1 6/19/2021

The code you have been writing simulates a neutron star. A neutron star, contrary to its name, is primarily made up of neutrons, protons, electrons, and particles called muons. The physics of all of those particles is contained within the equation of state. What I want to focus on right now is that all of the particles are *fermions* and, hence, neutron stars are stars made of fermions.

I am interested in studying mixed stars, where you have the ordinary matter of a neutron star mixed with dark matter. We do not know what dark matter is, so it is an open question of how to model it. The most commonly studied theory is that dark matter is an as-yet-undiscovered particle. If it is a particle, it can either be a fermion or a boson. I would like you to study the possibility that it is a boson (Troy will study the possibility that it is a fermion). You are therefore going to study a star made up of a mixture of fermions and bosons. These are sometimes called fermion-boson stars.

The fermion sector is the same as you have been working on. It is described by conservative and primitive variables:

$$\Phi, \qquad \Pi, \qquad \Leftrightarrow \qquad P, \qquad \rho, \qquad v. \tag{1}$$

The boson sector is made up of a complex scalar field. A complex scalar field is itself made up of two real scalar fields: ϕ_1 and ϕ_2. You have previously written code for simulating a single real scalar field. You might recall that you needed two additional fields (which we previously labeled as Φ and Π). We again need two additional fields for each scalar field, making a total of four additional fields. This time I will label them as X and Y. The boson sector, then, has a total of six fields:

$$\phi_1, \quad X_1, \quad Y_1, \qquad \text{and} \qquad \phi_2, \quad X_2, \quad Y_2. \tag{2}$$

The definition of the X and Y fields are

$$X_1 = \phi_1', \quad Y_1 = \frac{a}{\alpha}\dot{\phi}_1 \qquad \text{and} \qquad X_2 = \phi_2', \quad Y_2 = \frac{a}{\alpha}\dot{\phi}_2. \tag{3}$$

Your code, then, must have a total of five variables for the fermion sector and six variables for the boson sector. There are then the two gravity variables, a and α.

I need to tell you the equations describing all of these fields. I am going to write them in terms of components of the energy-momentum tensor. The total energy-momentum tensor is the sum of the contributions from the fermion sector and the boson sector. We write this as

$$(T_{\text{tot}})^{\mu}{}_{\nu} = (T_f)^{\mu}{}_{\nu} + (T_b)^{\mu}{}_{\nu}. \tag{4}$$

The components of the energy-momentum tensor for the fermion sector that we will need are

$$\begin{aligned} (T_f)^t{}_t &= -\frac{1}{2}(\Pi + \Phi) \\ (T_f)^t{}_r &= \frac{a}{2\alpha}(\Pi - \Phi) \\ (T_f)^r{}_r &= \frac{1}{2}(\Pi - \Phi)v + P. \end{aligned} \tag{5}$$

We refer to these as the t, t component, the t, r component, and the r, r component. The components of the energy-momentum tensor for the boson sector that we will need are

$$\begin{aligned} (T_b)^t{}_t &= -\frac{1}{a^2}\left(X_1^2 + X_2^2 + Y_1^2 + Y_2^2\right) - V \\ (T_b)^r{}_r &= +\frac{1}{a^2}\left(X_1^2 + X_2^2 + Y_1^2 + Y_2^2\right) - V \\ (T_b)^t{}_r &= -\frac{2}{\alpha a}\left(Y_1 X_1 + Y_2 X_2\right), \end{aligned} \tag{6}$$

where V is called the *scalar potential* and is given by

$$V = \mu^2(\phi_1^2 + \phi_2^2) + \lambda(\phi_1^2 + \phi_2^2)^2. \tag{7}$$

μ is the mass of the scalar field and λ gives the strength at which the scalar field can interact with itself (it is called the *self coupling constant*). For now, take $\mu = 1$ and $\lambda = 0$.

For the fermion sector, the equations to solve are

$$\partial_t \mathbf{u} = -\frac{1}{r^2}\partial_r\left(r^2\frac{\alpha}{a}\mathbf{f}^{(1)}\right) - \partial_r\left(\frac{\alpha}{a}\mathbf{f}^{(2)}\right) + \mathbf{s}', \tag{8}$$

where

$$\mathbf{u} = \begin{pmatrix}\Pi\\ \Phi\end{pmatrix} \qquad \mathbf{f}^{(1)} = \begin{pmatrix}\frac{1}{2}(\Pi-\Phi)(1+v)\\ \frac{1}{2}(\Pi-\Phi)(1-v)\end{pmatrix} \qquad \mathbf{f}^{(2)} = \begin{pmatrix}+P\\ -P\end{pmatrix}, \qquad \mathbf{s}' = \begin{pmatrix}\Omega+\Theta\\ \Omega-\Theta\end{pmatrix} \tag{9}$$

and

$$\begin{aligned}
\Omega &= -4\pi G r\alpha^2\left\{(T_f)^t{}_r\left[(T_{\text{tot}})^r{}_r - (T_{\text{tot}})^t{}_t\right] - (T_{\text{tot}})^t{}_r[(T_f)^r{}_r - (T_f)^t{}_t]\right\}\\
\Theta &= 4\pi G r a\alpha\left[2\frac{\alpha^2}{a^2}(T_{\text{tot}})^t{}_r(T_f)^t{}_r + (T_{\text{tot}})^r{}_r(T_f)^t{}_t + (T_{\text{tot}})^t{}_t(T_f)^r{}_r\right]\\
&\quad + \frac{\alpha}{a}\frac{a^2-1}{2r}\left\{\frac{1}{2}\left[(\Pi-\Phi)v - (\Pi+\Phi)\right] + P\right\}.
\end{aligned} \tag{10}$$

Everything in (8) is the same as before, except the source term $\mathbf{s}'$. Equation (8) is treated just as it was before, and so must be put in the same form as you used previously.

For the boson sector, we need equations for ϕ_1, X_1, Y_1, ϕ_2, X_2, Y_2. They are

$$\begin{aligned}
\dot{\phi}_1 &= \frac{\alpha}{a}Y_1\\
\dot{X}_1 &= \partial_r\left(\frac{\alpha}{a}Y_1\right)\\
\dot{Y}_1 &= \frac{1}{r^2}\partial_r\left(\frac{r^2\alpha}{a}X_1\right) - \alpha a[\mu^2 + 2\lambda(\phi_1^2+\phi_2^2)]\phi_1
\end{aligned} \tag{11}$$

and

$$\begin{aligned}
\dot{\phi}_2 &= \frac{\alpha}{a}Y_2\\
\dot{X}_2 &= \partial_r\left(\frac{\alpha}{a}Y_2\right)\\
\dot{Y}_2 &= \frac{1}{r^2}\partial_r\left(\frac{r^2\alpha}{a}X_2\right) - \alpha a[\mu^2 + 2\lambda(\phi_1^2+\phi_2^2)]\phi_2.
\end{aligned} \tag{12}$$

For the boson sector, the inner boundary conditions are that ϕ_1, ϕ_2 and Y_1, Y_2 are even functions and X_1, X_2 are odd functions. The outer boundary conditions are

$$\begin{aligned}
\dot{\phi}_1 &= -\frac{\phi_1}{r} - X_1, \qquad \dot{Y}_1 = -\frac{Y_1}{r} - Y_1', \qquad X_1 = -\frac{\phi_1}{r} - Y_1,\\
\dot{\phi}_2 &= -\frac{\phi_2}{r} - X_2, \qquad \dot{Y}_2 = -\frac{Y_2}{r} - Y_2', \qquad X_2 = -\frac{\phi_2}{r} - Y_2.
\end{aligned} \tag{13}$$

Finally, there are the equations for the gravity variables. They are

$$\begin{aligned}\frac{\partial_r \alpha}{\alpha} &= +4\pi G r a^2 {(T_{\text{tot}})^r}_r + \frac{a^2-1}{2r} \\ \frac{\partial_r a}{a} &= -4\pi G r a^2 {(T_{\text{tot}})^t}_t - \frac{a^2-1}{2r} \\ \frac{\partial_t a}{a} &= -4\pi G r \alpha^2 {(T_{\text{tot}})^t}_r.\end{aligned} \tag{14}$$

These are exactly the same as you've been using, except now they are written in terms of the total energy-momentum tensor (and not just the energy-momentum tensor for the fermion sector). Note that you only use the $\partial_r a$ equation for the initial data (the $n = 0$ row). You evolve a using the $\partial_t a$ equation. And you must rewrite the $\partial_r \alpha$ equation using RK2.

2 7/2/21

In bosonic systems, we cannot evolve time with a second-order method, because it is not expected to be stable. This means that we cannot use the modified-Euler method. We have to instead move to a third-order method. We will use third-order Runge-Kutta (RK3). For differential equation

$$y' = f(y), \tag{15}$$

the RK3 algorithm we will use is

$$\begin{aligned}k_1 &= hf(y) \\ k_2 &= hf(y + k_1) \\ k_3 &= hf[y + (k_1 + k_2)/4] \\ y(t+h) &= y(t) + \frac{1}{6}(k_1 + k_2 + 4k_3).\end{aligned} \tag{16}$$

3 7/8/21

You have been working on code that can evolve a star with both a fermion and a boson sector. The fermion sector describes the ordinary nuclear matter in a neutron star. The boson sector describes dark matter, which we are modeling as a complex scalar field. In quantum theories, for a particle to have charge (like electric charge), the field must be complex. Therefore, a natural extension of what you are doing, is to consider a *charged* complex scalar field. In quantum theories, it is the photon that mediates the electric force. Since dark matter does not interact with photons (otherwise we would be able to see it and it would no longer be dark), we are modeling dark matter as a complex scalar field that interacts with dark photons. Using charged scalar fields will be the last extension we will make.

In addition to the fields for the complex scalar field,

$$\phi_1, \quad X_1, \quad Y_1, \qquad \text{and} \qquad \phi_2, \quad X_2, \quad Y_2, \tag{17}$$

we have the fields

$$A_t, \qquad A_r, \qquad Y, \qquad \Omega, \tag{18}$$

that describe the dark photons. Note that the field Y is different than Y_1, Y_2. It may be best to give Y a different symbol that Y in your code. We also have a new parameter,

$$g, \tag{19}$$

which is the amount of charge the scalar field has. g is analogous to the electric charge of a particle.

The equations for the complex scalar field are

$$\begin{aligned}
\dot{\phi}_1 &= \frac{\alpha}{a} Y_1 - g A_t \phi_2 \\
\dot{X}_1 &= \partial_r \left(\frac{\alpha}{a} Y_1 \right) - g A_t X_2 + g A_r \left(\frac{\alpha}{a} Y_2 \right) + g \frac{\alpha a}{r^2} Y \phi_2 \\
\dot{Y}_1 &= \frac{1}{r^2} \partial_r \left(\frac{r^2 \alpha}{a} X_1 \right) - g \left(Y_2 A_t - \frac{\alpha}{a} X_2 A_r \right) - \alpha a [\mu^2 + 2\lambda(\phi_1^2 + \phi_2^2)] \phi_1
\end{aligned} \tag{20}$$

and

$$\begin{aligned}
\dot{\phi}_2 &= \frac{\alpha}{a} Y_2 + g A_t \phi_1 \\
\dot{X}_2 &= \partial_r \left(\frac{\alpha}{a} Y_2 \right) + g A_t X_1 - g A_r \left(\frac{\alpha}{a} Y_1 \right) - g \frac{\alpha a}{r^2} Y \phi_1 \\
\dot{Y}_2 &= \frac{1}{r^2} \partial_r \left(\frac{r^2 \alpha}{a} X_2 \right) + g \left(Y_1 A_t - \frac{\alpha}{a} X_1 A_r \right) - \alpha a [\mu^2 + 2\lambda(\phi_1^2 + \phi_2^2)] \phi_2 .
\end{aligned} \tag{21}$$

The equations for the photon are

$$\begin{aligned}
\dot{\Omega} &= \frac{1}{r^2} \partial_r \left[r^2 \frac{\alpha}{a} A_r \right] \\
A_t &= \frac{\alpha}{a} \Omega \\
\dot{A}_r &= \frac{\alpha a}{r^2} Y + A_t' \\
\dot{Y} &= -\alpha a r^2 J^r \\
Y' &= \alpha a r^2 J^t ,
\end{aligned} \tag{22}$$

where

$$J^t = \frac{g}{\alpha a} \left(\phi_1 Y_2 - \phi_2 Y_1 \right), \qquad J^r = -\frac{g}{a^2} \left(\phi_1 X_2 - \phi_2 X_1 \right). \tag{23}$$

Note that Ω and A_r have evolution equations and A_t has an algebraic equation. In fact, it is probably best to simply replace everywhere you see A_t with $\alpha\Omega/a$, are simply remove A_t in favor of Ω. Further, the Ω equation has a form similar to the Y equation and therefore must be finite differences using the r^3 method. Lastly, note that there are two equations for Y. This gives a choice: We can either evolve Y using the $\dot{Y}$ equation or solve for Y similar to how we solve for α using the Y' equation. It's not yet clear which one is better. For now, let's go with the α method and use the Y' equation.

The outer boundary conditions for the complex scalar field are

$$\begin{aligned}
&\dot{\phi}_1 = -\frac{\phi_1}{r} - X_1 + g A_r \phi_2, \qquad \dot{Y}_1 = -\frac{Y_1}{r} - Y_1', \qquad X_1 = -\frac{\phi_1}{r} - Y_1 + g A_t \phi_2 + g A_r \phi_2, \\
&\dot{\phi}_2 = -\frac{\phi_2}{r} - X_2 - g A_r \phi_1, \qquad \dot{Y}_2 = -\frac{Y_2}{r} - Y_2', \qquad X_2 = -\frac{\phi_2}{r} - Y_2 - g A_t \phi_1 - g A_r \phi_1.
\end{aligned} \tag{24}$$

For the outer boundary with the dark photons, we have

$$\dot{A}_r = -A_r', \qquad \dot{\Omega} = -\Omega', \qquad \dot{Y} = g r^2 \left(\phi_1 X_2 - \phi_2 X_1 \right). \tag{25}$$

If we use the Y' equation in (22) to determine Y, then we need an inner boundary condition, which is

$$Y(0) = 0. \tag{26}$$

The other inner boundary conditions are that ϕ_1, Y_1, ϕ_2, Y_2, A_t, Ω, a, and α are even, while X_1, X_2, A_r, and Y are odd.

The energy-momentum tensor components are

$$\begin{aligned}
(T_b)^t{}_t &= -\frac{1}{a^2}\left(X_1^2 + X_2^2 + Y_1^2 + Y_2^2\right) - V - \frac{Y^2}{2r^4} \\
(T_b)^r{}_r &= +\frac{1}{a^2}\left(X_1^2 + X_2^2 + Y_1^2 + Y_2^2\right) - V - \frac{Y^2}{2r^4} \\
(T_b)^t{}_r &= -\frac{2}{\alpha a}\left(X_1 Y_1 + X_2 Y_2\right),
\end{aligned} \tag{27}$$

where, just as before

$$V = \mu^2(\phi_1^2 + \phi_2^2) + \lambda(\phi_1^2 + \phi_2^2)^2. \tag{28}$$

[Possible issue: it may be that $g \to \mu g$ because of scaling. I will need to work this out.]

4 7/13/21: Static solutions for fermion stars

As you know, we can compute static (i.e. time-independent) solutions. We can then use these solutions as the initial data for a time-dependent evolution. In this section I'll show you the equations for computing static solutions for fermion stars. Later, we'll learn how to find the static solutions for fermion-boson stars and for fermion-charged boson stars.

As you know, we have been describing gravity through the variables α and a. When finding static solutions, it is convenient to switch to a different set of variables:

$$\sigma \equiv a\alpha, \qquad m = \frac{r}{2}\left(1 - \frac{1}{a^2}\right). \tag{29}$$

$m(r)$ gives the total mass of the system inside a radius r (actually, it's the total energy, but the convention is to refer to it as mass). σ (like a and α) does not have as simple a physical interpretation.

Static solutions are described by the variables $\rho(r)$ and $P(r)$, which are the energy density and pressure, which we are already familiar with. They are functions of r, the distance from the center of the star and are no longer functions of time. As you know, they are linked through an equation of state, $\rho(P)$. The equations are

$$\begin{aligned}
\frac{dm}{dr} &= 4\pi r^2 \rho \\
\frac{dP}{dr} &= -\frac{(4\pi r^3 P + m)(\epsilon + P)}{r^2 N},
\end{aligned} \tag{30}$$

where

$$N \equiv 1 - \frac{2m}{r}. \tag{31}$$

There is also an equation for σ, but since σ decouples from these two equations, it does not have to be solved for. These are called the Tolman-Oppenheimer-Volkoff (TOV) equations.

Solutions to the TOV equations are the static solutions we are after. They are uniquely identified by $P(0)$, which we call the *central pressure*. Thus, you specify a central pressure (say, $P(0) = 10^{-4}$) and, using the the inner boundary condition

$$m(0) = 0, \tag{32}$$

you integrate the TOV equations outward from a very small r (which is effectively $r = 0$). The edge of the star, at radial position $r = R$, is defined by

$$P(R) = 0 \tag{33}$$

(or really, a very small value of P, such as $P \sim 10^{-10}$ to 10^{-14}). The total mass of the star is given by $M = m(R)$.

I would like you to write Python code and use the `solve_ivp` command to solve the TOV equations in (30). The `solve_ivp` uses Runge-Kutta to solve the equation and therefore solves it in integration steps, just as you do in your evolution code. At teach integration step, the TOV equations will have found the value m and P at some r. You need to know the value of ρ so that you can use it for the right hand side of the bottom equation in (30). You determine this using the equation of state.

There is one other tricky part. I would like you to look up "events" for `solve_ivp`. The differential equation solver can be told to trigger some code if the solution it is finding matches some criteria. This is what is meant by an event. For us, we want the solver to terminate as soon as P drops below some value, such as 10^{-10}. This way, as soon as we hit the edge of the star, the integrator stops.

After you have working code, there are two things to try. The first is to loop through central pressure values from $10^{-6} < P(0) < 10^{-1}$. For the results you can plot M vs. R, which is the mass-radius diagram, as well as M vs. $P(0)$. The latter diagram will tell us the value of the central pressures for which the static solutions are *linearly unstable*.

The second thing to do is to write the ρ values for a given static solution (and the pressure values, if you'd like) to file and then load them into your evolution code to evolve. We expect to see the stable solutions holding their configuration and the unstable ones evolving to something else.

5 7/13/21: The shooting method

What is coming up next, I'd like you to use a numerical method called the shooting method. To get a sense for it, I would like you to watch six YouTube videos:

https://www.youtube.com/watch?v=6ONtZz0fmvw&list=PLC245A479FD6059DB&index=1

Make sure to watch them in order ("Background," "The Method," and then parts 1 to 4).

6 7/15/21: Static solutions for boson stars

As a warm-up to fermion-boson stars and fermion-charged-boson stars, we'll start with static solutions for just boson stars. Boson stars are made of a complex-scalar field. The equations to solve are

$$\begin{aligned}
\varphi'' &= \varphi'\left(\frac{a'}{a} - \frac{\alpha'}{\alpha} - \frac{2}{r}\right) - \frac{\hat{\omega}^2}{N^2\hat{\sigma}^2}\varphi + \frac{1}{N}\left(\mu^2\varphi + 2\lambda\varphi^3\right) \\
\hat{\sigma}' &= \hat{\sigma}\left(\frac{a'}{a} + \frac{\alpha'}{\alpha}\right) \\
m' &= -4\pi r^2 (T_b)^t{}_t,
\end{aligned} \tag{34}$$

where

$$\begin{aligned}
\frac{\alpha'}{\alpha} &= +\frac{4\pi r}{N}(T_b)^r{}_r + \frac{m}{r^2 N} \\
\frac{a'}{a} &= -\frac{4\pi r}{N}(T_b)^t{}_t - \frac{m}{r^2 N} \\
N &= 1 - \frac{2m}{r}
\end{aligned} \tag{35}$$

and

$$\begin{aligned}
(T_b)^t{}_t &= -\frac{\hat{\omega}^2}{N\hat{\sigma}^2}\varphi^2 - N\varphi'^2 - \mu^2\varphi^2 - \lambda\varphi^4 \\
(T_b)^r{}_r &= +\frac{\hat{\omega}^2}{N\hat{\sigma}^2}\varphi^2 + N\varphi'^2 - \mu^2\varphi^2 - \lambda\varphi^4.
\end{aligned} \tag{36}$$

φ is the scalar field and is real. The way it works out is that the complex scalar field, ϕ, gets written as

$$\phi = \varphi e^{i\omega t}, \tag{37}$$

where φ is real and ω is a real constant. We will have to determine the value of ω when finding a solution. We'll do this using the shooting method. μ is the mass, which we can set to 1 for now (but write your code so that it's value can be changed), and λ is the self-coupling constant, which we will set to 0 (but still include it in your code).

The only matter field to solve for is φ. We have also the gravity fields $\hat{\sigma}$ and m and the parameter $\hat{\omega}$. Each static solution is uniquely identified by $\varphi(0)$, which we supply. We then use $m(0) = 0$, $\hat{\sigma}(0) = 1$, $\phi'(0) = 0$, and a *trial value* for $\hat{\omega}$. We take $\hat{\omega}$ to be a shooting parameter that we tune so that are solution satisfies the outer boundary conditions, which is simply that $\varphi \to 0$ as $r \to \infty$.

You will find that for any value of ω, eventually the fields (such as φ') shoot off to infinity. You will want to write an event that terminates if φ' or φ or some field gets too large. Depending on which way it shoots off, this will tell you how to tune $\hat{\omega}$. You should start by doing this by hand, i.e. tuning $\hat{\omega}$ by hand and plotting solutions as go and not yet implementing events. This will allow you to see the pattern. You can then implement events and automate searching for the value of $\hat{\omega}$, which you can do to arbitrary precision.

The reason $\hat{\omega}$ and $\hat{\sigma}$ have hats on them is because they have been scaled so that $\hat{\sigma}(0) = 1$, making the inner boundary conditions simpler. Once you have a solution, you can unscale $\hat{\omega}$ to find its physical value:

$$\omega = \frac{\hat{\omega}}{\hat{\sigma}(\infty)}, \tag{38}$$

where $\hat{\sigma}(\infty)$ is the value of $\hat{\sigma}$ at the outer edge of the solution. In practice, you can take the value of r for $|\varphi|$ is smallest to be the outer edge of the solution. Beyond this value, φ usually starts moving away from zero because the solution is being lost out there.

7 7/19/21: Static solutions for fermion-boson stars

We will now look at mixed stars, with a fermion sector and with a boson sector, where the boson is a neutral (i.e. uncharged) complex scalar field. This is a combination of the two static solutions you have code for, but there are some important differences. The equations are conveniently written in terms of the components of the energy-momentum tensor. For the fermion sector,

$$(T_f)^t{}_t = -\rho, \qquad (T_f)^r{}_r = P. \tag{39}$$

ρ and P are linked through the equation of state, $\rho(P)$. For the boson sector, it is the same as before,

$$\begin{aligned}
(T_b)^t{}_t &= -\frac{\hat{\omega}^2}{N\hat{\sigma}^2}\varphi^2 - N\varphi'^2 - \mu^2\varphi^2 - \lambda\varphi^4 \\
(T_b)^r{}_r &= +\frac{\hat{\omega}^2}{N\hat{\sigma}^2}\varphi^2 + N\varphi'^2 - \mu^2\varphi^2 - \lambda\varphi^4.
\end{aligned} \tag{40}$$

φ is the scalar field and is real. As before, we will have to determine the value of ω when finding a solution.

From now on, I would like you to start using

$$\mu = 1.122089. \tag{41}$$

This is the correct value so that the scalar field has a mass of 10^{-10} eV. This (very small) mass is the value needed for both the boson sector and fermion sector to contribute appreciably to the star, which is the most interesting case. You'll need to start using this value in your evolution code as well. Stick with $\lambda = 0$.

The equations to solve are

$$\begin{aligned}
\varphi'' &= \varphi'\left(\frac{a'}{a} - \frac{\alpha'}{\alpha} - \frac{2}{r}\right) - \frac{\hat{\omega}^2}{N^2\hat{\sigma}^2}\varphi + \frac{1}{N}\left(\mu^2\varphi + 2\lambda\varphi^3\right) \\
P' &= -\frac{\alpha'}{\alpha}(\rho + P) \\
\sigma' &= \sigma\left(\frac{a'}{a} + \frac{\alpha'}{\alpha}\right) \\
m' &= -4\pi r^2 (T_{\text{tot}})^t{}_t,
\end{aligned} \tag{42}$$

where

$$\begin{aligned}
\frac{\alpha'}{\alpha} &= +\frac{4\pi r}{N}(T_{\text{tot}})^r{}_r + \frac{m}{r^2 N} \\
\frac{a'}{a} &= -\frac{4\pi r}{N}(T_{\text{tot}})^t{}_t - \frac{m}{r^2 N} \\
N &= 1 - \frac{2m}{r}
\end{aligned} \tag{43}$$

and

$$(T_{\text{tot}})^\mu{}_\nu = (T_f)^\mu{}_\nu + (T_b)^\mu{}_\nu. \tag{44}$$

The fields to solve for are φ and P. Each static solution is uniquely identified by the central values $P(0)$ and $\varphi(0)$. You therefore have to specify two values. The inner boundary conditions are as before: $m(0) = 0$, $\hat{\sigma}(0) = 1$, $\varphi'(0) = 0$. In the fermion sector, at each integration step, you know P and you use the equation of state to find ρ. The edge of the fermion sector occurs at $r = R_f$, defined by $P(R_f) = 0$ (or when P is very small). The outer boundary condition for φ is $\varphi \to 0$ as $r \to \infty$. Just as with the boson star, you want to use the shooting method to determine $\hat{\omega}$, so that φ satisfies the outer boundary condition.

For the fermion-boson star (as opposed to the fermion-charged-boson star) I'm pretty sure the outer edge of the fermion sector always occurs before the outer edge of the boson sector. (Technically, there is not suppose to be an outer edge of the boson sector, since φ is suppose to asymptotically head to zero. But, with the fermion-charged-boson star, it can be that the φ reaches such a small value, that the edge of the boson sector is effectively reached before the edge of the fermion sector.) As such, I'm pretty sure you can safely assume that P will reach zero first. At this point, you want to break the evolution using an event. You then restart the evolution using *only* the pure boson-star equations (i.e. single-sector equations), continuing the solution to the edge of the boson sector, as well as continuing the determination of $\hat{\omega}$.

Because the static solutions are identified by two parameters ($\varphi(0)$ and $P(0)$), we can no longer make mass-radius *curves*. What we can make is more complicated and we'll skip that for now. Instead, start importing the static solutions your evolution code. My expectation is that stable solutions will stay-put as usual, but that some unstable solutions will migrate to a stable solution.

To make static solutions for your evolution code, you want to use

$$\phi_1 = \varphi \qquad X_1 = \varphi' \qquad Y_1 = 0 \tag{45}$$

$$\phi_2 = 0 \qquad X_2 = 0 \qquad Y_2 = \frac{\hat{\omega}\varphi}{N\hat{\sigma}}. \tag{46}$$

8 7/22/21: Static solutions for fermion-charged-boson stars

We will now look at mixed stars, with a fermion sector and with a boson sector, where the boson is a charged complex scalar field. For the fermion sector, the energy-momentum tensor is as before

$$(T_f)^t{}_t = -\rho, \qquad (T_f)^r{}_r = P. \tag{47}$$

ρ and P are linked through the equation of state, $\rho(P)$. For the boson sector, there is no longer the parameter ω. The energy-momentum tensor is

$$\begin{aligned}(T_b)^t{}_t &= -N\varphi'^2 - \frac{\mu^2 g^2 A_t^2 \varphi^2}{N\sigma^2} - \mu^2\varphi^2 - \lambda\varphi^4 - \frac{A_t'^2}{2\sigma^2}\\ (T_b)^r{}_r &= +N\varphi'^2 + \frac{\mu^2 g^2 A_t^2 \varphi^2}{N\sigma^2} - \mu^2\varphi^2 - \lambda\varphi^4 - \frac{A_t'^2}{2\sigma^2}\end{aligned} \tag{48}$$

φ is the scalar field and is real and A_t is called the gauge field and describes a photon. Just as with fermion-boson stars, use

$$\mu = 1.122089. \tag{49}$$

This is the correct value so that the scalar field has a mass of 10^{-10} eV. This (very small) mass is the value needed for both the boson sector and fermion sector to contribute appreciably to the star, which is the most interesting case. You'll need to start using this value in your evolution code as well. Stick with $\lambda = 0$.

The equations to solve are

$$
\begin{aligned}
\sigma' &= \frac{4\pi r\sigma}{N}\left[(T_{\rm tot})^r{}_r - (T_{\rm tot})^t{}_t\right]\\
m' &= -4\pi r^2 (T_{\rm tot})^t{}_t.\\
\varphi'' &= -\left\{\frac{2}{r} + \frac{4\pi r}{N}\left[(T_{\rm tot})^r{}_r + (T_{\rm tot})^t{}_t\right] + \frac{2m}{Nr^2}\right\}\varphi' - \frac{1}{N}\left(\frac{\mu^2 g^2 A_t^2}{N\sigma^2} - \mu^2 - 2\lambda\phi^2\right)\phi \\
A_t'' &= \left\{\frac{4\pi r}{N}\left[(T_{\rm tot})^r{}_r - (T_{\rm tot})^t{}_t\right] - \frac{2}{r}\right\}A_t' + \frac{2\mu^2 g^2}{N}A_t\phi^2\\
p' &= -\frac{\alpha'}{\alpha}(\epsilon + p).
\end{aligned}
\tag{50}
$$

where

$$
\begin{aligned}
N &= 1 - \frac{2m}{r}\\
\frac{\alpha'}{\alpha} &= \frac{4\pi r^3 (T_{\rm tot})^r{}_r + m}{r^2 N}.
\end{aligned}
\tag{51}
$$

The fields to solve for are φ, A_t, and P. Each static solution is uniquely identified by the central values $P(0)$ and $\varphi(0)$. You therefore have to specify two values. The inner boundary conditions are as before: $m(0) = 0$, $\hat{\sigma}(0) = 1$, $\varphi'(0) = 0$. In the fermion sector, at each integration step, you know P and you use the equation of state to find ρ. The edge of the fermion sector occurs at $r = R_f$, defined by $P(R_f) = 0$ (or when P is very small).

For the boson sector, you use the shooting method. This time, the shooting parameter is $A_t(0)$, while $A_t'(0) = 0$. The outer boundary conditions are $\varphi \to 0$ and $A_t' \to 0$ as $r \to 0$. There is no condition of A_t as $r \to 0$.

There is a complication compared to the fermion-boson star. I'm pretty sure there are choices for the charge, g, such that the boson sector ends at a radius r before the fermion sector ends. What this means is that the φ will drop to such a small value, that it is effectively zero and we reach the edge of the fermion sector first. Your code, then, must allow for either the fermion or the boson sector to meet its outer edge first. You then break the integration and restart it with the single-fluid equations. These should be the single-fluid equations for either a pure fermion star or a pure charged-boson star.

Once you have the solution, you can import this into your evolution code as

$$\phi_1 = \varphi \qquad X_1 = \varphi' \qquad Y_1 = 0 \tag{52}$$

$$\phi_2 = 0 \qquad X_2 = 0 \qquad Y_2 = -\frac{\mu g A_t \varphi}{N\sigma} \tag{53}$$

$$\Omega = \frac{A_t}{\sigma N} \qquad Z = -r^2\frac{A_t'}{\sigma} \tag{54}$$

Recall that Ω replaces A_t in the evolution code. You can also take $A_r = 0$ initially.

Finally, remember to use

$$g = g_*\sqrt{8\pi}, \tag{55}$$

where g_* takes simple values (say, 0.5, 0.65, and 0.7). Don't take $g_* > 0.7$.

[There is a possible mistake in the equations for the evolution code for the fermion-charged-boson star. It is a small one that I will need to work out at a later time. I made a note of it at the end of section 3. I would appreciate if you could remind me about this sometime in the Fall semester.]