ICTS Summer School on Gravitational-Wave Astronomy

Core-Collapse Supernovae

You will find that some of these problems don't contain all the information that you need to solve them. That's intentional: Filling in the dots by using the literature or an informed guess based on something else that we've covered in the lecture is part of the exercises. The point is not so much to get the correct equations and numerical results, but to discuss and justify the assumptions that you make.

1. Order-of-magnitude estimates for stellar structure

(a) Derive an order-of-magnitude estimate for the structure of stars on the zero-age main sequence by solving the equation of hydrostatic equilibrium,

$$\frac{\partial P}{\partial r} = -\frac{G m(r) \rho(r)}{r^2},$$

the equation for the enclosed mass $m(r)$,

$$m(r) = \int 4\pi \rho r'^2 \, \mathrm{d}r',$$

and the equation of state (EoS) for the pressure P for a one-zone stellar model. To this end, you can approximate $\partial P/\partial r \sim P_\mathrm{c}/R$ in terms of the central pressure P_c and stellar radius R, and use similar one-zone approximations in other places. Show that both for a radiation-dominated EoS and for a perfect EoS, this implies

$$\frac{T^3}{\rho} \propto M^{1/2}.$$

Assuming that the specific entropy is dominated by the radiation component of the stellar plasma, conclude that more massive stars start their lives with higher specific entropy. Use your findings to interpret the evolutionary tracks of central density vs. temperature discussed in the lectures.

(b) The shell structure during the advanced neutrino-cooled burning stages of massive stars can be understood as a result of *balanced power*. To good approximation, energy release $\dot{\epsilon}_\mathrm{nuc}$ by nuclear burning and neutrino cooling $\dot{\epsilon}_\nu$ balance each other in the (convective) C, Ne, O, or Si burning shells. Using a one-zone approximation, derive a relationship between the density, temperature, and fuel mass fraction X_fuel assuming a two-particle reaction with $\dot{\epsilon}_\mathrm{nuc} \propto \mathcal{R}_0 \rho X_\mathrm{fuel}^2 T^\alpha$ and pair annihilation as the primary neutrino energy loss channel. Then show that the condition of balanced power determines the shell entropy if the burning temperature is known. Why can we assume that burning temperature for the various burning stages is almost fixed?

(c) Estimate the convective velocity and Mach number in selected convective burning shells (C burning, O burning, Si burning) using mixing-length theory (MLT). According to MLT, the typical convective energy flux,

$$F_{\mathrm{conv}} \sim 4\pi r^2 \rho v^3,$$

is equal to the total nuclear energy generation rate $\dot{Q}_{\mathrm{nuc}}$ in a convective shell. You can estimate $\dot{Q}_{\mathrm{nuc}}$ using reasonable values for the shell mass (e.g., from stellar models at `https://2sn.org/stellarevolution/`), rough numbers for the burning time scale from the lecture, and the typical energy release Δq per nucleon during a given burning stage. Relate the shell temperature to the shell radius r using $kT \sim GM/(3r)$. Any other structural quantities that you may need can be taken from the aforementioned stellar models. For the duration of burning stages, find some Kippenahn diagrams like `https://2sn.org/stellarevolution/explain.gif`.

2. MHD-Driven Hypernovae – Requirements

Most scenarios for obtaining explosion with energies $\gg 10^{51}$ erg rely on tapping the rotational energy of the supernova core to create strong magnetic fields that ultimately power the explosion (e.g. by creating jets). This requires rapidly rotating progenitor cores, especially if the neutron star is to survive in the explosion as in the millisecond magnetar model:

(a) Estimate the required spin rate of a neutron star for reaching a rotational energy $E_{\mathrm{rot}} = I\omega^2/2 = 10^{52}$ erg (where I is the moment of inertia and ω is the angular velocity).

(b) Using conservation of angular momentum ($L = I\omega$), infer the required rotation rate in the progenitor. You can assume that the matter that makes up the neutron star is initially located within a radius of ~1000 km.

3. Neutrino Trapping

The dominant scattering process for neutrinos during iron core collapse is neutrino-nucleus scattering $\nu + A \rightarrow \nu + A$. The scattering opacity depends on the number density n_A of nuclei, their mass and charge number (A and Z), and the neutrino energy ϵ,

$$\kappa_{\mathrm{s}} \approx \frac{\sigma_0}{32}\left(\frac{\epsilon}{m_e c^2}\right)^2 A^2 n_A \left[C_A - C_V + (2 - C_A - C_V)\left(\frac{2Z - A}{A}\right)\right]^2,$$

where the Fermi constant for weak interactions is hidden in $\sigma_0 = 1.761 \times 10^{-44}\ \mathrm{cm}^2$, and $C_V = 0.96$ and $C_A = 1/2$ are vector and axial-vector coupling constants. We shall use this scattering opacity to estimate when neutrino trapping occurs during collapse:

(a) What are reasonable values for Z and A? Assuming that the average energy of escaping neutrino is 5 MeV, express κ_{s} as a function of ρ.

(b) Neutrinos trapping occurs roughly when the mean free path ($\lambda = 1/\kappa_{\mathrm{s}}$) equals the radius R of the collapsing core. Estimate R and then solve $\kappa_{\mathrm{s}} R = 1$ for the trapping density.

4. Numerical Treatment of Neutrino Transport

(a) Stiff Source Terms in the Collision Integral

The contributions of absorption/emission reactions like $n + \nu_e \leftrightharpoons p + e^-$ in the collision integral can be written in the following form,

$$\left(\frac{\partial f}{\partial t}\right)_{n\ \nu_e \leftrightharpoons p\ e^-} = -\kappa c(f - f_{\mathrm{eq}}).$$

From the previous problem, you know that the equilibration time-scale $1/(\kappa c)$ can be extremely short. This causes problems if we want to take reasonable time steps in a simulation ($\Delta t = 10^{-7} \dots 10^{-6}$ s).

Consider the simple differential equation,

$$\frac{\mathrm{d}u}{\mathrm{d}t} = -\lambda(u - \sin t).$$

In the PYTHON programme `stiff_ode_explicit.py`, we try to solve this with a simple forward Euler method, where we advance to the next time step according to

$$u_{i+1} = u_i - \delta t(u_i - \sin t_i).$$

As initial condition, we choose $u(0) = 1$. Run the programme by executing:

`python3.5 stiff_ode_explicit.py`

What happens at very small t? Next, edit the programme to set the time step to $\Delta t = 0.018$, then to $\Delta t = 0.02$ and to $\Delta t = 0.025$. Is this still an acceptable solution?

Apparently, we face a severe time step constraint. Going to higher-order integration methods (Runge-Kutta) does not cure this limitation. To avoid the problem, we can compute the right-hand side at the next time step instead:

$$u_{i+1} = u_i - \delta t(u_{i+1} - \sin t_{i+1}).$$

This is the backward Euler method. For this simple example, you can easily modify the algorithm to implement it. Verify that the solution remains stable for large time steps.

The backward Euler method is stable but not very accurate. One could try to achieve higher-order accuracy by solving

$$u_{i+1} = u_i - \delta t\left(\frac{u_{i+1} + u_i}{2} - \sin t_{i+1/2}\right).$$

instead (implicit midpoint method). Try whether you can implement this method and find out whether it is stable for arbitrary Δt. If so does it also give a “nice solution”?

Note: Stiff source terms also occur in many other contexts (e.g. nuclear reaction networks, cooling in radiative shock). For some of these applications, implicit higher-order methods (Kaps-Rentrop, Bader-Deuflhard) are the method of choice, which we do not treat here.

(b) The Diffusion Equation

Verify that the diffusion approximation we discussed it in the lectures boils down to a single evolution equation for J:

$$\frac{\partial J}{\partial t} - \frac{1}{3}\nabla \cdot \left(\frac{1}{\kappa_s + \kappa_a}\nabla J\right) = \kappa_a (J_{\mathrm{eq}} - J).$$

This is very similar to the prototypical 1D equation,

$$\frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} = 0.$$

For this equation, a simple finite-difference scheme to advance a solution on from time step n to time step $n+1$ might be:

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} - \frac{1}{\Delta x}\left(\frac{u_{i+1}^n - u_i^n}{\Delta x} - \frac{u_i^n - u_{i-1}^n}{\Delta x}\right) = 0.$$

This is implemented in the Python programme `diffusion_explicit.py`. If you run the programme, you will find the numerical solution (green) for the diffusion of a Gaussian peak (initial conditions in blue) compared to the analytic one (red dashed curve). Right now the time step is set to $\Delta t = \Delta x^2/2$. Check what happens for $\Delta t = \Delta x^2$ by editing the programme and re-running it.

Again, the way to avoid the severe time step constraint is to use an implicit method. In `diffusion_bw_euler.py`, I have implemented the backward Euler method,

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} - \frac{1}{\Delta x}\left(\frac{u_{i+1}^{n+1} - u_i^{n+1}}{\Delta x} - \frac{u_i^{n+1} - u_{i-1}^{n+1}}{\Delta x}\right) = 0.$$

Since this method couples the new solution values at all grid points with each other, we need to solve a linear system with a tridiagonal coefficient matrix, which is done using the SciPy package – have look at line 62 if you want to see how this is done.

Run the code with various time steps far greater than the explicit stability limit (e.g. $\Delta t = (100\ldots200)\Delta x^2$) and verify that the method remains stable. Also run with $\Delta t = \Delta x^2/2$. Can you see a difference in the numerical solution? If you are brave, you can try to adjust the code to get second-order convergence in time (how?). Does this make the numerical solution more accurate for large Δt?

(c) Two-Moment Approximation Let us now solve a two-moment system in 1D assuming a constant scattering opacity κ and no source term in the equation for J:

$$\begin{aligned}\frac{\partial J}{\partial t} + \frac{\partial H}{\partial x} &= 0\\ \frac{\partial H}{\partial t} + \frac{1}{3}\frac{\partial K}{\partial x} &= -\kappa H.\end{aligned}$$

The programme `two_moment_eddington_approximation_hlle.py` solves this equation by first advancing the hyperbolic part of the system (without source terms) for one time-step and then updates H using an implicit method. Run with $\kappa = 100$ and convince yourself that the numerical solution remains close to the analytic solution of a diffusion equation with the same κ (Why?).

Now set $\kappa = 10$. Do you see changes?

Next, set $\kappa = 10^5$ and re-run. Nothing happens because diffusion is now extremely slow. Then change the width of the Gaussian (variable `width`); set it to 0.01 instead of 0.1. What happens now? Is this physical, and if not what is the reason for this phenomenon?

If you have some background in computational astrophysics, you can try to fix this: For the solution of the hyperbolic part, we use the Kurganov-Tadmor central scheme with signal velocity $c_s = 1/\sqrt{3}$. See what happens if you switch of the diffusive term in the flux formula if the optical depth per zone $\tau = \kappa\delta x$ is larger than unity, or suppress this term by multiplying it with $\max(1-\tau, 0)$.

5. Neutrino Mean Free Path and Equilibration Time

Neglecting a few blocking factors and assuming neutrino energies ϵ much larger than the proton-neutron mass difference, the cross section for absorption of electron neutrinos by neutrons (or electron antineutrinos by protons) is roughly,

$$\sigma \simeq \frac{\sigma_0 \epsilon^2}{4 m_e^2 c^4}(g_V^2 + 3g_A^2),$$

where $g_V = 1$ and $g_A = 1.254$.

(a) From σ, we obtain an opacity for neutrino absorption as $\kappa = \sigma n_n$, where n_n is the neutron number density (Check its dimensions). Estimate the mean free path $1/\kappa$ and the equilibration time-scale $t_{\mathrm{eq}} = 1/(\kappa c)$ for a neutrino energy of $\epsilon = 100$ MeV and a density of $4 \times 10^{14}\ \mathrm{g\ cm^{-1}}$ (How can you estimate n_n from this?).

(b) The primary detection channel for supernova neutrinos in water Cherenkov detectors is $p + \bar{\nu}_e \rightarrow n + e^+$, where the hydrogen nuclei in the water molecules are the proton targets. Using the water mass of 50 kT in Super-Kamiokande, the neutron star binding energy radiated in neutrinos in a supernova ($\sim 3 \times 10^{53}$ erg), and an average electron antineutrino energy of 15 MeV, calculate the number of detection events in Super-K expected from a supernova in the Large Magellanic Cloud at a distance of 50 kpc (1 kpc $= 3.086 \times 10^{21}$ cm). Assume that roughly 1/6 of the neutrinos come out as $\bar{\nu}_e$.

6. Gravitational Waves from Supernovae

(a) Starting from the Einstein quadrupole formula for the gravitational wave strain at a distance D from the source,

$$h^{ij} = \frac{2G}{c^4 D}\frac{\partial^2}{\partial t^2}\int \rho(x^i x^j - \delta^{ij} r^2)\,\mathrm{d}^3 x,$$

show that h^{ij} can also be expressed without a time derivative by the stress formula

$$h^{ij} = \frac{2G}{c^4 D}\mathrm{STF}\left(\int \rho(g^i x^j + v^i v^j)\,\mathrm{d}^3 x\right).$$

Here, STF denotes the symmetric trace-free part of a three-dimensional tensor,

$$\mathrm{STF}(X^{ij}) = \frac{1}{2}(X^{ij} + X^{ji}) - \frac{1}{3}\delta^{ij}\mathrm{Tr}\,X.$$

Use the continuity equation and the Euler equation of hydrodynamics in combination with integration by parts to eliminate the time derivatives. Discuss possible advantages/disadvantages of the stress formula and the original quadrupole formula for computing gravitational wave amplitudes from hydrodynamical simulation data.

(b) Based on the stress formula, argue that the gravitational wave amplitude $h^{ij}D$ from a given process (i.e., convection, neutron star oscillation) is set by the kinetic/potential energy contained in the fluid motions, i.e.,

$$h^{ij}D \sim \alpha \frac{E_{\mathrm{conv}}}{Gc^4},$$

or,

$$h^{ij}D \sim \alpha \frac{E_{\mathrm{osci}}}{Gc^4},$$

where $\alpha < 1$ is an efficiency factors. Estimate the kinetic energy E_{conv} of neutrino-driven convection in the gain region from a typical heating rate $\dot{Q}_\nu \sim 5 \times 10^{52}\,\mathrm{erg\,s^{-1}}$ and a dwell time of $\sim$ 10 ms in the gain region, and use this to calculate the typical strain h from a supernova at a distance of 10 kpc.

(c) You can download raw time series for h^{ij} from real simulation data from `SOMEURL` along with a simple Jupyter notebook to analyse and plot the waveforms. The notebook allows you to select an observer direction by specifying the latitude and longitude of the observer direction relative to the spherical polar coordinate system of the supernova simulation, calculates the symmetric transverse traceless part of h^{ij} (which is not yet done by simulation code), and decomposes it into the two polarisation modes h_+ and $h_\times$.

Plot h_+ and $h_\times$ in the time domain and in the time-frequency using a wavelet transform for different models. Identify signal components that were discussed in the lectures, and discuss the dynamics of the underlying supernova models.

The notebook allows you to specify the sampling rate of the time series before you compute the wavelet transform. Experiment with different sampling rates discuss how the spectrum changes when you set the sampling rate to 2000 Hz or 1000 Hz.

The wavelet transform uses the Morlet wavelet with a specifiable shape parameter that controls the number of oscillations in the mother wavelet. Vary the shape parameter and discuss the optimal choice for supernova waveforms. How is this choice related to the physics of the oscillation modes that produce the waves?

For some models, you will notice strong edge effects in the spectrograms. How do these come about and how can you minimise edge effects?

7. Explosive Burning with Nuclear Reaction Networks

In this problem, we will work with a full-scale reaction network to further illustrate the discussion of advanced burning stages in the lectures. You will need to download the Torch network from the website `http://cococubed.asu.edu/code_pages/net_torch.shtml` (maintained by F. Timmes from Arizona State University). Unpack the archive, and compile:

```
gfortran -o net.x public_torch.f90
```

Then execute `net.x` to perform the reaction network calculations below. Use the default options unless instructed to do otherwise. When asked for the bunring mode, choose option 3 ("adiabatic expansion"). This will calculate the burning in matter that expands adiabatically from a certain peak temperature and density with $\rho = \rho_0 \exp(-t/\tau)$, where $\tau = 446\,\mathrm{s}\,\sqrt{\rho_0/(\mathrm{g\,cm^{-3}})}$. τ is the free-fall time-scale for matter with density ρ_0. This gives a rough approximation for the expansion of material during explosive burning in a supernova. When prompted for the ending time, always choose 1000 s. When asked for the initial composition, choose option 3 and specify it manually.

(a) Start with an initial condition of pure carbon, and a peak temperature and density of 2 GK and $10^8\,\mathrm{g\,cm^{-3}}$. Plot the final elemental composition as a function of Z, and interpret the outcome. Repeat the same exercise after changing the peak temperature to 3.5 GK 4 GK, 4.5 GK, and 6 GK.

(b) Now we shall consider freeze-out from NSE with an initial temperature of 10 GK and $Y_e = 0.5$. Start with a composition of neutrons and protons only. Use a larger network for this problem (network 7, 640 isotopes). Calculate the freeze-out nucleosynthesis for $\rho_0 = 10^7\,\mathrm{g\,cm^{-3}}$, $\rho_0 = 4 \times 10^7\,\mathrm{g\,cm^{-3}}$, $\rho_0 = 10^9\,\mathrm{g\,cm^{-3}}$. What nuclei are predominantly produced? Plot the *isotopic* abundances of Cr, Mn, Fe, Co, Ni, Cu, Zn after β-decays using an appropriate scale (one plot for each initial density, with separate curves for each elements). Interpret the outcome.

(c) Now try the case with $\rho_0 = 4 \times 10^7\,\mathrm{g\,cm^{-3}}$ again, but consider the cases of $Y_e = 0.55$, $Y_e = 0.46$, and $Y_e = 0.42$. What are the dominant heavy nuclei in each case? Make an appropriate plot to demonstrate that Y_e roughly determines the ratio Z/A of the dominant nucleus for neutron-rich matter? Is this also true for proton-rich matter?

(d) If you want, you can modify the code to use a different expansion time-scale. How are the results for NSE freeze-out with $\rho_0 = 4 \times 10^7\,\mathrm{g\,cm^{-3}}$ and $Y_e = 0.5$ affected if you increase the expansion time-scale τ by a factor of 10? What is the effect of a decrease by a factor of 10?

Note: You will need to figure out yourself what information the output files contain – that is part of the assignment as a preparation for the "real world" of sparsely documented astrophysics codes. For reading the output files, the numpy routines `loadtxt()` and `genformtxt()` are useful. For example, to load the mass fractions after β-decay together with Z and A, you can use:

```
d=numpy.genfromtxt('foo_decayed.dat',usecols=(0,1,2))
```

8. Light Curves and Spectra – Simple Estimates

Let us consider Type IIP supernovae from red supergiants, and assume a typical explosion energy $E \sim 10^{51}$ erg and ejecta mass $M \sim 12\,M_\odot$.

(a) We expect that the width of the line features will roughly reflect the ejecta velocity (though the details of line formation are somewhat complicated). Estimate the typical ejecta velocity and find some spectra of Type IIP supernovae to check whether your estimate is reasonable.

(b) During the plateau phase of Type IIP supernovae, the photon luminosity feeds mostly on the thermal energy from shock heating. Let us form a crude estimate of the plateau luminosity – note that there are more much more sophisticated ways to do this.

First, we need the internal energy E_{therm} of the ejecta at shock breakout. How do you expect this to be related to the explosion energy?

(c) Calculate the typical (average) temperature T_0 of the shock H envelope at shock breakout assuming that the ejecta are in the radiation-dominated regime (i.e. the internal energy per unit volume is $u = aT^4$). You'll have to make simplifying assumptions about the temperature distribution of the ejecta. Effectively, you will get a value for a shell roughly in the middle of the H envelope.

(d) You should find that T_0 is considerably higher than the recombination temperature T_{rec}. Before we form a recombination front that propagates down through the hydrogen envelope, the temperature needs to drop considerably. Let us assume that this happens by adiabatic cooling (though radiative cooling by diffusion also plays an important role in practice). How much does the photosphere need to expand until roughly half of the envelope has dropped below the recombination temperature? You can assume that the ejecta expand self-similarly.

(e) Using your estimate for the photospheric radius R_{phot} during the plateau, estimate the luminosity using the Stefan-Boltzmann law. Note that you will not recover the exact scaling law from the lecture, because this would require a more sophisticated derivation including effects of radiative diffusion.

Check whether the predicted value roughly agrees with observed Type-IIP plateau luminosities.