Einstein Toolkit hackathon session

Michele Grasso

Inhomogeneous Cosmologies IV

July 18th 2019, Torun

What the ET is?

- “The Einstein Toolkit is a community-driven software platform of core computational tools to advance and support research in relativistic astrophysics and gravitational physics.”

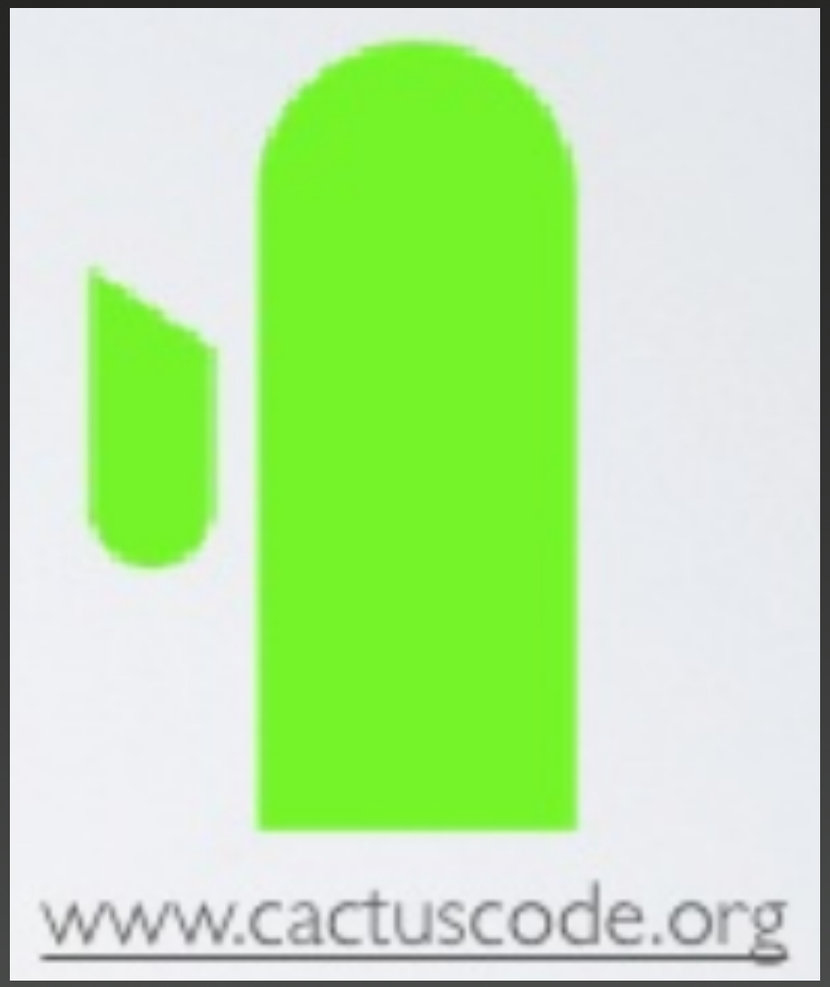
What the ET does?

- Mainly, solve Einstein and GRHD equations

- thus, it contains modules to:
    - solve Einstein and GRHD equations
    - set initial data and computational domain
    - do analysis
    - ...

Structure

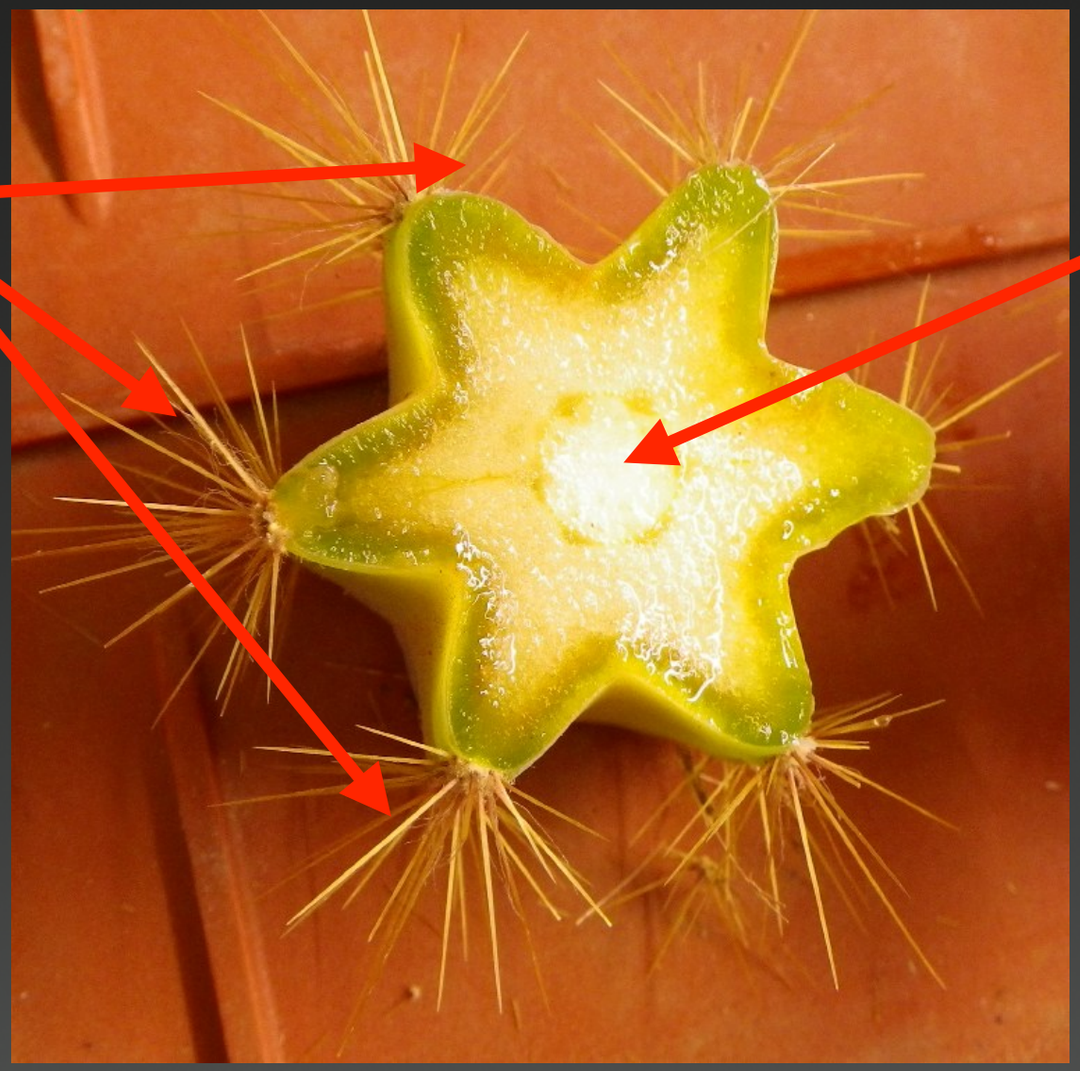
It is based on the **Cactus code:** why Cactus?



Structure

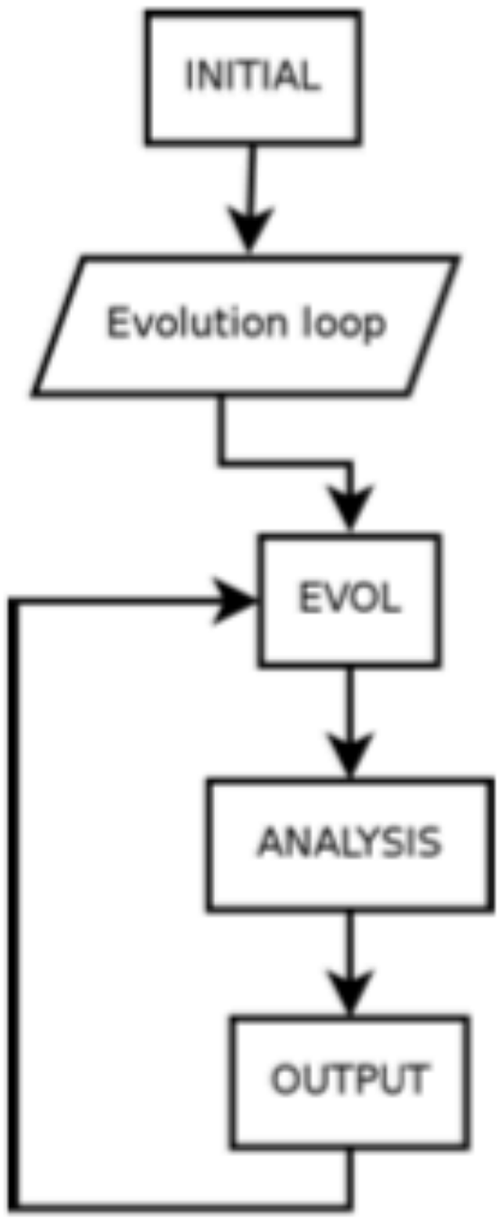
It is based on the **Cactus code:** why Cactus?

The thorns: separate modules to implement some functionalities. (C,C++)



The flesh: core part which acts as utility and service library

Rule-based schedule



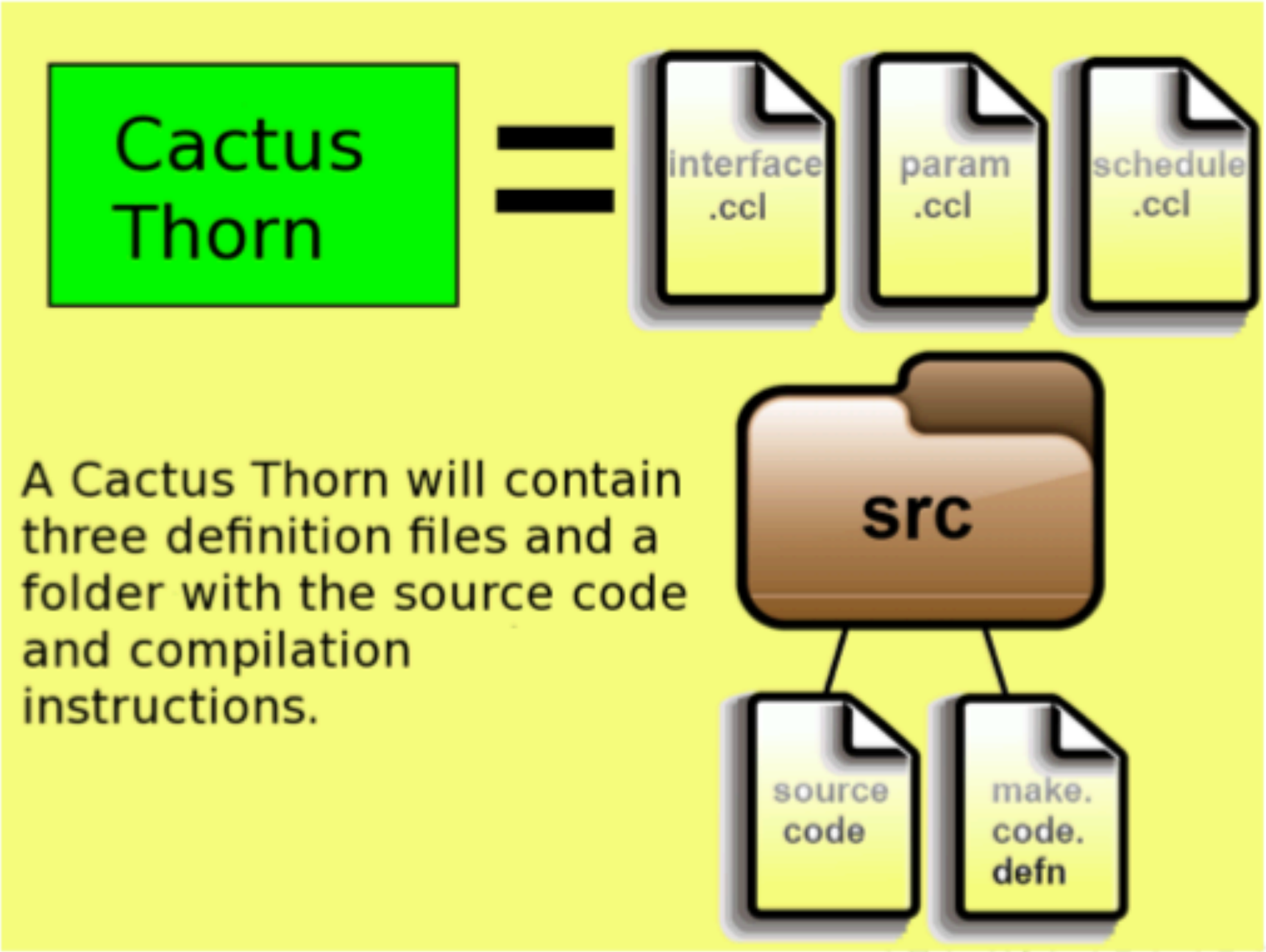
Basic schedule bins:

- STARTUP
- PARAMCHECK: check parameters consistency;
- INITIAL: set up initial data;
- CHECKPOINT: write simultaion checkpoint;
- RECOVER: recover from a checkpoint;
- PRESTEP, EVOL, POSTSTEP: evolution steps;
- ANALYSIS: periodic analysis and output;
- TERMINATE: clean-up phase.

https://docs.einsteintoolkit.org/et-docs/images/9/95/Cactusintro.pdf

What's inside a thorn?

Inside view of a plug-in module, or thorn for Cactus



https://docs.einsteintoolkit.org/et-docs/images/9/95/Cactusintro.pdf

What we will do during this session?

1. Install the ET
2. Run 1: inhomogeneous density + EdS
3. Run 2: inhomogeneous density + FLRW

…quicker version

Download

a. Create a new folder “ET” (optional)

b. curl -kLO https://raw.githubusercontent.com/gridaphobe/CRL/ET_2019_03/GetComponents

c. curl -kLO https://bitbucket.org/einsteintoolkit/manifest/raw/ET_2019_03/einsteintoolkit.th

d. now you have **GetComponents** and the thornlist **einsteintoolkit.th**

e. chmod a+x GetComponents

let's speed up!

a. Let's **modify the thornlist!**
    a. open einsteintoolkit.th with an editor
    b. find **eloisa/ctthorns.git**
    c. !REPO_BRANCH=$ET_RELEASE => **!REPO_BRANCH=master**
    d. search for CTThorns and add the line **CTThorns/CT_Dust** to the list
    e. comment out **Llama/WaveExtractL, EinsteinInitialData/NoExcision** and **EinsteinInitialData/Meudon_BH**

b. ./GetComponents einsteintoolkit.th

ET structure

Cactus

arrangements

configs: OptionList, RunScript, SubmitScript, ThornList, properties.ini

exe

repos

simfactory

par

bin, utils,…

Installing

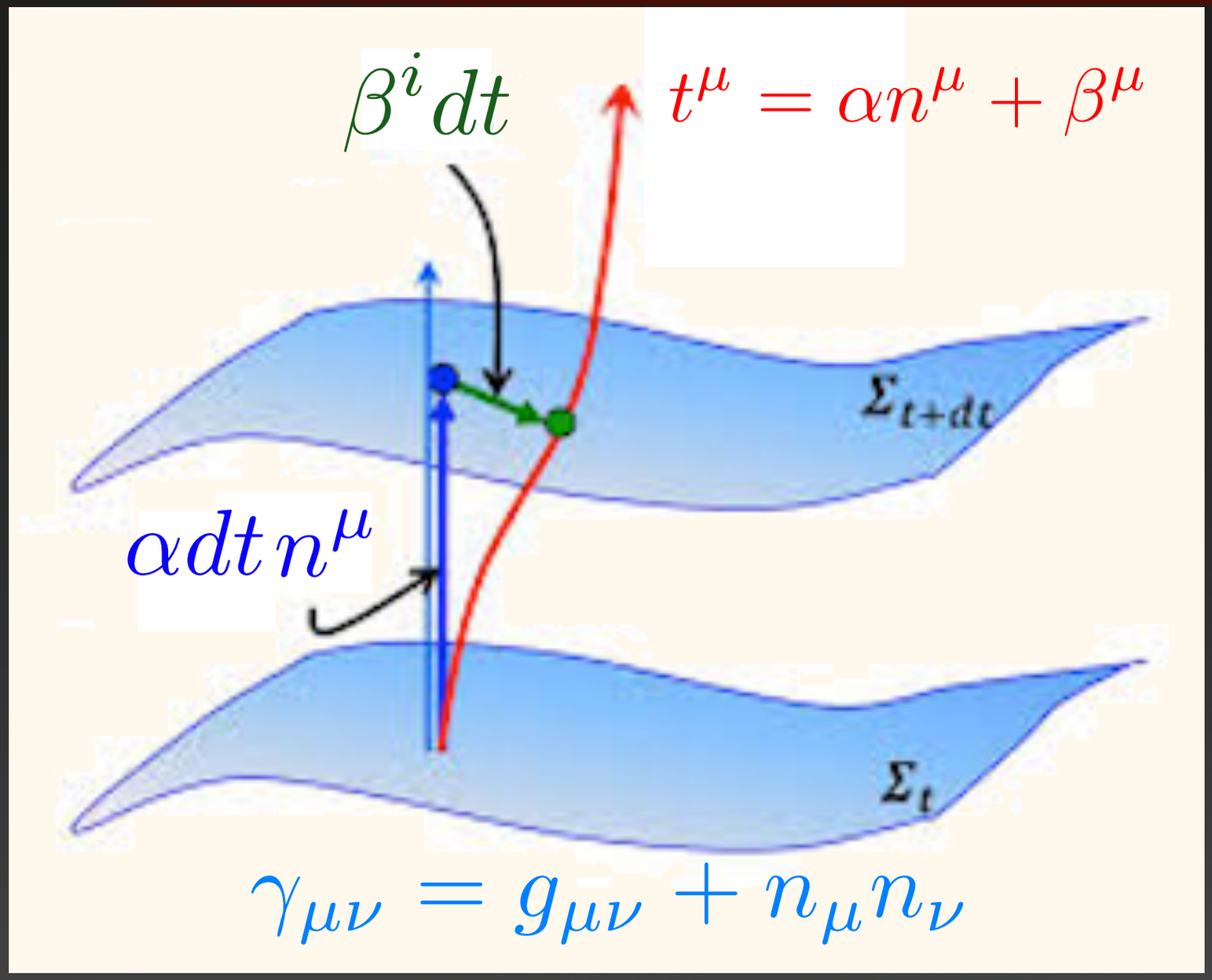
a. Now we have all the components into de folder **Cactus.** So, let's enter in Cactus using cd Cactus
b. Configure your machine: ./simfactory/bin/sim setup-silent generate a **file.ini** located in **simfactory/mdb/machines/**
c. The file.ini contains some general setups like nome, source folder, simulation folder, ppn, etc
d. Build the ET: ./simfactory/bin/sim build --mdbkey make 'make -**jn**' --thornlist=../einsteintoolkit.th

"n" in red, represent the number of cores of your machine!

Since we are waiting…

- We said that the ET mainly, solve Einstein and GRHD equations
- and contains modules to:
    - solve Einstein and GRHD equations
    - set initial data and computational domain
    - do analysis
    - …

Einstein equations? How?



$$(\mathcal{M}, g_{\mu\nu})$$

$$(\alpha, \beta^i, \gamma_{ij}, K_{ij})$$

$$ds^2 = -\alpha^2 dt^2 + \gamma_{ij}\left(dx^i \beta^i dt\right)\left(dx^j \beta^j dt\right)$$

ADM

Einstein equation

$$R_{\mu\nu} - \frac{1}{2} g_{\mu\nu} R = 8\pi T_{\mu\nu}$$

ADM

ADM

Evolution equations

$$\partial_t \gamma_{ij} = -2\alpha K_{ij} + D_i \beta_j + D_j \beta_i$$

$$\partial_t K_{ij} = -D_i D_j \alpha + \alpha(R_{ij} - 2K_{il}K^l_j + K^l_l K_{ij}) - 8\pi\alpha(S_{ij} - \frac{1}{2}\gamma_{ij}(S-\rho)) + \beta^l D_l K_{ij} + K_{il} D_j \beta^l + K_{jl} D_i \beta^l$$

ADM

Evolution equations

$$\partial_t \gamma_{ij} = -2\alpha K_{ij} + D_i \beta_j + D_j \beta_i$$

$$\partial_t K_{ij} = -D_i D_j \alpha + \alpha(R_{ij} - 2K_{il}K^l_j + K^l_l K_{ij}) - 8\pi\alpha(S_{ij} - \frac{1}{2}\gamma_{ij}(S-\rho)) + \beta^l D_l K_{ij} + K_{il} D_j \beta^l + K_{jl} D_i \beta^l$$

$$+$$

ADM

Evolution equations

$$\partial_t \gamma_{ij} = -2\alpha K_{ij} + D_i \beta_j + D_j \beta_i$$

$$\partial_t K_{ij} = -D_i D_j \alpha + \alpha(R_{ij} - 2K_{il}K^l_j + K^l_l K_{ij}) - 8\pi\alpha(S_{ij} - \frac{1}{2}\gamma_{ij}(S-\rho)) + \beta^l D_l K_{ij} + K_{il} D_j \beta^l + K_{jl} D_i \beta^l$$

+

Constraint equations

$$R + K^2 - K_{ij}K^{ij} = 16\pi\rho$$

$$D_j(K^{ij} - \gamma^{ij}K) = 8\pi S^i$$

ADM

Evolution equations

EinsteinBasic, EinsteinEvolve, McLachlan, ...

+

Constraint equations

$$R + K^2 - K_{ij}K^{ij} = 16\pi\rho$$

$$D_j(K^{ij} - \gamma^{ij}K) = 8\pi S^i$$

ADM

Evolution equations

EinsteinBasic, EinsteinEvolve, McLachlan, ...

+

Constraint equations

EinsteinInitialData, EinsteinEOS, CTThorns,..

ADM

Evolution equations

EinsteinBasic, EinsteinEvolve, McLachlan, ...

+

Constraint equations

EinsteinInitialData, EinsteinEOS, CTThorns,..

and this is almost the truth (BSSN!)...

Running: EdS using CT_Dust thorn

a. Finally, we are ready to start a simulation, but first copy the parfile into the folder **par** cp repos/ctthorns/par/eds.par par/.

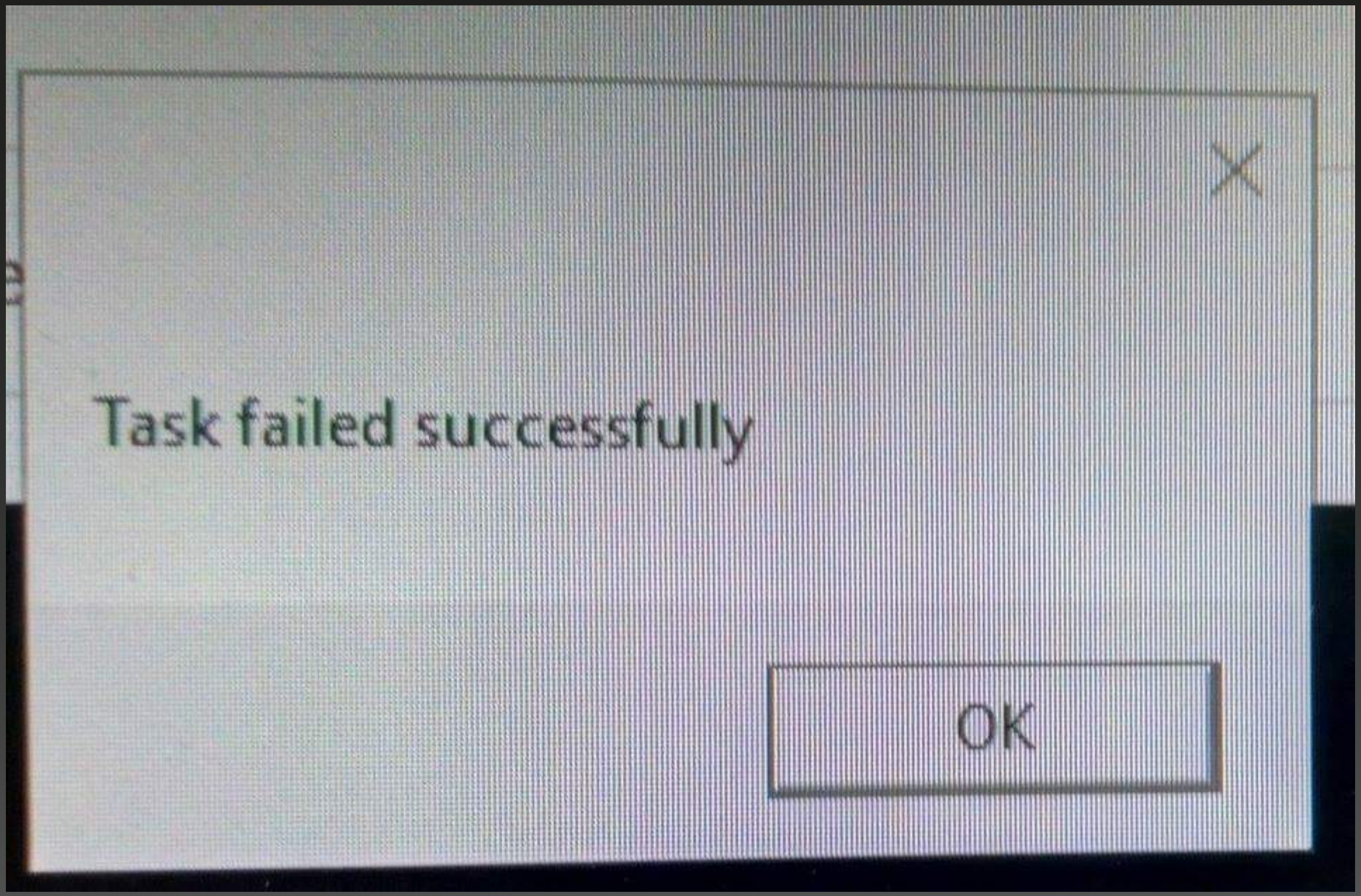
b. ./simfactory/bin/sim create-submit run001 --parfile par/eds.par --walltime 0:30:0 this will start a simulation called **run001** using the informations stored in **eds.par** which last for **30 min**

c. the simulation status can be checked typing:

./simfactory/bin/sim list-simulations run001 returns the status (running, queue, finished)

./simfactory/bin/sim show-output --follow run001 shows the output

Show the results



FLRWSolver

a. git clone https://bitbucket.org/hayleymac/flrwsolver-public-2.git

b. cp -r flrwsolver-public-2 Cactus/repos/einsteininitialdata/FLRWSolver

c. cd Cactus/arrangements/EinsteinInitialData/

d. ln -s repos/einsteininitialdata/FLRWSolver **.**

e. edit **../einsteintoolkit.th**:

(i) find EinsteinInitialData and add the line EinsteinInitialData/FLRWSolve

(ii) comment out Meudon_BH if needed

f. Build the exe: ./simfactory/bin/sim build --mdbkey make 'make -**jn**' --thornlist=../einsteintoolkit.th

Since we are waiting…

Questions?

https://docs.einsteintoolkit.org/et-docs/images/9/95/Cactusintro.pdf

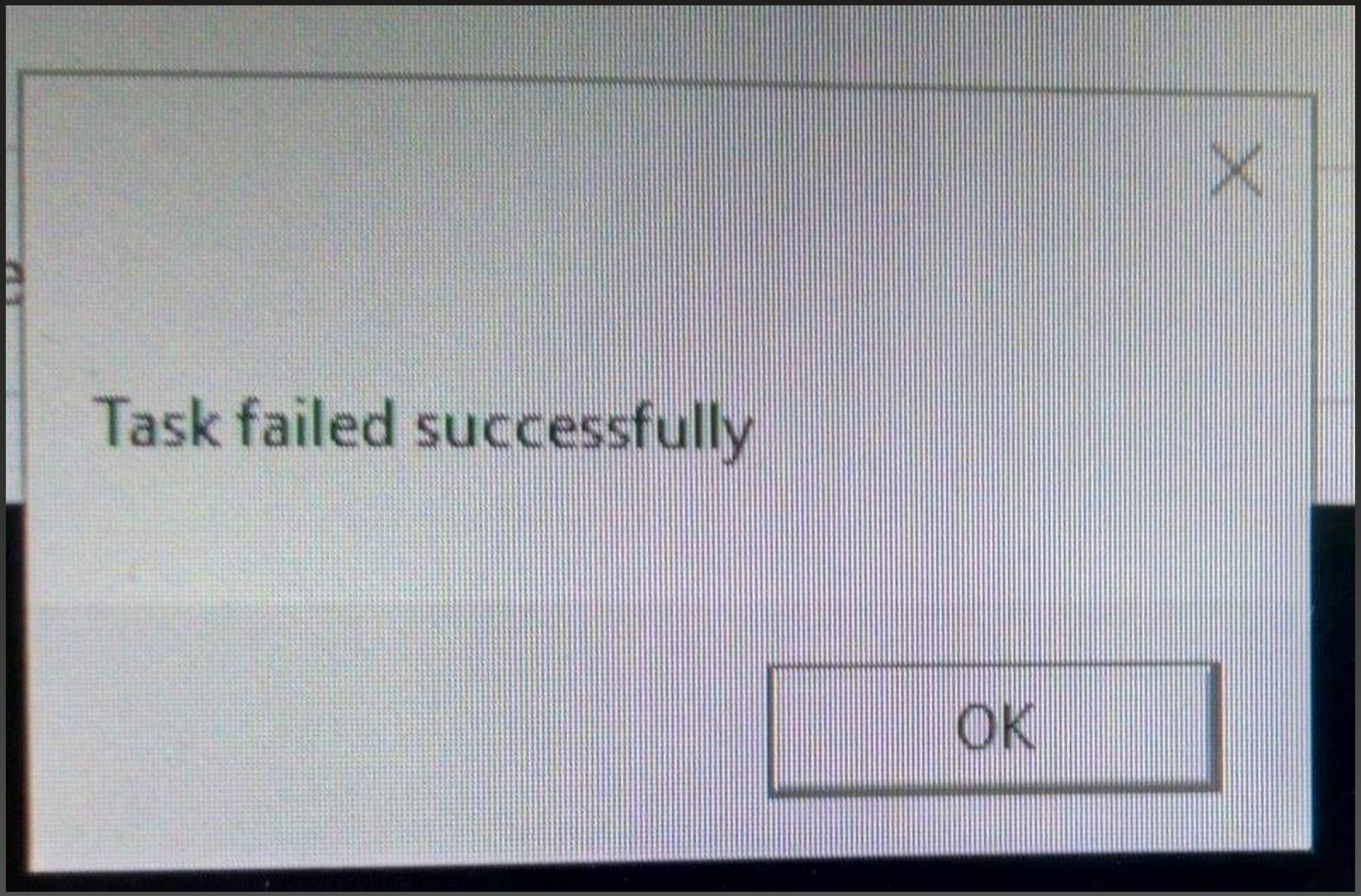
Running: FLRW using FLRWSolve thorn

a. Copy the parfile into the folder **par** cp arrangements/EinsteinInitialData/FLRWSolver/FLRW.par par/.

b. ./cactus_simflrw create-submit FLRW001 --parfile par/FLRW.par --walltime 0:30:0 this will start a simulation called **FLRW001** using the informations stored in **FLRW.par** which last for **30 min**

c. the simulation status can be checked typing:

   i) ./simfactory/bin/sim list-simulations FLRW001 returns the status (running, queue, finished)

   ii) ./simfactory/bin/sim show-output --follow FLRW001 shows the output

Show the results