Prelude

Programs: Requirements

- In this course we will use an adaptive step 4(5) order Runge-Kutta method with a shooting strategy.
- You will need a C compiler and a 11+ Mathematica.

- The required files for the hands on lecture can be downloaded in

CODES

Programs: C Integrator

- If you have a linux machine it is already installed (*a priori*) and can be run in the terminal.
- If you use a Windows/Mac operating system, you need to install a proper compiler:

  How To Install GCC

- Please follow the tutorial.

Programs: Running

- In the CODES file you have an isolated SOLUTION generator and a DOMAIN space generator.
- Open the terminal in the desired folder and run the executable. There is already an executable file

To run the executable:

```
$ sh Exec.sh
```

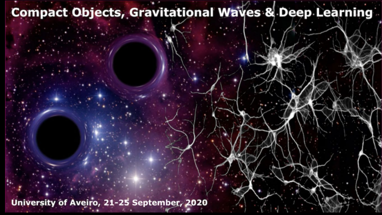
If you want to run by hand:

```
$  gcc -ofast BS.c -o S -lm
$  ./S
```

- After generating the solutions you can go back to the initial file and run the MATH notebooks.

Programs: Final thoughts

- Please run the files to test if everything is working.
- Have a nice weekend.
- See you monday.



How we numerically solve ODEs: Boson Star

Alexandre M. Pombo

Introduction: Physics

- In physics, and in particular in GR, the equations that describe a system and its evolution is described by differential equations.
- Which can be divided into ODEs and PDEs.
- There is a series of problems that do not have an algebraic solutions.
- For those problems, a numerical strategy is required for the numerical integration and to implement proper boundary conditions.

Introduction: Physics

- In physics, and in particular in GR, the equations that describe a system and its evolution is described by differential equations.
- Which can be divided into ODEs and PDEs.
- There is a series of problems that do not have an algebraic solutions.
- For those problems, a numerical strategy is required for the numerical integration and to implement proper boundary conditions.

Introduction: Physics

- In physics, and in particular in GR, the equations that describe a system and its evolution is described by differential equations.
- Which can be divided into ODEs and PDEs.
- There is a series of problems that do not have an algebraic solutions.
- For those problems, a numerical strategy is required for the numerical integration and to implement proper boundary conditions.

Introduction: Numerics

- The aim of this hands on lecture is to understand the construction of a viable research numerical program.

- The developed program will be able to numerically solve ODEs.

- It is written in a C where:
  - The integrator is a 4(5)th O Runge-Kutta method;
  - The BC are implemented through a secant algorithm.

- Let us implement in the Boson Star problem.

Introduction: Numerics

- The aim of this hands on lecture is to understand the construction of a viable research numerical program.

- The developed program will be able to numerically solve ODEs.

- It is written in a C where:
  - The integrator is a 4(5)th O Runge-Kutta method;
  - The BC are implemented through a secant algorithm.

- Let us implement in the Boson Star problem.

Introduction: Numerics

- The aim of this hands on lecture is to understand the construction of a viable research numerical program.

- The developed program will be able to numerically solve ODEs.

- It is written in a C where:
  - The integrator is a 4(5)th O Runge-Kutta method;
  - The BC are implemented through a secant algorithm.

- Let us implement in the Boson Star problem.

Introduction: Numerics

- The aim of this hands on lecture is to understand the construction of a viable research numerical program.

- The developed program will be able to numerically solve ODEs.

- It is written in a C where:
  - The integrator is a 4(5)th O Runge-Kutta method;
  - The BC are implemented through a secant algorithm.

- Let us implement in the Boson Star problem.

Boson Star Equations

The Field equations

- The set of field equations that describe a spherically symmetric BS are

$$m' = 4\pi r^2 \left(N(\varphi')^2 + \mu^2\varphi^2 + \frac{\omega^2\varphi^2}{N\sigma^2} \right) \,,$$

$$\sigma' = 8\pi\sigma r \left((\varphi')^2 + \frac{\omega^2\varphi^2}{N^2\sigma^2} \right) \,,$$

$$\varphi'' = -\frac{2\varphi'}{r} - \frac{N'\varphi'}{N} - \frac{\sigma'\varphi'}{\sigma} + \frac{\mu^2\varphi}{N} - \frac{\omega^2\varphi}{N^2\sigma^2} \,.$$

The Field equations

- The set of field equations that describe a spherically symmetric BS are

$$m' = 4\pi r^2 \left(N(\varphi')^2 + \mu^2\varphi^2 + \frac{\omega^2\varphi^2}{N\sigma^2} \right) ,$$

$$\sigma' = 8\pi\sigma r \left((\varphi')^2 + \frac{\omega^2\varphi^2}{N^2\sigma^2} \right) ,$$

$$\varphi'' = -\frac{2\varphi'}{r} - \frac{N'\varphi'}{N} - \frac{\sigma'\varphi'}{\sigma} + \frac{\mu^2\varphi}{N} - \frac{\omega^2\varphi}{N^2\sigma^2} .$$

Which are written in the form:

$$\frac{d^n y}{dx^n} = f(x, y, y', \ldots, y^{(n)})$$

Asymptotic approximations: Origin

At the origin,

$$m(0) = 0\,,\ \sigma(0) = \sigma_0\,,\ \varphi(0) = \varphi_0\,,$$

The field equations can be approximated by a power series expansion

$$m(r) \simeq \frac{4\pi\left(\mu^2\sigma_0^2+\omega^2\right)}{3\sigma_0}\varphi_0^2\, r^3 + \mathcal{O}(r^5)\,,$$

$$\sigma(r) \simeq \sigma_0 + \frac{4\pi\omega\varphi^2}{\sigma_0} r^2 + \mathcal{O}(r^4)\,,$$

$$\varphi(r) \simeq \varphi_0 + \frac{1}{6}\left(\mu^2 - \frac{\omega^2}{\sigma_0^2}\right)\varphi_0 r^2 + \mathcal{O}(r^4)\,.$$

Asymptotic approximations: At infinity

- At infinity we require asymptotically flatness

$$m(\infty) = M\,,\ \sigma(\infty) = 1\,,\ \varphi(\infty) = 0\,,$$

- The above condition for the metric function σ_0 fixes the symmetry of scale invariance $\{\sigma, \omega\} \to \lambda\{\sigma, \omega\}$, with λ a positive constant.

- The problem reduces to a two parameters shooting for φ_0 and σ_0

Asymptotic approximations: At infinity

- At infinity we require asymptotically flatness

$$m(\infty) = M\,,\ \sigma(\infty) = 1\,,\ \varphi(\infty) = 0\,,$$

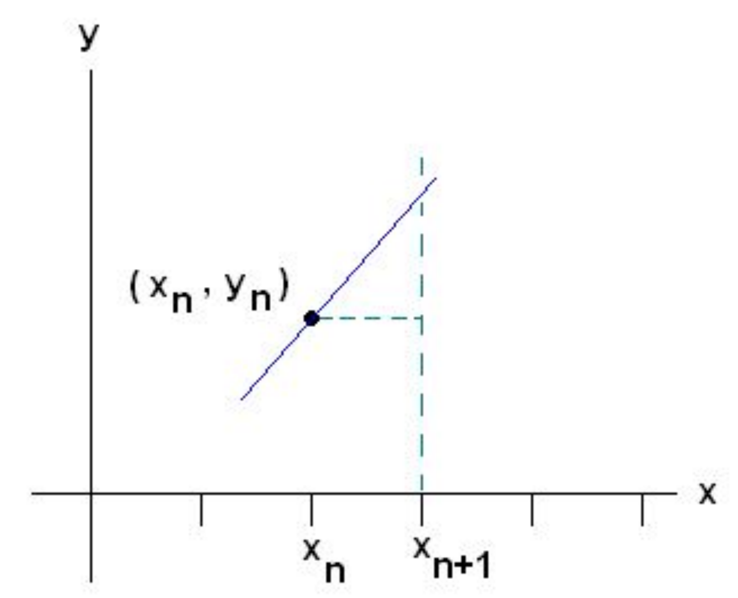
- The above condition for the metric function σ_0 fixes the symmetry of scale invariance $\{\sigma, \omega\} \longrightarrow \lambda\{\sigma, \omega\}$, with λ a positive constant.

- The problem reduces to a two parameters shooting for φ_0 and σ_0

Adaptive step Runge-Kutta

Numerical Procedure: Integrator

- The integrator is based on an adaptive Runge-Kutta strategy: RK(5)6



Numerical Procedure: Adaptive Step

- To implement an adaptive step, you need to compute the error/changing rate of a given function

- One can do that in several ways:
  - Calculate the same point with two different methods;
  - Calculate with two different steps;
  - Calculate the same point with two different order methods.

Numerical Procedure: Adaptive Step

- To implement an adaptive step, you need to compute the error/changing rate of a given function

- One can do that in several ways:
    - Calculate the same point with two different methods;
    - Calculate with two different steps;
    - Calculate the same point with two different order methods.

Numerical Procedure: Adaptive Step

- To implement an adaptive step, you need to compute the error/changing rate of a given function

- One can do that in several ways:
    - Calculate the same point with two different methods;
    - Calculate with two different steps;
    - Calculate the same point with two different order methods.

Numerical Procedure: Adaptive Step

- To implement an adaptive step, you need to compute the error/changing rate of a given function

- One can do that in several ways:
  - Calculate the same point with two different methods;
  - Calculate with two different steps;
  - Calculate the same point with two different order methods.

Numerical Procedure: Adaptive Step

- To implement an adaptive step, you need to compute the error/changing rate of a given function

- One can do that in several ways:
    - Calculate the same point with two different methods;
    - Calculate with two different steps;
    - Calculate the same point with two different order methods.

Dormand-Prince method

$$
\begin{array}{c|ccccccc}
0 & & & & & & & \\
\frac{1}{5} & \frac{1}{5} & & & & & & \\
\frac{3}{10} & \frac{3}{40} & \frac{9}{40} & & & & & \\
\frac{4}{5} & \frac{44}{45} & -\frac{56}{15} & \frac{32}{9} & & & & \\
\frac{8}{9} & \frac{19372}{6561} & -\frac{25360}{2187} & \frac{64448}{6561} & -\frac{212}{729} & & & \\
1 & \frac{9017}{3168} & -\frac{355}{33} & \frac{46732}{5247} & \frac{49}{176} & -\frac{5103}{18656} & & \\
1 & \frac{35}{384} & 0 & \frac{500}{1113} & \frac{125}{192} & -\frac{2187}{6784} & \frac{11}{84} & \\
\hline
y & \frac{35}{384} & 0 & \frac{500}{1113} & \frac{125}{192} & -\frac{2187}{6784} & \frac{11}{84} & 0 \\
\hline
\hat{y} & \frac{5179}{57600} & 0 & \frac{7571}{16695} & \frac{393}{640} & -\frac{92097}{339200} & \frac{187}{2100} & \frac{1}{40} \\
\hline
\end{array}
$$

Numerical Scheme: Integrator

ODEs functions

↓

Integration for a step dx

Numerical Scheme: Integrator

ODEs functions

↓

Integration for a step dx

↓

Compute the error 4(5)

Numerical Scheme: Integrator

ODEs functions

Integration for a step dx

Compute the error 4(5)

Error is larger than a given tolerance

Numerical Scheme: Integrator

ODEs functions

↓

Integration for a step dx

↓

Compute the error 4(5)

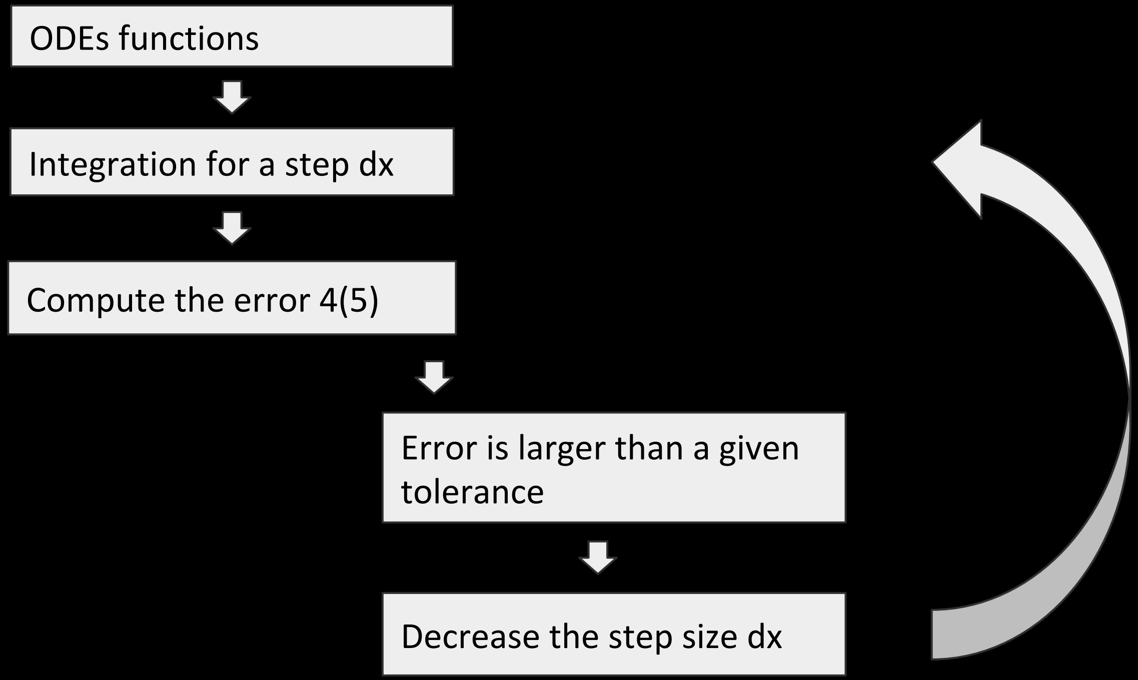
↓

Error is larger than a given tolerance

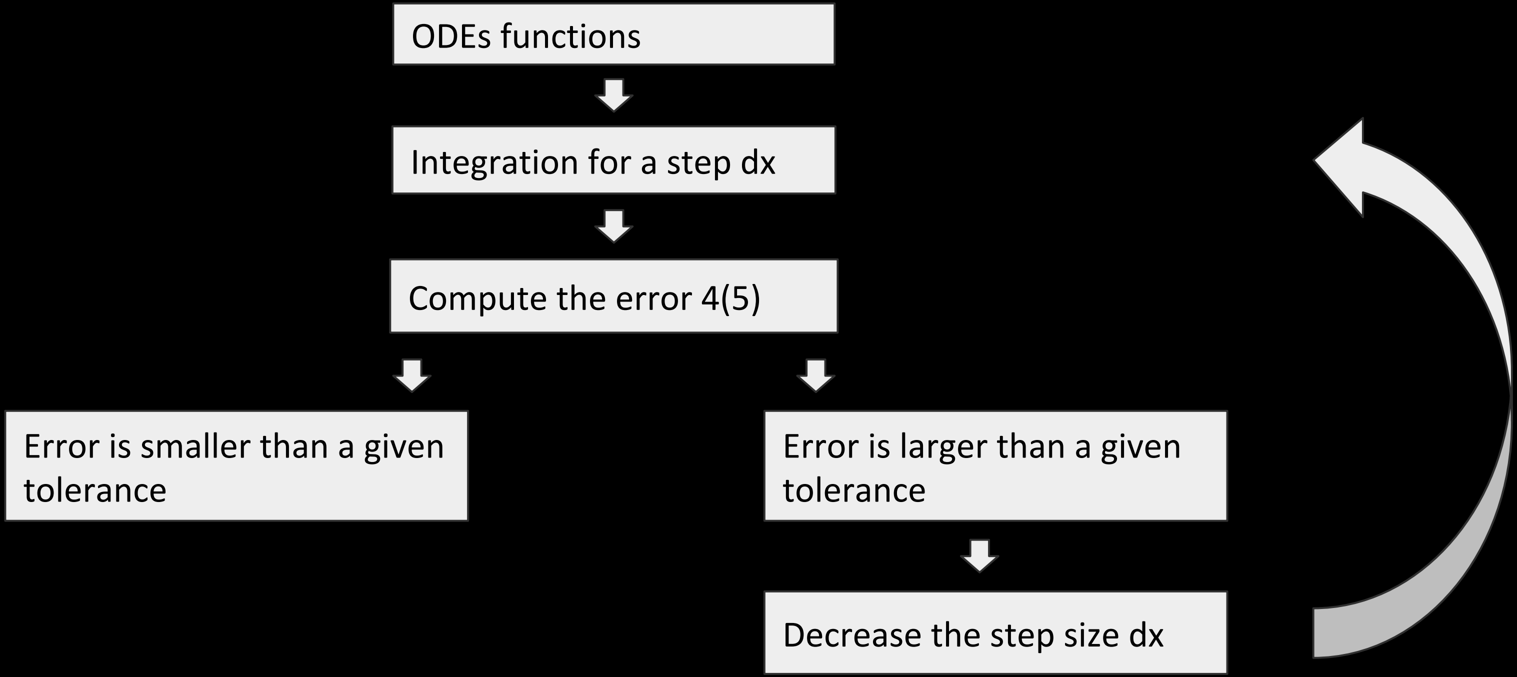
↓

Decrease the step size dx

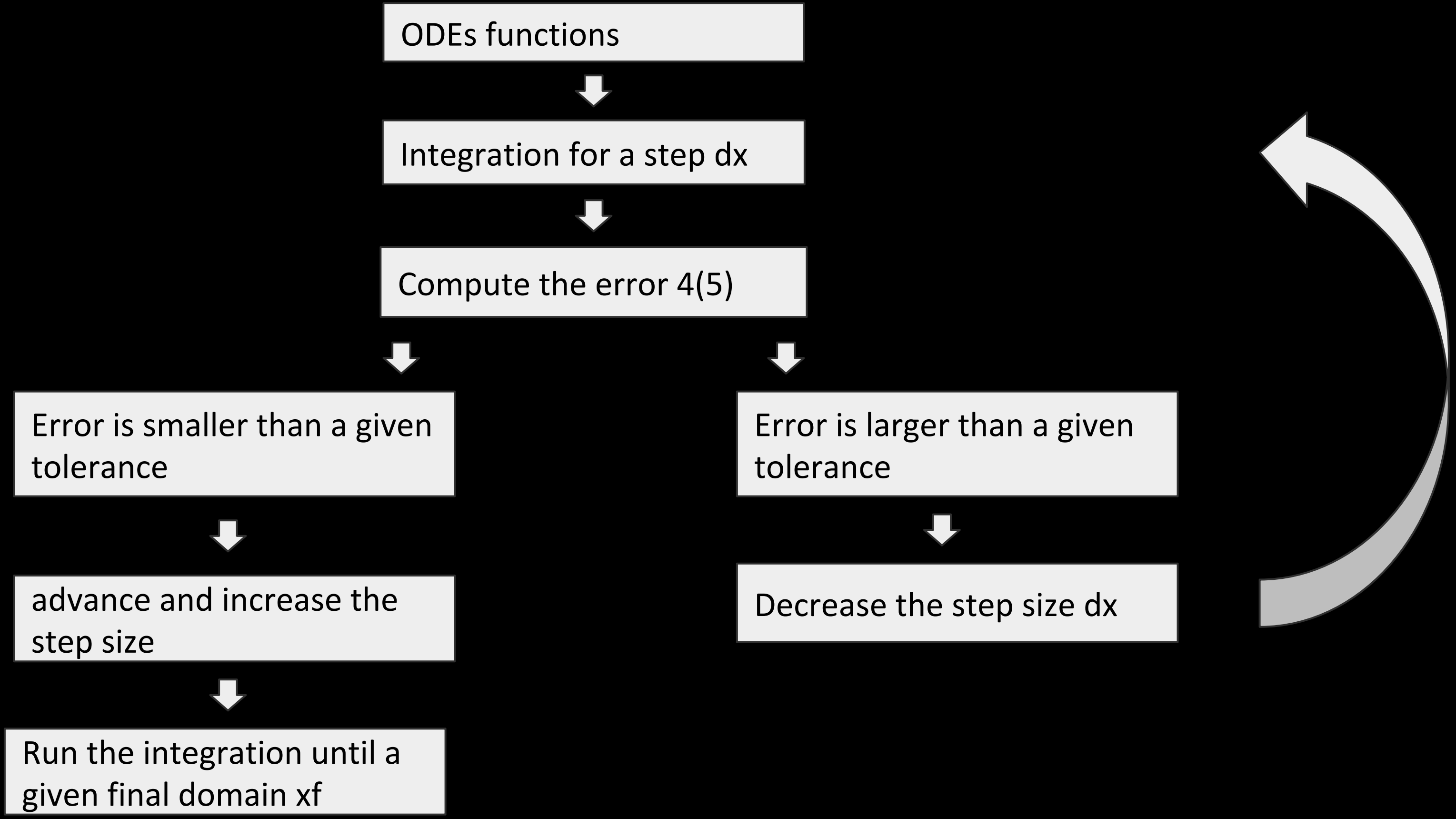
Numerical Scheme: Integrator



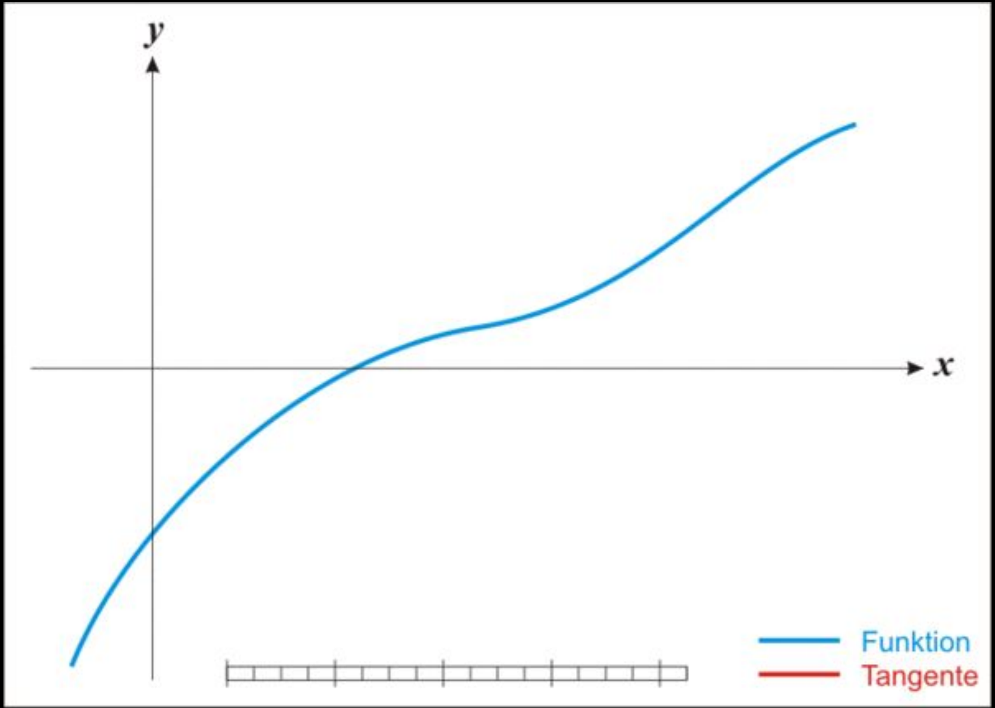
Numerical Scheme: Integrator



Numerical Scheme: Integrator



Shooting Strategy: Secant method



Numerical Procedure: Shooting

- We only have two unknown parameters
- These are φ_0 and σ_0
- The σ_0 parameters can be solved by a symmetry property
- The asymptotic conditions are solved through a sechant strategy on φ_0

Numerical Procedure: Shooting

- We only have two unknown parameters
- These are φ_0 and σ_0
- The σ_0 parameters can be solved by a symmetry property
- The asymptotic conditions are solved through a sechant strategy on φ_0

Numerical Procedure: Shooting

- We only have two unknown parameters
- These are φ_0 and σ_0
- The σ_0 parameters can be solved by a symmetry property
- The asymptotic conditions are solved through a sechant strategy on φ_0

Numerical Scheme: Shooting

Suggest an initial guess for the solution

↓

Run the integration until a given final domain xf

Numerical Scheme: Shooting

Suggest an initial guess for the solution

Run the integration until a given final domain xf

Evaluate the final value of the scalar field

Numerical Scheme: Shooting

Suggest an initial guess for the solution

↓

Run the integration until a given final domain xf

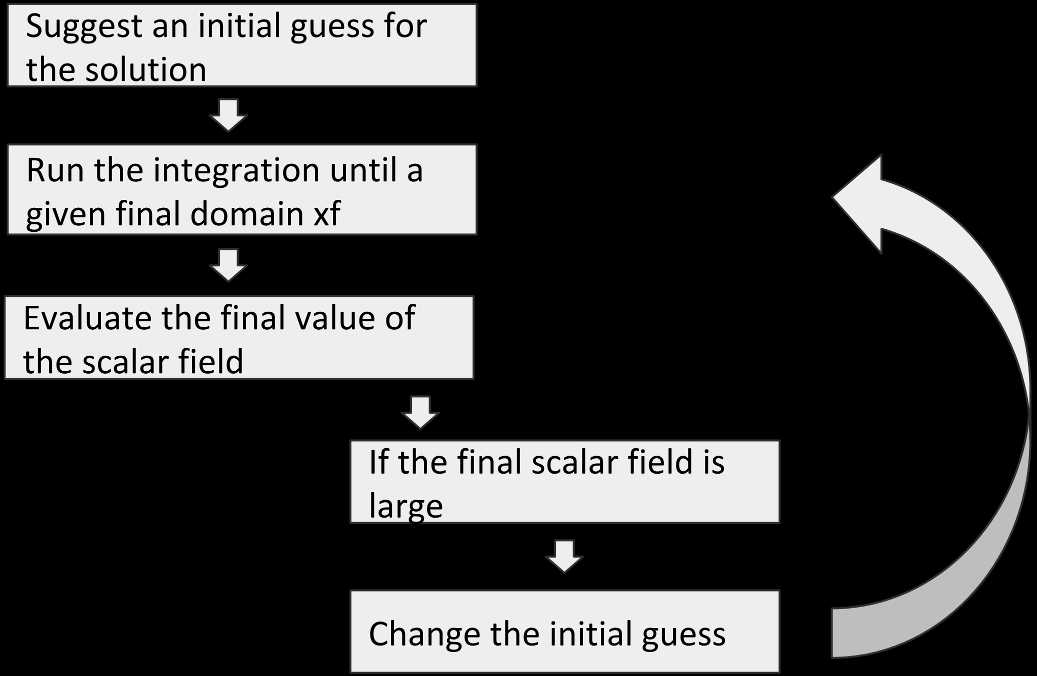
↓

Evaluate the final value of the scalar field

↓

If the final scalar field is large

Numerical Scheme: Shooting



Numerical Scheme: Shooting

Suggest an initial guess for the solution

Run the integration until a given final domain xf

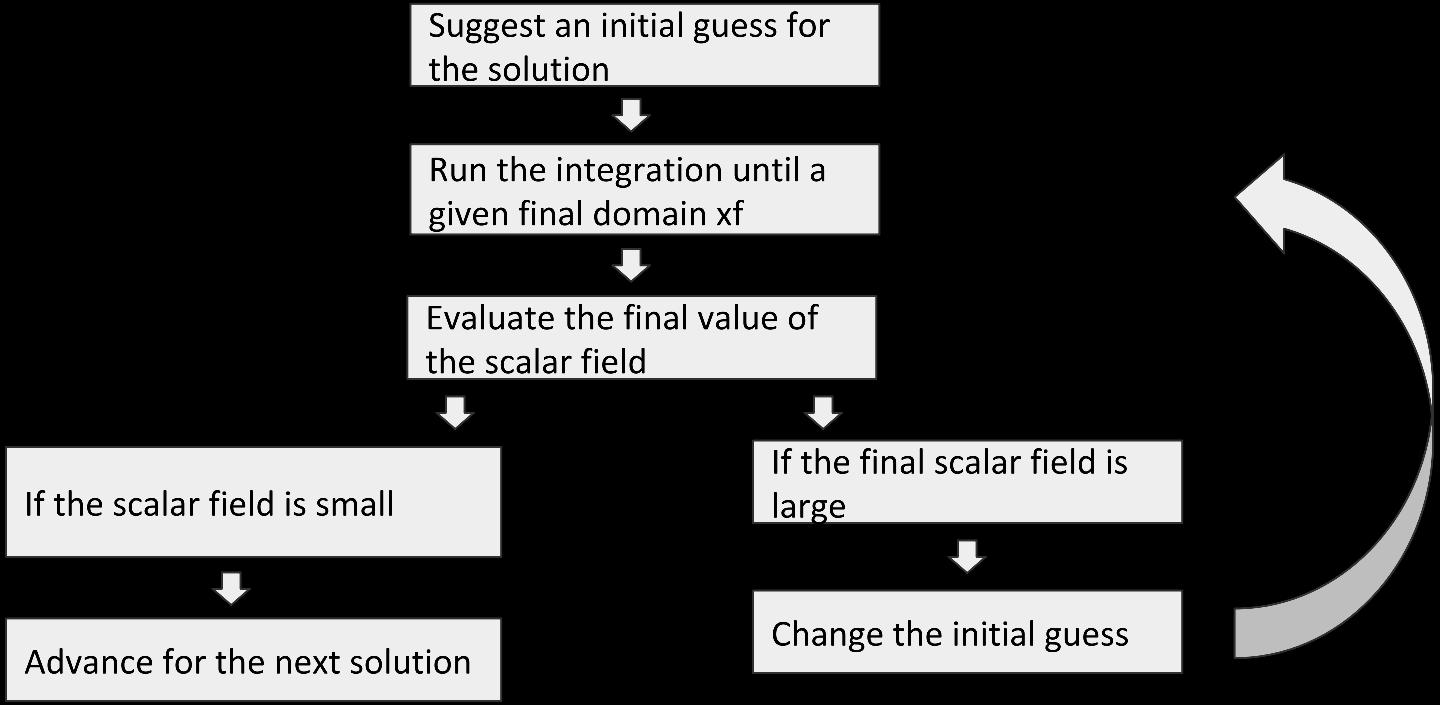
Evaluate the final value of the scalar field

If the scalar field is small

If the final scalar field is large

Change the initial guess

Numerical Scheme: Shooting



Numerical Procedure: Structure

function.h

Contains the ODE equations in the function format

Integrator.h

Contains the Dormand Prince method

Evaluates the Step error

Noether charge

Implements the initial conditions

BS.c

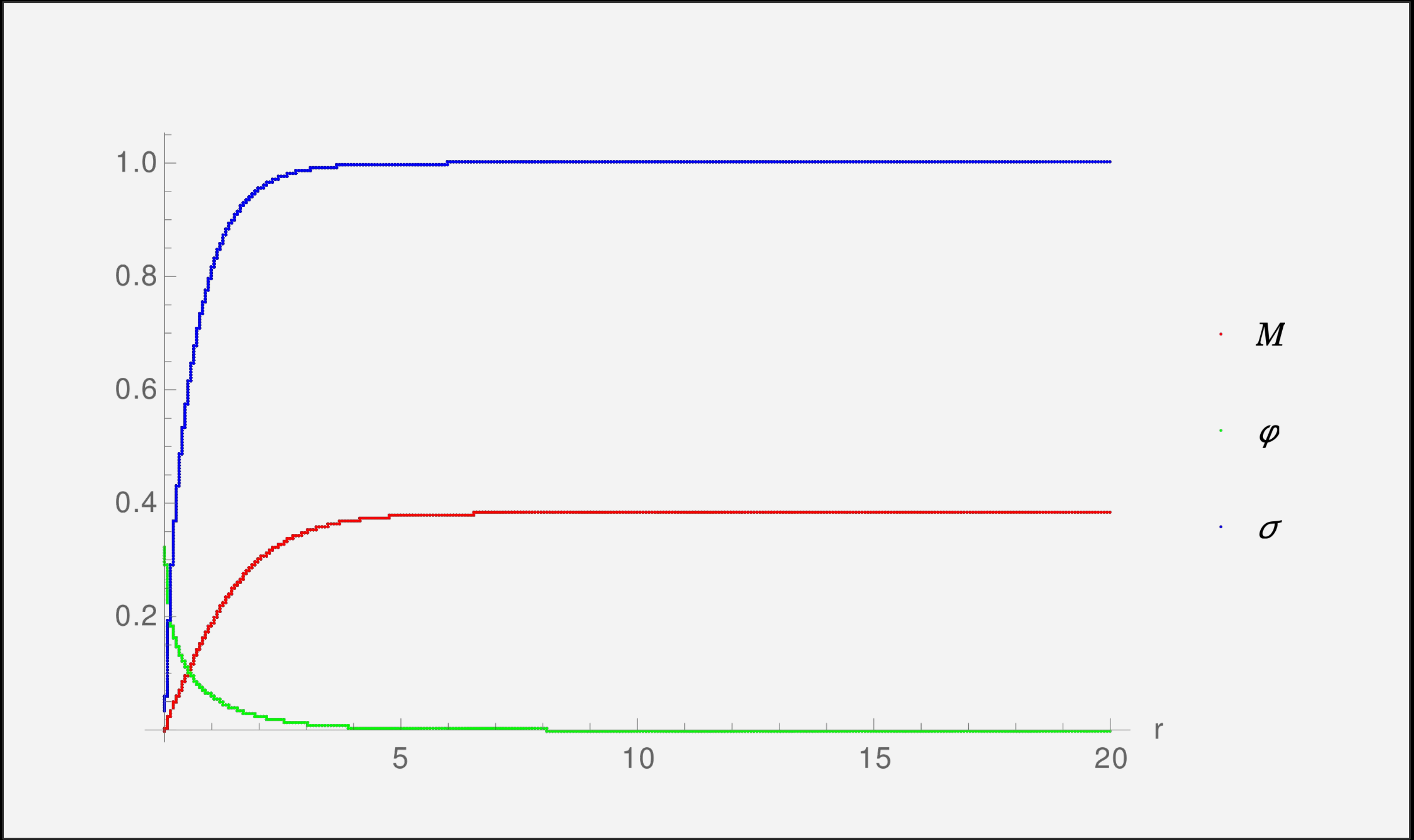
Compiles all the functions

Generates the solution or phase space.

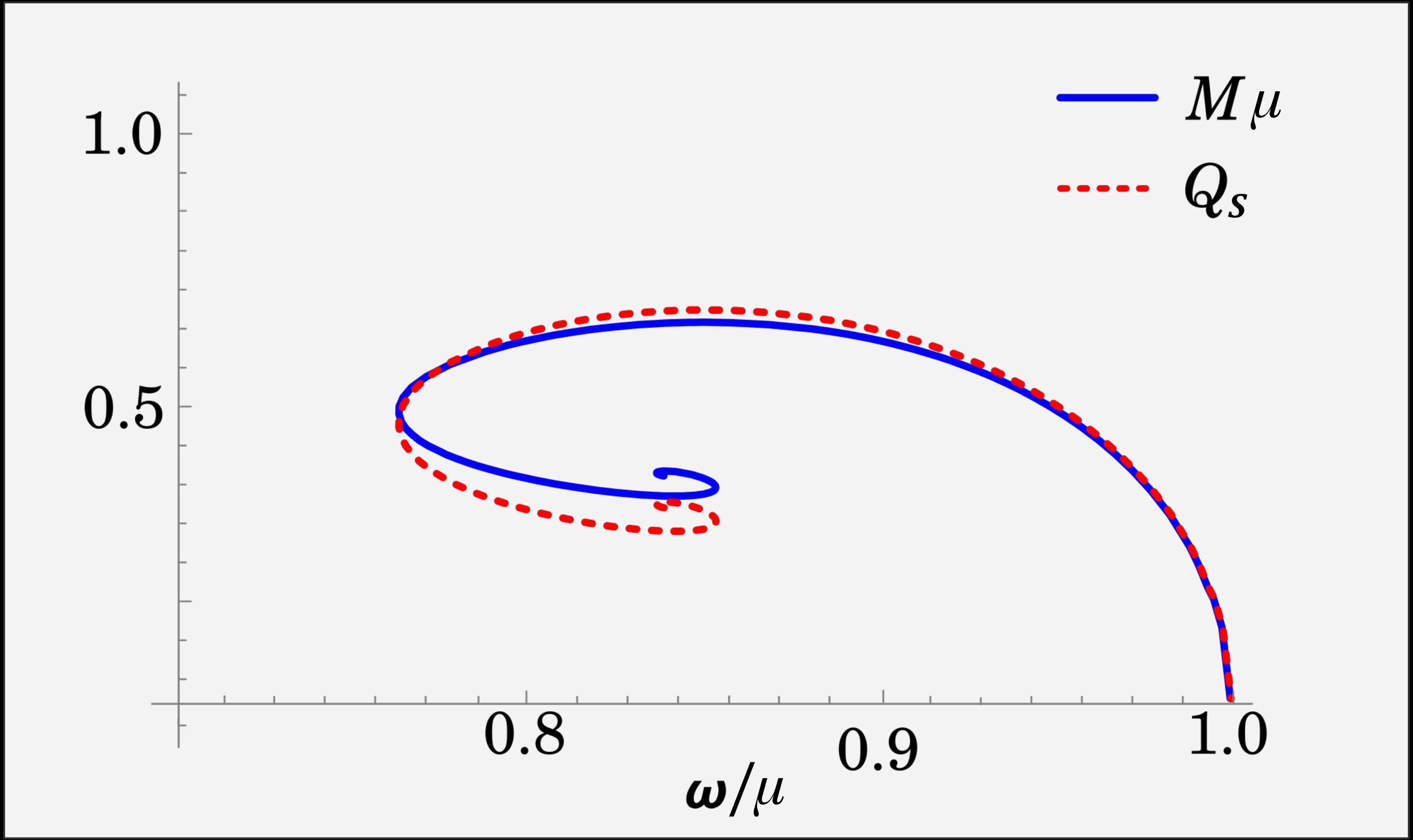
Contains a Secant strategy able to implement proper boundary conditions

Data Analysis

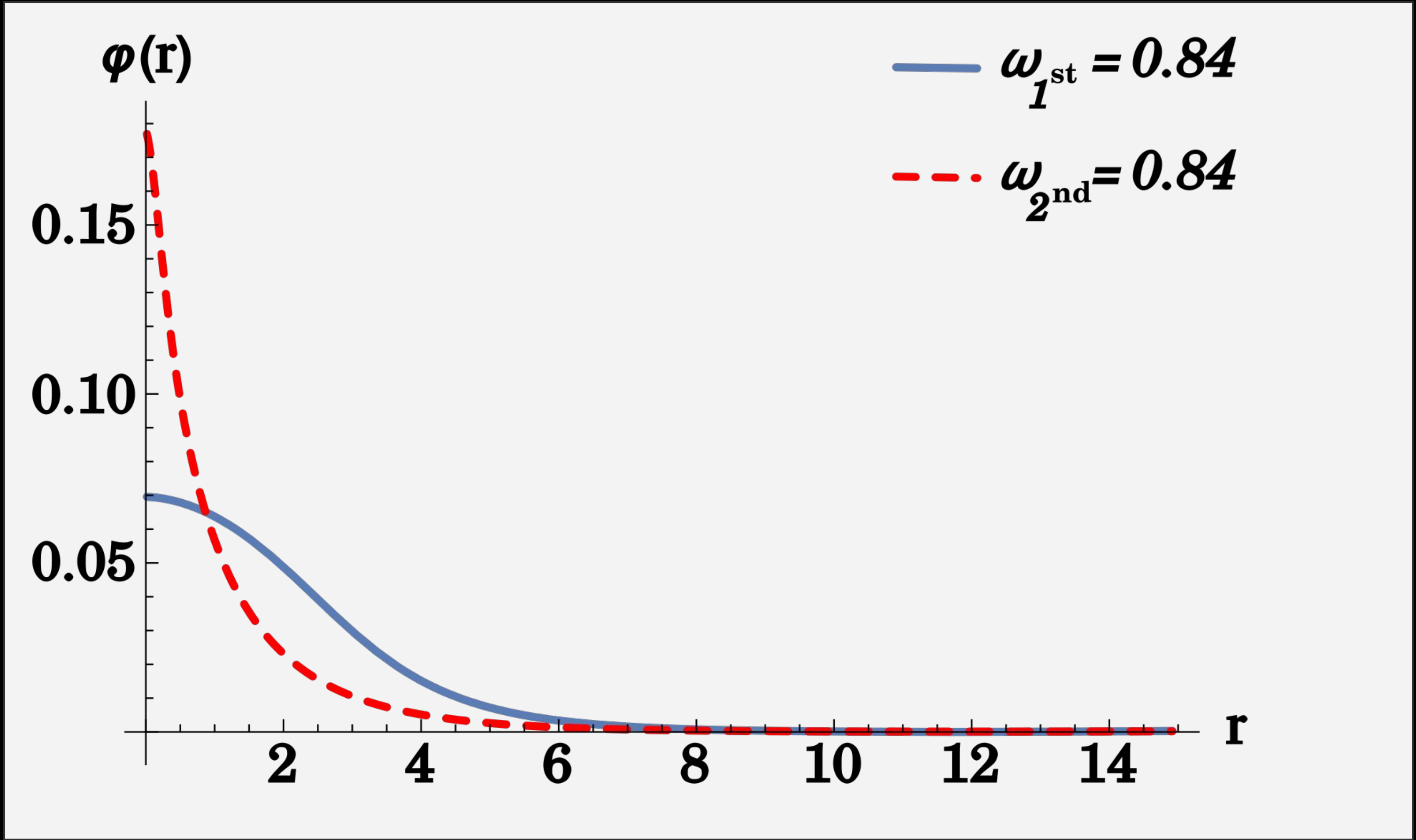
Numerical Procedure: Solution profile



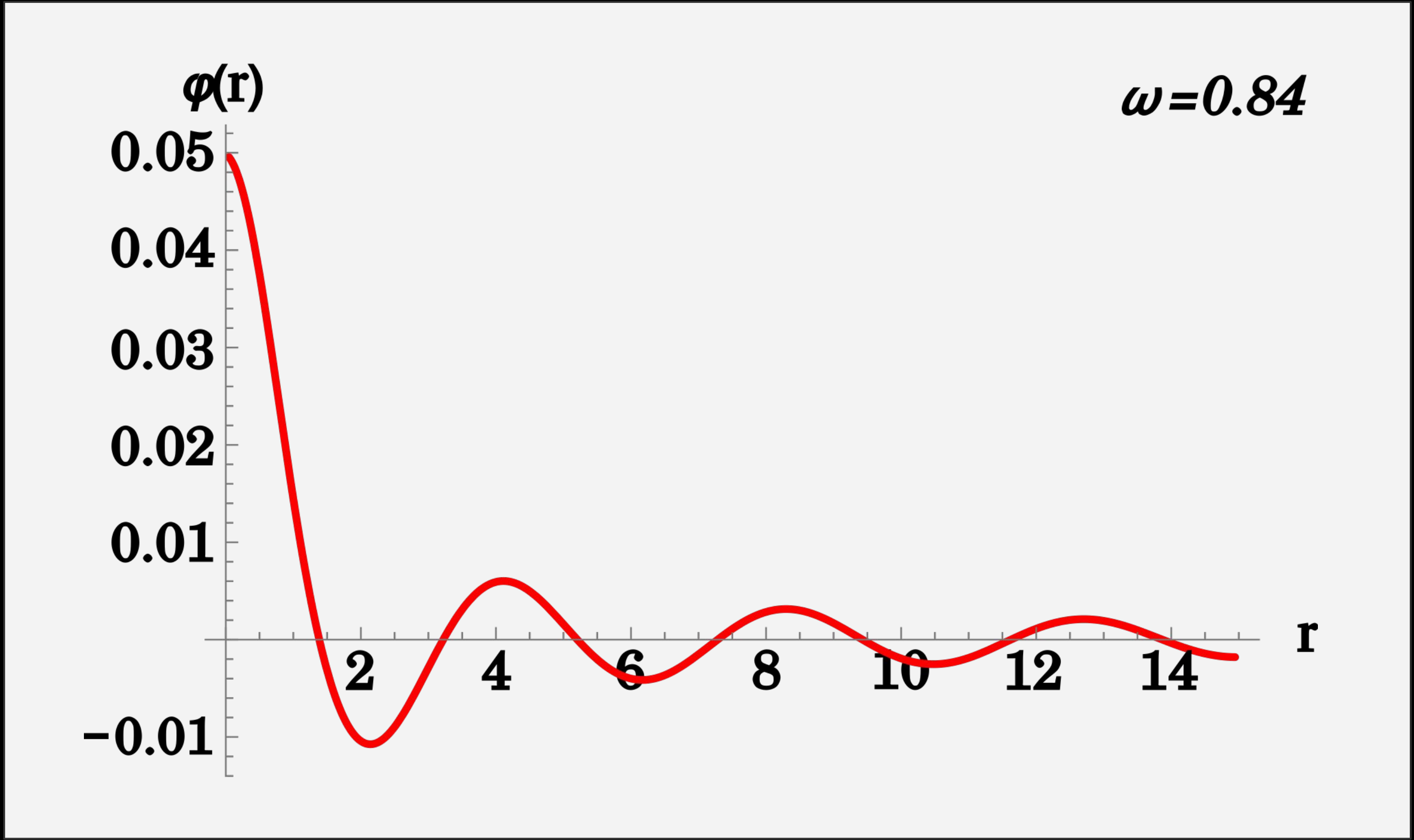
Numerical Procedure: Domain of existence

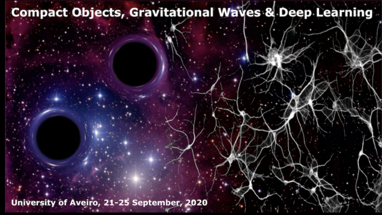


Numerical Procedure: Solution profile



Numerical Procedure: Excited States

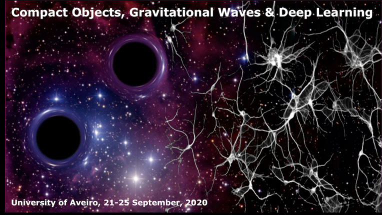




How we numerically solve ODEs: Boson Star

Thank you!
Obrigado!

pomboalexandremira@ua.pt



How we numerically solve ODEs: Boson Star

pomboalexandremira@ua.pt