarr, and J. R. Wilson. A numerical study of nonspherical black hole accretion. I Equations and test problems. ApJ, 277:296–311, February 1984a. doi: 10.1086/161696.

J. F. Hawley, L. L. Smarr, and J. R. Wilson. A numerical study of nonspherical black hole accretion. II - Finite differencing and code calibration. ApJS, 55:211–246, June 1984b. doi: 10.1086/190953.

P. H. Hilscher. An implicit general relativistic hydrodynamic solver for dissipative flows in high-energy astrophysics. Master's thesis, Ruprecht-Karls Universität Heidelberg, 2009.

A. Hujeirat. Hydrodynamical calculations towards steady state structures in boundary layers in accretion disks. 2: 2-D models. A&A, 295:268–288, March 1995.

A. Hujeirat. A problem-orientable numerical algorithm for modeling multi-dimensional radiative MHD flows in astrophysics — the hierarchical solution scenario. *Computer Physics Communications*, 168:1–24, May 2005a. doi: 10.1016/j.cpc.2005.01.013.

A. Hujeirat. A method for relaxing the Courant-Friedrich-Levy condition in time-explicit schemes. A&A, 430:893–903, February 2005b. doi: 10.1051/0004-6361:20041530.

A. Hujeirat. A method for enhancing the stability and robustness of explicit schemes in CFD. *New Astronomy*, 10:173–193, January 2005c. doi: 10.1016/j.newast.2004.09.003.

A. Hujeirat and R. Rannacher. On the efficiency and robustness of implicit methods in computational astrophysics. *New Astronomy Review*, 45:425–447, April 2001. doi: 10.1016/S1387-6473(00)00142-1.

A. Hujeirat and F.-K. Thielemann. Angular momentum transport during X-ray bursts in neutron stars: a numerical general relativistic hydrodynamical study. A&A, 496:609–617, March 2009. doi: 10.1051/0004-6361/200810793.

A. Hujeirat, B. W. Keil, and F. Heitsch. Advanced numerical methods in astrophysical fluid dynamics. *ArXiv e-prints*, December 2007.

A. Hujeirat, M. Camenzind, and B.W. Keil. An implicit numerical algorithm for solving the general relativistic hydrodynamical equations around accreting compact objects. *New Astronomy*, 13(6):436 – 450, 2008. ISSN 1384-1076. doi: DOI:10.1016/j.newast.2007.12.004.

B. Keil and M. Schartmann. Ein Satellit auf Einsteins Spuren. *Sterne und Weltraum*, pages 14–15, January 2005.

R. P. Kerr. Gravitational field of a spinning mass as an example of algebraically special metrics. *Phys. Rev. Lett.*, 11(5):237–238, Sep 1963. doi: 10.1103/PhysRevLett.11.237.

S. Koide, K. Shibata, and T. Kudoh. Relativistic Jet Formation from Black Hole Magnetized Accretion Disks: Method, Tests, and Applications of a General RelativisticMagnetohydrodynamic Numerical Code. ApJ, 522:727–752, September 1999. doi: 10.1086/307667.

S. S. Komissarov. General relativistic magnetohydrodynamic simulations of monopole magnetospheres of black holes. MNRAS, 350:1431–1436, June 2004. doi: 10.1111/j.1365-2966.2004.07738.x.

F. K. Lamb and M. C. Miller. Sonic-Point and Spin-Resonance Model of the Kilohertz QPO Pairs. *ArXiv Astrophysics e-prints*, August 2003.

R. J. Leveque. Nonlinear Conservation Laws and Finite Volume Methods. In O. Steiner and A. Gautschy, editors, *Saas-Fee Advanced Course 27: Computational Methods for Astrophysical Fluid Flow.*, pages 1–+, 1998.

J. Li, W. Liao, A. Choudhary, R. Ross, R. Thakur, W. Gropp, R. Latham, A. Siegel, B. Gallagher, and M. Zingale. Parallel netcdf: A scientific high-performance i/o interface. In *Proceedings of Supercomputing Conference*, Phoenix, Arizona, November 2003.

M. Liebendörfer, S. Rosswog, and F.-K. Thielemann. An Adaptive Grid, Implicit Code for Spherically Symmetric, General Relativistic Hydrodynamics in Comoving Coordinates. ApJS, 141:229–246, July 2002. doi: 10.1086/339872.

R. Maartens. Causal Thermodynamics in Relativity. *ArXiv Astrophysics e-prints*, September 1996.

R. W. MacCormack and G. V. Candler. The solution of the Navier-Stokes equations using Gauss-Seidel line relaxation. *Computers and Fluids*, 17:135–150, 1989.

J. M. Martí and E. Müller. Numerical hydrodynamics in special relativity. *Living Reviews in Relativity*, 6(7), 2003. URL http://www.livingreviews.org/lrr-2003-7.

J. M. Martí, J. M. Ibáñez, and J. A. Miralles. Numerical relativistic hydrodynamics: Local characteristic approach. Phys. Rev. D, 43:3794–3801, June 1991. doi: 10.1103/PhysRevD.43.3794.

W. G. Mathews. The Hydromagnetic Free Expansion of a Relativistic Gas. ApJ, 165:147–+, April 1971. doi: 10.1086/150883.

M. M. May and R. H. White. Hydrodynamic Calculations of General-Relativistic Collapse. *Physical Review*, 141:1232–1241, January 1966. doi: 10.1103/PhysRev.141.1232.

A. Meister. *Numerik linearer Gleichungssysteme - Eine Einführung in moderne Verfahren*. Vieweg Verlag, Wiesbaden, 2nd revised edition, 2005. ISBN 3-528-13135-7.

F. C. Michel. Accretion of Matter by Condensed Objects. Ap&SS, 15:153–160, January 1972. doi: 10.1007/BF00649949.

A. Mignone and G. Bodo. A version of PPM for multidimensional relativistic hydrodynamics. *New Astronomy Review*, 47:581–583, October 2003. doi: 10.1016/S1387-6473(03)00098-8.

A. Mignone, G. Bodo, S. Massaglia, T. Matsakos, O. Tesileanu, C. Zanni, and A. Ferrari. PLUTO: A Numerical Code for Computational Astrophysics. ApJS, 170:228–242, May 2007. doi: 10.1086/513316.

Charles W. Misner, Kip S. Thorne, and John A. Wheeler. *Gravitation*. W. H. Freeman, San Francisco, 1973. ISBN 0-7167-0344-0 (paper).

Y. Mizuno, K.-I. Nishikawa, S. Koide, P. Hardee, and G. J. Fishman. General Relativistic Magnetohydrodynamic Simulations of Jet Formation with a Thin Keplerian Disk. *ArXiv Astrophysics e-prints*, September 2006. (RAISHIN).

A. Müller. *Black Hole Astrophysics: Magnetohydrodynamics on the Kerr Geometry*. PhD thesis, Ruprecht-Karls Universität Heidelberg, 2004.

M. L. Norman and K.-H. A. Winkler. Why Ultrarelativistic Numerical Hydrodynamics is Difficult. In K.-H. A. Winkler & M. L. Norman, editor, *Astrophysical Radiation Hydrodynamics*, pages 449–+, 1986.

B. W. O'Shea, G. Bryan, J. Bordner, M. L. Norman, T. Abel, R. Harkness, and A. Kritsuk. Introducing Enzo, an AMR Cosmology Application. *ArXiv Astrophysics e-prints*, March 2004.

R. Penrose. Gravitational Collapse: the Role of General Relativity. *Nuovo Cimento Rivista Serie*, 1: 252–+, 1969.

R. Rabenseifner (ed.). Parallel Programming Workshop, Course Material for HLRS Course 2007-C. Interner Bericht 5 (38th revised and extended edition), Hochleistungsrechenzentrum Stuttgart (HLRS), Universität Stuttgart, March 2007.

R. Rew, G. Davis, S. Emmerson, H. Davies, and E. Hartne. *The NetCDF Users Guide Data Model, Programming Interfaces, and Format for Self-Describing, Portable Data NetCDF Version 4.0.1*, 2009. URL http://www.unidata.ucar.edu/software/netcdf/.

G. A. Richardson and T. J. Chung. Computational Relativistic Astrophysics Using the Flow Field-dependent Variation Theory. ApJS, 139:539–563, April 2002. doi: 10.1086/338508.

D. C. Robinson. Uniqueness of the Kerr black hole. *Physical Review Letters*, 34:905–+, April 1975. doi: 10.1103/PhysRevLett.34.905.

P.L. Roe. Approximate riemann solvers, parameter vectors and difference schemes. *J. Comput. Phys.*, 43:357–372, 1981.

D. Ryu, I. Chattopadhyay, and E. Choi. Equation of State in Numerical Relativistic Hydrodynamics. *ApJs*, 166:410–420, 2006. doi: 10.1086/505937.

K. Schwarzschild. Über das Gravitationsfeld eines Massenpunktes nach der Einstein'schen Theorie. page 189, 1916.

N. I. Shakura and R. A. Sunyaev. Black holes in binary systems. Observational appearance. A&A, 24:337–355, 1973.

G. A. Sod. A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *Journal of Computational Physics*, 27:1–31, April 1978.

L. Stella and M. Vietri. Lense-Thirring Precession and Quasi-periodic Oscillations in Low-Mass X-Ray Binaries. ApJ, 492:L59+, January 1998. doi: 10.1086/311075.

J. M. Stone and M. L. Norman. ZEUS-2D: A radiation magnetohydrodynamics code for astrophysical flows in two space dimensions. I - The hydrodynamic algorithms and tests. ApJS, 80:753–790, June 1992. doi: 10.1086/191680.

F. D. Swesty. Implicit general relativistic hydrodynamic methods for modeling the late-time explosion mechanism in core collapse supernovae. ApJ, 445:811–827, June 1995. doi: 10.1086/175742.

J.L. Synge. *The Relativistic Gas*. North-Holland; Interscience, Amsterdam, Netherlands; New York, U.S.A., 1957.

A. H. Taub. Relativistic Rankine-Hugoniot Equations. *Physical Review*, 74:328–334, August 1948. doi: 10.1103/PhysRev.74.328.

E. F. Taylor and J. A. Wheeler. *Spacetime physics - Introduction to special relativity (2nd edition)*. 1992.

G. I. Taylor. Stability of a Viscous Liquid Contained between Two Rotating Cylinders. *Royal Society of London Philosophical Transactions Series A*, 223:289–343, 1923.

E. F. Toro, M. Spruce, and W. Speares. Restoration of the contact surface in the HLL-Riemann solver. *Shock Waves*, 4:25–34, July 1994.

G. Toth, R. Keppens, and M. A. Botchev. Implicit and semi-implicit schemes in the Versatile Advection Code: numerical tests. A&A, 332:1159–1170, April 1998.

C. M. Urry and P. Padovani. Unified Schemes for Radio-Loud Active Galactic Nuclei. PASP, 107: 803–+, September 1995. doi: 10.1086/133630.

B. van Leer. Towards the ultimate conservative difference scheme. V - A second-order sequel to Godunov's method. *Journal of Computational Physics*, 32:101–136, July 1979. doi: 10.1016/0021-9991(79)90145-1.

G. V. Vaughan, B. Elliston, T. Tromey, and I. L. Taylor. *GNU Autoconf, Automake and Libtool*. New Riders/Sams, 2000. URL http://sources.redhat.com/autobook.

R. F. Warming and Richard M. Beam. An extension of a-stability to alternating direction implicit methods. *BIT Numerical Mathematics*, 19:395–417, September 1979. doi: 10.1007/BF01930993.

M. C. Weisskopf, M. Karovska, G. G. Pavlov, V. E. Zavlin, and T. Clarke. Chandra observations of neutron stars: an overview. Ap&SS, 308:151–160, April 2007. doi: 10.1007/s10509-007-9322-1.

J. R. Wilson. Numerical Study of Fluid Flow in a Kerr Space. ApJ, 173:431–+, April 1972. doi: 10.1086/151434.

K.-H. A. Winkler, M. L. Norman, and M. J. Newman. Adaptive mesh techniques for fronts in star formation. *Physica D Nonlinear Phenomena*, 12:408–425, July 1984. doi: 10.1016/0167-2789(84)90545-1.

G. Wuchterl. Hydrodynamics of giant planet formation. I - Overviewing the kappa-mechanism. A&A, 238:83–94, November 1990.

W. Zhang and A. I. MacFadyen. RAM: A Relativistic Adaptive Mesh Refinement Hydrodynamics Code. ApJS, 164:255–279, May 2006. doi: 10.1086/500792.

U. Ziegler. NIRVANA + : An adaptive mesh refinement code for gas dynamics and MHD. *Computer Physics Communications*, 109:111–134, April 1998. doi: 10.1016/S0010-4655(98)00022-8.

Index