Title: OGRePy and Time Travel Paradoxes

Speakers: Barak Shoshany

Collection/Series: Cosmology and Gravitation

Subject: Cosmology

Date: February 04, 2025 - 11:00 AM

URL: https://pirsa.org/25020024

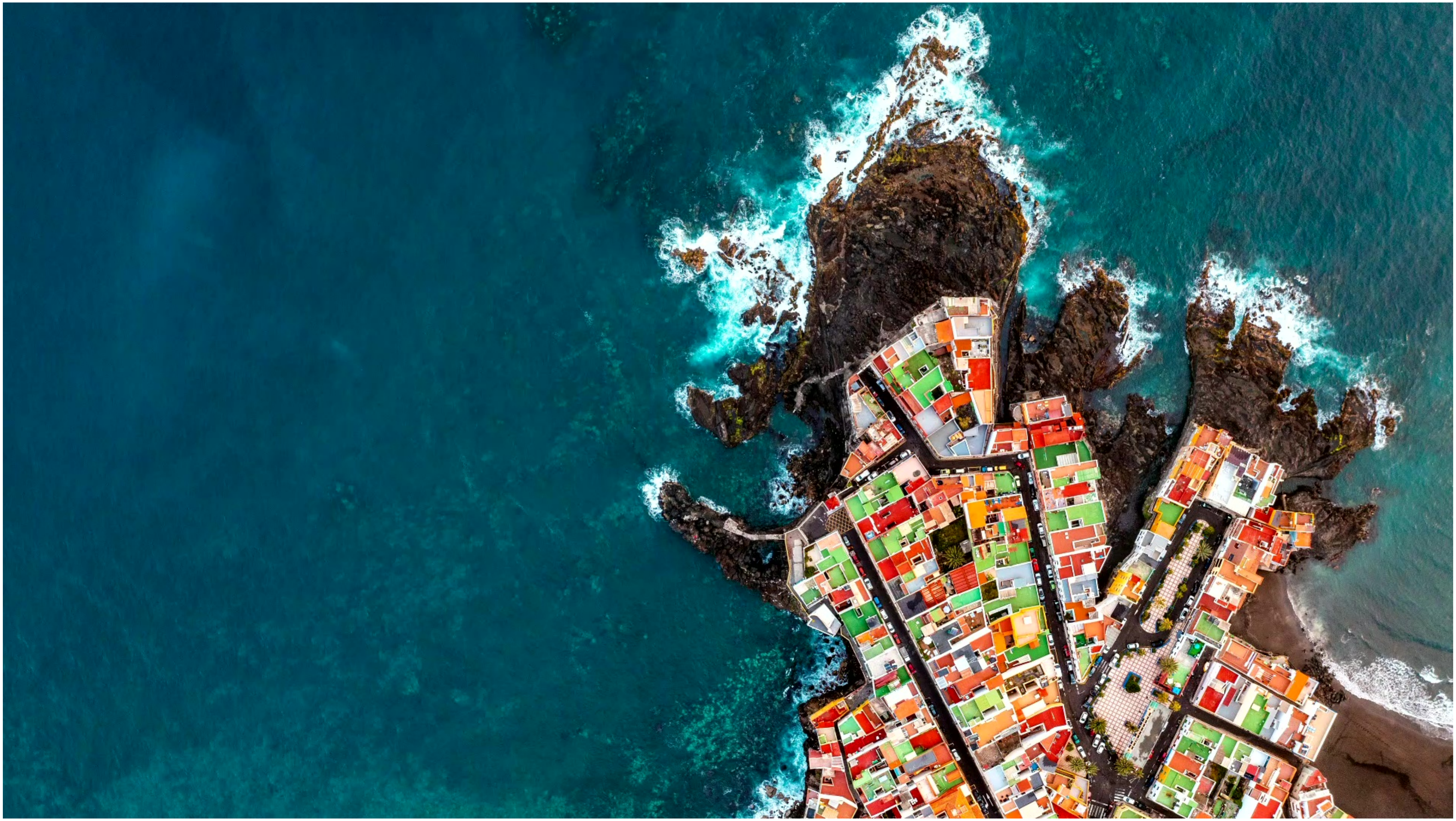
Abstract:

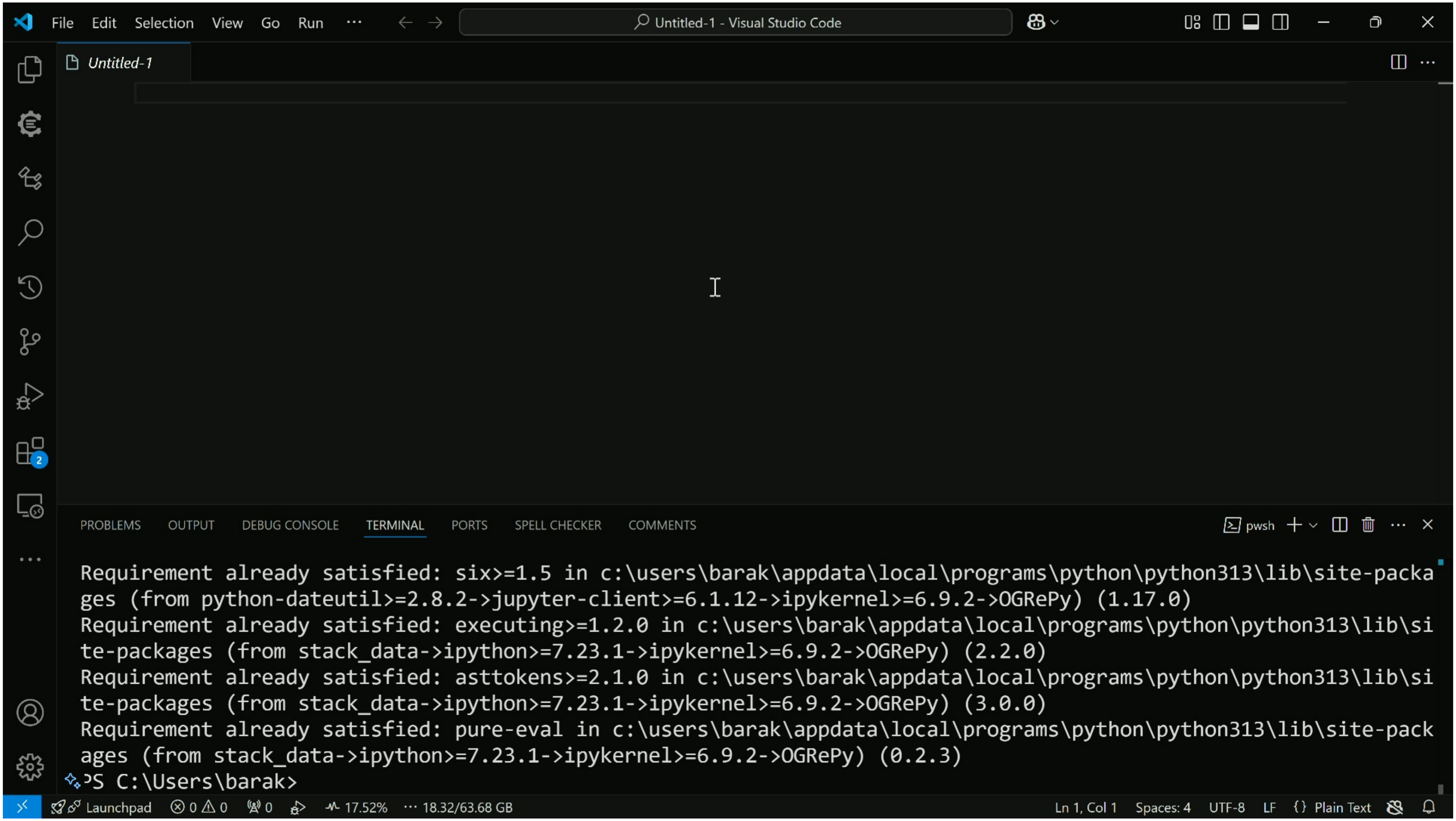
Part I: OGRePy: Object-Oriented General Relativity in Python

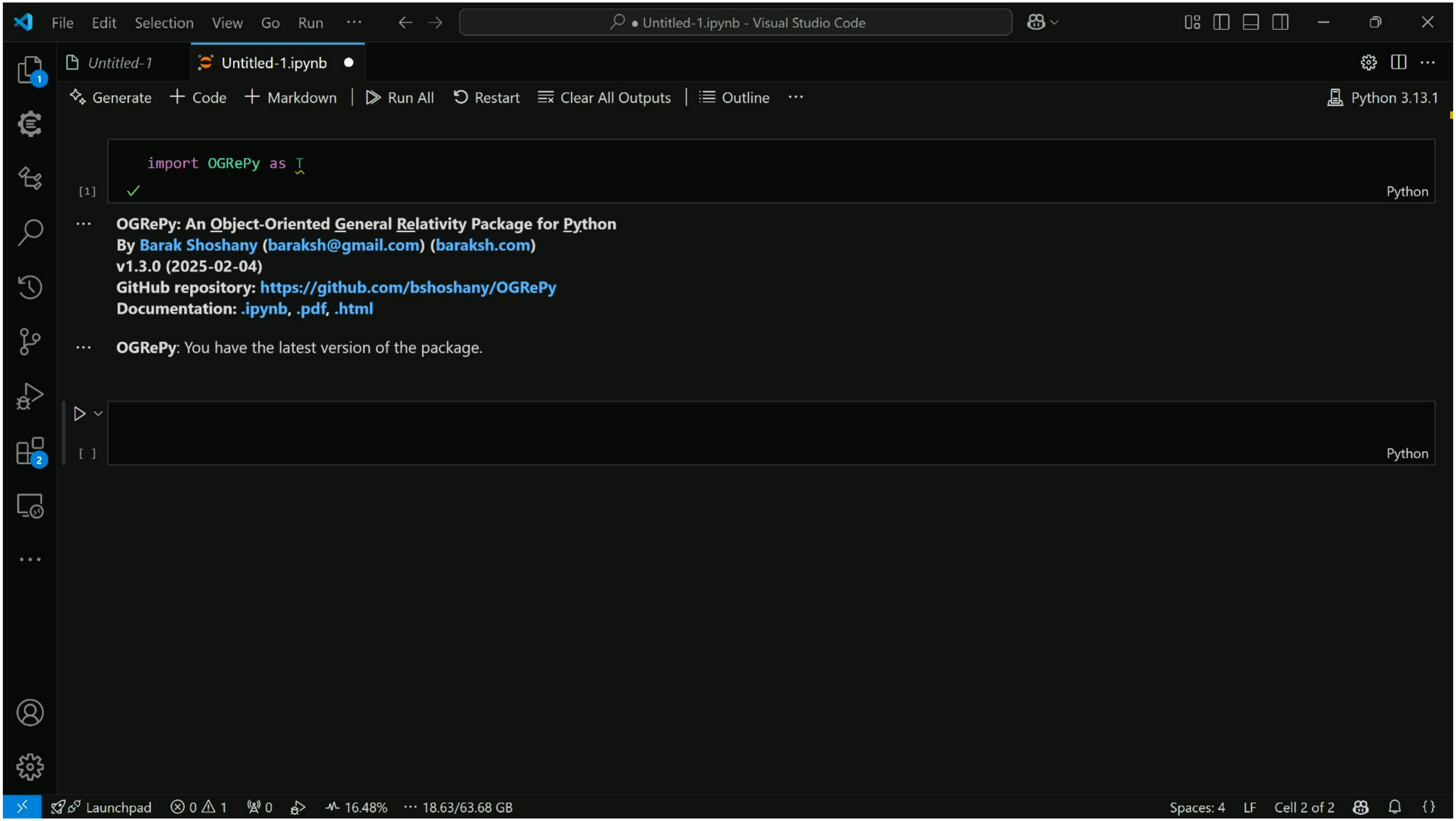
I will present a detailed introduction to my new Python package, OGRePy: (O)bject-Oriented (G)eneral (Re)lativity for (Py)thon, a port of my popular Mathematica package OGRe, which is used by many researchers in general relativity and related areas. I will demonstrate the package's usage and features, including its ability to manipulate tensors of arbitrary rank using an intuitive interface, and calculate arbitrary tensor formulas involving any combination of addition, multiplication, trace, contraction, and partial and covariant derivatives - while automatically figuring out the proper index configuration and coordinate system to use for each tensor, eliminating user error.

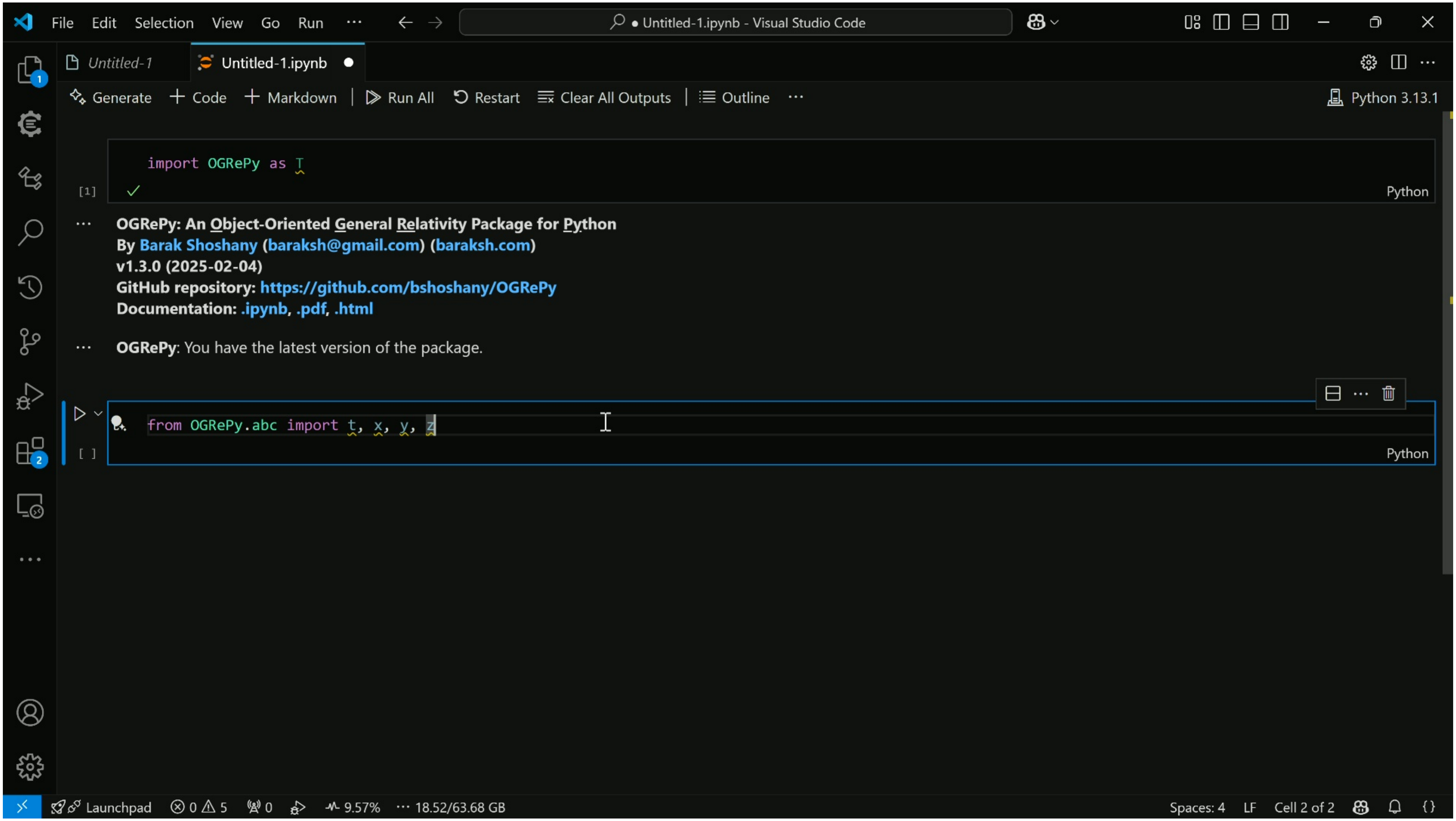
Part II: Time Travel Paradoxes and Entangled Timelines

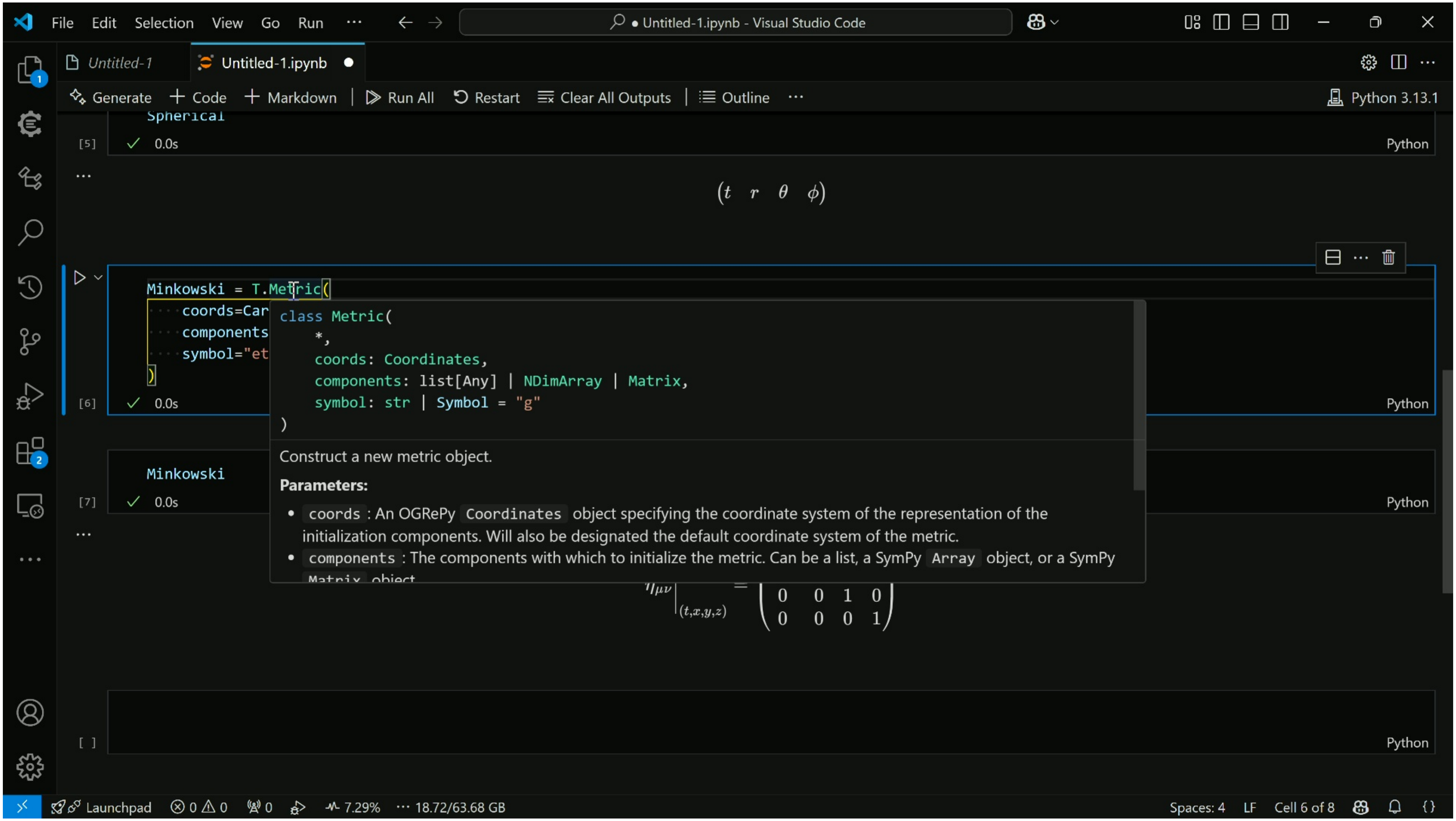
If time travel is possible, it seems that it would inevitably lead to paradoxes, indicating an internal inconsistency in our current theories of nature. Can these paradoxes be resolved by new laws of physics, or perhaps even existing ones? I will first review the different types of time travel paradoxes and their proposed resolutions. Then I will present the results of my 3 recent papers (1911.11590, 2110.02448, 2303.07635) discussing different aspects of time travel paradoxes from the perspectives of both general relativity and quantum mechanics. I will argue that generic time travel paradoxes can only be resolved using the concept of parallel timelines, and suggest possibilities for how such timelines may manifest themselves.




File Edit Selection View Go Run ···
Untitled-1 - Visual Studio Code
Untitled-1
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SPELL CHECKER COMMENTS
pwsh
Requirement already satisfied: six>=1.5 in c:\users\barak\appdata\local\programs\python\python313\lib\site-packages (from python-dateutil>=2.8.2->jupyter-client>=6.1.12->ipykernel>=6.9.2->OGRePy) (1.17.0)
Requirement already satisfied: executing>=1.2.0 in c:\users\barak\appdata\local\programs\python\python313\lib\site-packages (from stack_data->ipython>=7.23.1->ipykernel>=6.9.2->OGRePy) (2.2.0)
Requirement already satisfied: asttokens>=2.1.0 in c:\users\barak\appdata\local\programs\python\python313\lib\site-packages (from stack_data->ipython>=7.23.1->ipykernel>=6.9.2->OGRePy) (3.0.0)
Requirement already satisfied: pure-eval in c:\users\barak\appdata\local\programs\python\python313\lib\site-packages (from stack_data->ipython>=7.23.1->ipykernel>=6.9.2->OGRePy) (0.2.3)
PS C:\Users\barak>
Launchpad
17.52%
18.32/63.68 GB
Ln 1, Col 1 Spaces: 4 UTF-8 LF {} Plain Text


File Edit Selection View Go Run ···
Untitled-1.ipynb - Visual Studio Code
Untitled-1
Untitled-1.ipynb
Generate
+ Code
+ Markdown
Run All
Restart
Clear All Outputs
Outline
Python 3.13.1
import OGRePy as T
[1]
Python
OGRePy: An Object-Oriented General Relativity Package for Python
By Barak Shoshany (baraksh@gmail.com) (baraksh.com)
v1.3.0 (2025-02-04)
GitHub repository: https://github.com/bshoshany/OGRePy
Documentation: .ipynb, .pdf, .html
OGRePy: You have the latest version of the package.
[ ]
Python
Launchpad
16.48%
18.63/63.68 GB
Spaces: 4
LF
Cell 2 of 2


File Edit Selection View Go Run ···
● Untitled-1.ipynb - Visual Studio Code
Untitled-1
Untitled-1.ipynb
Generate
Code
Markdown
Run All
Restart
Clear All Outputs
Outline
Python 3.13.1
import OGRePy as T
[1]
Python
OGRePy: An Object-Oriented General Relativity Package for Python
By Barak Shoshany (baraksh@gmail.com) (baraksh.com)
v1.3.0 (2025-02-04)
GitHub repository: https://github.com/bshoshany/OGRePy
Documentation: .ipynb, .pdf, .html
OGRePy: You have the latest version of the package.
from OGRePy.abc import t, x, y, z
[ ]
Python
Launchpad
0 5
0
9.57%
18.52/63.68 GB
Spaces: 4
LF
Cell 2 of 2


File Edit Selection View Go Run ···
● Untitled-1.ipynb - Visual Studio Code
Untitled-1
Untitled-1.ipynb
Generate
Code
Markdown
Run All
Restart
Clear All Outputs
Outline
Python 3.13.1
Spherical
[5]
0.0s
Python
(t r θ ϕ)
Minkowski = T.Metric(
coords=Car
components
symbol="et
)
[6]
0.0s
Python
class Metric(
*,
coords: Coordinates,
components: list[Any] | NDimArray | Matrix,
symbol: str | Symbol = "g"
)
Construct a new metric object.
Parameters:
coords : An OGRePy Coordinates object specifying the coordinate system of the representation of the initialization components. Will also be designated the default coordinate system of the metric.
components : The components with which to initialize the metric. Can be a list, a SymPy Array object, or a SymPy
Minkowski
[7]
0.0s
Python
[ ]
Python
Launchpad
7.29%
18.72/63.68 GB
Spaces: 4
LF
Cell 6 of 8

```python
from OGRePy.abc import M

Schwarzschild = T.Metric(
    coords=Spherical,
    components=T.diag(
        -(1 - 2 * M / r),
        1 / (1 - 2 * M / r),
        r**2,
        r**2 * T.s.sin(theta) ** 2,
    ),
)
```

[8] ✓ 0.0s

```python
Schwarzschild
```

[9] ✓ 0.0s

$$g_{\mu\nu}\Big|_{(t,r,\theta,\phi)} = \begin{pmatrix} \frac{2M}{r} - 1 & 0 & 0 & 0 \\ 0 & \frac{1}{-\frac{2M}{r}+1} & 0 & 0 \\ 0 & 0 & r^2 & 0 \\ 0 & 0 & 0 & r^2 \sin^2(\theta) \end{pmatrix}$$

File Edit Selection View Go Run ··· ● Untitled-1.ipynb - Visual Studio Code

Untitled-1 | Untitled-1.ipynb ●

Generate + Code + Markdown | Run All | Restart | Clear All Outputs | Outline ··· Python 3.13.1

```python
Schwarzschild.line_element()
```

[13] ✓ 0.0s Python

$$d\phi^2 r^2 \sin^2(\theta) + d\theta^2 r^2 + \frac{dr^2}{-\frac{2M}{r} + 1} + dt^2\left(\frac{2M}{r} - 1\right)$$

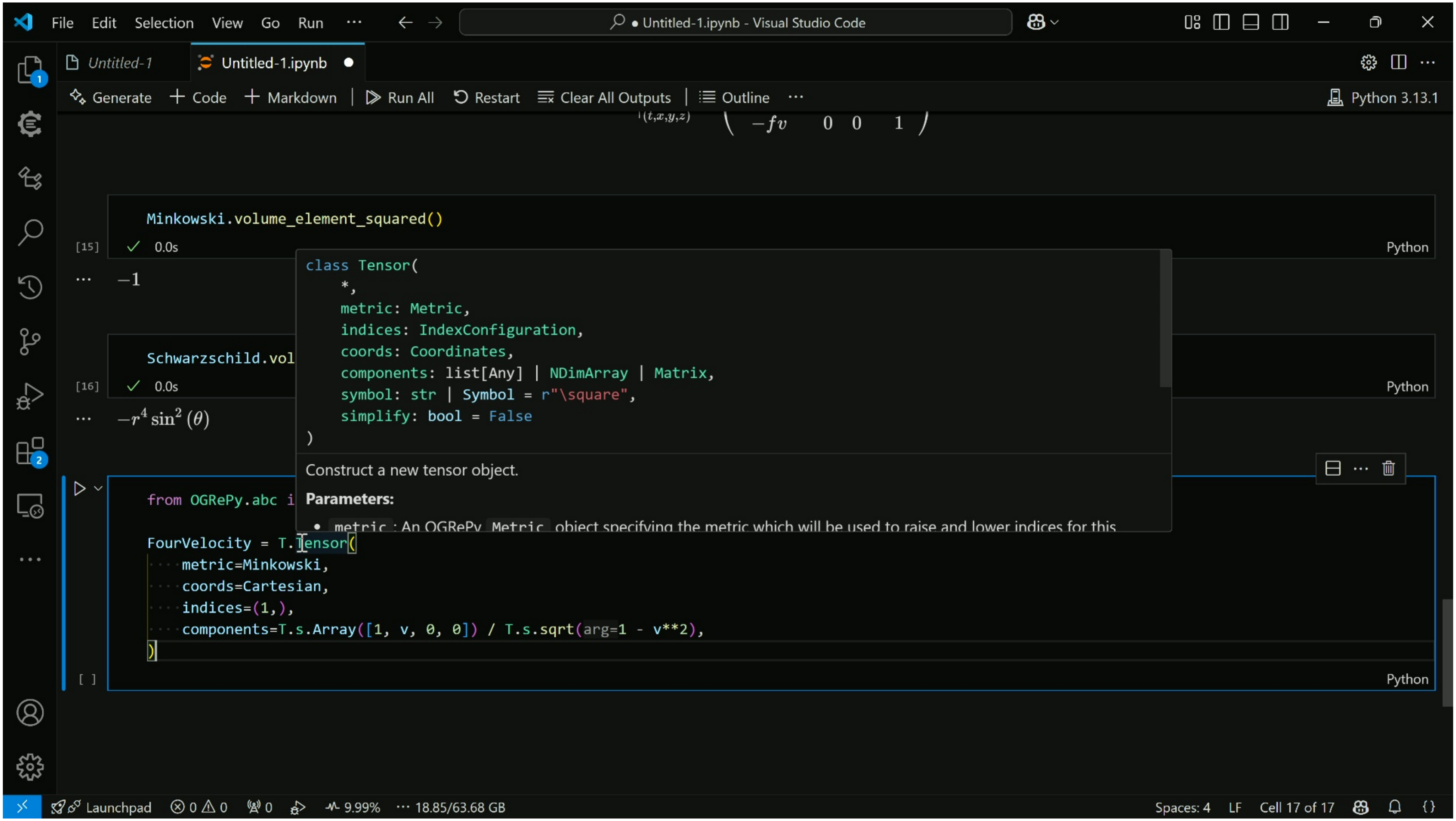
```python
v_t: Any = T.func(symbol="v")(t)
f_t_x_y_z: Any = T.func(symbol="f")(t, x, y, z)
Alcubierre = T.Metric(
    coords=Cartesian,
    components=[
        [-1 + f_t_x_y_z**2 * v_t**2, 0, 0, -f_t_x_y_z * v_t],
        [0, 1, 0, 0],
        [0, 0, 1, 0],
        [-f_t_x_y_z * v_t, 0, 0, 1],
    ],
)
```

[14] ✓ 0.2s Python

$$g_{\mu\nu}\Big|_{(t,x,y,z)} = \begin{pmatrix} f^2v^2 - 1 & 0 & 0 & -fv \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -fv & 0 & 0 & 1 \end{pmatrix}$$

[ ] Python

Launchpad 0 0 0 7.90% 18.82/63.68 GB Spaces: 4 LF Cell 14 of 15


File Edit Selection View Go Run ···
● Untitled-1.ipynb - Visual Studio Code
Untitled-1
Untitled-1.ipynb ●
Generate + Code + Markdown Run All Restart Clear All Outputs Outline ···
Python 3.13.1
Minkowski.volume_element_squared()
[15] ✓ 0.0s
Python
−1
Schwarzschild.vol
[16] ✓ 0.0s
Python
$-r^4 \sin^2(\theta)$
class Tensor(
*,
metric: Metric,
indices: IndexConfiguration,
coords: Coordinates,
components: list[Any] | NDimArray | Matrix,
symbol: str | Symbol = r"\square",
simplify: bool = False
)
Construct a new tensor object.
Parameters:
from OGRePy.abc i
FourVelocity = T.Tensor(
metric=Minkowski,
coords=Cartesian,
indices=(1,),
components=T.s.Array([1, v, 0, 0]) / T.s.sqrt(arg=1 - v**2),
)
[ ]
Python
Launchpad 0 0 0 9.99% 18.85/63.68 GB
Spaces: 4 LF Cell 17 of 17

```python
    components=T.s.Array([1, v, 0, 0]) / T.s.sqrt(arg=1 - v**2),
    symbol="v",
)
```

[19] ✓ 0.0s Python

$$v^{\mu}\Big|_{(t,x,y,z)} = \begin{pmatrix} \frac{1}{\sqrt{1-v^2}} \\ \frac{v}{\sqrt{1-v^2}} \\ 0 \\ 0 \end{pmatrix}$$

```python
from OGRePy.abc import p, rho

PerfectFluid = T.Tensor(
    metric=Minkowski,
    coords=Cartesian,
    indices=(1, 1),
    components=T.diag(rho, p, p, p),
    symbol="T",
)
```

[18] ✓ 0.0s Python

$$T^{\mu\nu}\Big|_{(t,x,y,z)} = \begin{pmatrix} \rho & 0 & 0 & 0 \\ 0 & p & 0 & 0 \\ 0 & 0 & p & 0 \\ 0 & 0 & 0 & p \end{pmatrix}$$

File Edit Selection View Go Run ··· Untitled-1.ipynb - Visual Studio Code

Untitled-1 | Untitled-1.ipynb

Generate + Code + Markdown | Run All | Restart | Clear All Outputs | Outline ··· Python 3.13.1

```python
Schwarzschild.list(indices=(1, -1))
```

[23] ✓ 0.0s Python

$$g^{t}{}_{t} = g^{r}{}_{r} = g^{\theta}{}_{\theta} = g^{\phi}{}_{\phi} = 1$$

```python
Cartesian.set_coord_transformation(
    target=Spherical,
    rules={
        x: r * T.s.sin(theta) * T.s.cos(phi),
        y: r * T.s.sin(theta) * T.s.sin(phi),
        z: r * T.s.cos(theta),
    },
)

Spherical.set_coord_transformation(
    target=Cartesian,
    rules={
        r: T.s.sqrt(arg=x**2 + y**2 + z**2),
        theta: T.s.acos(z / T.s.sqrt(arg=x**2 + y**2 + z**2)),
        phi: T.s.atan2(y, x),
    },
)
```

Launchpad ⊗ 0 ⚠ 0 0 10.76% 18.90/63.68 GB Spaces: 4 LF Cell 23 of 23

```python
Minkowski.show(coords=Spherical)
```

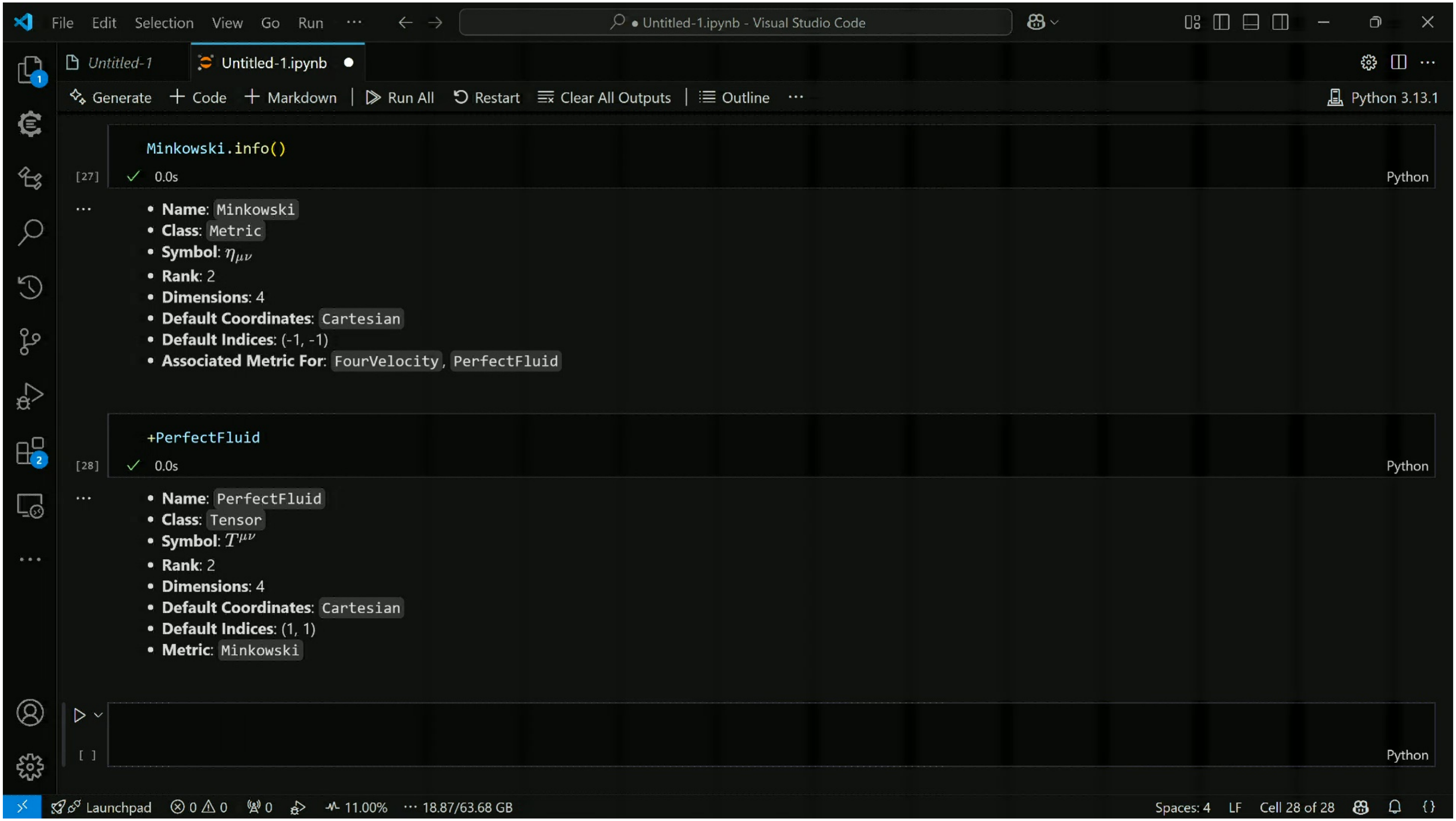
[25] ✓ 0.1s

$$\eta_{\mu\nu}\Big|_{(t,r,\theta,\phi)} = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & r^2 & 0 \\ 0 & 0 & 0 & r^2\sin^2(\theta) \end{pmatrix}$$

```python
PerfectFluid.show(coords=Spherical, indices=(1, 1))
```

[26] ✓ 0.1s

$$T^{\mu\nu}\Big|_{(t,r,\theta,\phi)} = \begin{pmatrix} \rho & 0 & 0 & 0 \\ 0 & p & 0 & 0 \\ 0 & 0 & \frac{p}{r^2} & 0 \\ 0 & 0 & 0 & \frac{p}{r^2\sin^2(\theta)} \end{pmatrix}$$


File Edit Selection View Go Run ···
Untitled-1.ipynb - Visual Studio Code
Untitled-1
Untitled-1.ipynb
Generate + Code + Markdown Run All Restart Clear All Outputs Outline ···
Python 3.13.1
Minkowski.info()
[27] 0.0s
Python
• Name: Minkowski
• Class: Metric
• Symbol: $\eta_{\mu\nu}$
• Rank: 2
• Dimensions: 4
• Default Coordinates: Cartesian
• Default Indices: (-1, -1)
• Associated Metric For: FourVelocity, PerfectFluid
+PerfectFluid
[28] 0.0s
Python
• Name: PerfectFluid
• Class: Tensor
• Symbol: $T^{\mu\nu}$
• Rank: 2
• Dimensions: 4
• Default Coordinates: Cartesian
• Default Indices: (1, 1)
• Metric: Minkowski
[ ]
Python
Launchpad 0 0 0 11.00% 18.87/63.68 GB
Spaces: 4 LF Cell 28 of 28

File Edit Selection View Go Run ··· ● Untitled-1.ipynb - Visual Studio Code

Untitled-1 | Untitled-1.ipynb ●

Generate + Code + Markdown | Run All | Restart | Clear All Outputs | Outline ··· | Python 3.13.1

$$N_{\mu\nu}\Big|_{(t,x,y,z)} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

```python
Minkowski(mu, nu) + NonSymmetric(mu, nu)
```

[31] ✓ 0.0s Python

$$\eta_{\mu\nu} + N_{\mu\nu}\Big|_{(t,x,y,z)} = \begin{pmatrix} -1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```python
Minkowski(mu, nu) + NonSymmetric(nu, mu)
```

[32] ✓ 0.0s Python

$$\eta_{\mu\nu} + N_{\nu\mu}\Big|_{(t,x,y,z)} = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

[ ] Python

Launchpad ⊗ 0 ⚠ 0 0 11.23% ··· 19.12/63.68 GB Spaces: 4 LF Cell 32 of 32

```python
2 * Minkowski(mu, nu)
```

[35] ✓ 0.0s

$$2\eta_{\mu\nu}\Big|_{(t,x,y,z)} = \begin{pmatrix} -2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix}$$

```python
PerfectFluidFromVelocity: Any = (rho + p) * FourVelocity(mu) @ FourVelocity(nu) + p * Minkowski(mu, nu)
# PerfectFluidFromVelocity.symbol = "T"
# PerfectFluidFromVelocity
```

[36] ✓ 0.1s

$$T^{\mu\nu}\Big|_{(t,x,y,z)} = \begin{pmatrix} \frac{-\rho-pv^2}{v^2-1} & -\frac{v(\rho+p)}{v^2-1} & 0 & 0 \\ -\frac{v(\rho+p)}{v^2-1} & \frac{-\rho v^2-p}{v^2-1} & 0 & 0 \\ 0 & 0 & p & 0 \\ 0 & 0 & 0 & p \end{pmatrix}$$

```python
PerfectFluidFromVelocity: Any = (rho + p) * FourVelocity(mu) @ FourVelocity(nu) + p * Minkowski(mu, nu)
# PerfectFluidFromVelocity.symbol = "T"
# PerfectFluidFromVelocity
```

[37] ✓ 0.0s Python

$$((\rho + p)\, v^{\mu}) v^{\nu} + p\eta^{\mu\nu}\Big|_{(t,x,y,z)} = \begin{pmatrix} \frac{-\rho - pv^2}{v^2-1} & -\frac{v(\rho+p)}{v^2-1} & 0 & 0 \\ -\frac{v(\rho+p)}{v^2-1} & \frac{-pv^2-p}{v^2-1} & 0 & 0 \\ 0 & 0 & p & 0 \\ 0 & 0 & 0 & p \end{pmatrix}$$

```python
PerfectFluidFromVelocity.show(replace={v: 0})
```

[38] ✓ 0.0s Python

$$((\rho + p)\, v^{\mu}) v^{\nu} + p\eta^{\mu\nu}\Big|_{(t,x,y,z)} = \begin{pmatrix} \rho & 0 & 0 & 0 \\ 0 & p & 0 & 0 \\ 0 & 0 & p & 0 \\ 0 & 0 & 0 & p \end{pmatrix}$$

```python
~Schwarzschild.christoffel()
```

[ ] Python

File Edit Selection View Go Run ··· ● Untitled-1.ipynb - Visual Studio Code

Untitled-1 | Untitled-1.ipynb ●

Generate | + Code | + Markdown | Interrupt | Restart | Clear All Outputs | Go To | Outline ··· | Python 3.13.1

$$\partial_t \Gamma^r{}_{rt} - \partial_r \Gamma^r{}_{tt} + \Gamma^r{}_{t\mu}\Gamma^\mu{}_{rt}$$

$$\partial_t \Gamma^\theta{}_{\theta t} - \partial_\theta \Gamma^\theta{}_{tt} + \Gamma^\theta{}_{t\mu}\Gamma^\mu{}_{\theta t} - \Gamma^\theta{}_{\theta\mu}\Gamma^\mu{}_{tt} = \partial_t \Gamma^\phi{}_{\phi t} - \partial_\phi \Gamma^\phi{}_{tt} + \Gamma^\phi{}_{t\mu}\Gamma^\mu{}_{\phi t}$$

$$\partial_r \Gamma^t{}_{tr} - \partial_t \Gamma^t{}_{rr} + \Gamma^t{}_{r\mu}\Gamma^\mu{}_{tr}$$

$$\partial_r \Gamma^r{}_{tt} - \partial_t \Gamma^r{}_{rt} + \Gamma^r{}_{r\mu}\Gamma^\mu{}_{tt}$$

$$\partial_r \Gamma^\theta{}_{\theta r} - \partial_\theta \Gamma^\theta{}_{rr} + \Gamma^\theta{}_{r\mu}\Gamma^\mu{}_{\theta r} - \Gamma^\theta{}_{\theta\mu}\Gamma^\mu{}_{rr} = \partial_r \Gamma^\phi{}_{\phi r} - \partial_\phi \Gamma^\phi{}_{rr} + \Gamma^\phi{}_{r\mu}\Gamma^\mu{}_{\phi r}$$

$$\partial_\theta \Gamma^\theta{}_{tt} - \partial_t \Gamma^\theta{}_{\theta t} + \Gamma^\theta{}_{\theta\mu}\Gamma^\mu{}_{tt} - \Gamma^\theta{}_{t\mu}\Gamma^\mu{}_{\theta t} = \partial_\phi \Gamma^\phi{}_{tt} - \partial_t \Gamma^\phi{}_{\phi t} + \Gamma^\phi{}_{\phi\mu}\Gamma^\mu{}_{tt}$$

$$\partial_\theta \Gamma^\theta{}_{rr} - \partial_r \Gamma^\theta{}_{\theta r} + \Gamma^\theta{}_{\theta\mu}\Gamma^\mu{}_{rr} - \Gamma^\theta{}_{r\mu}\Gamma^\mu{}_{\theta r} = \partial_\phi \Gamma^\phi{}_{rr} - \partial_r \Gamma^\phi{}_{\phi r} + \Gamma^\phi{}_{\phi\mu}\Gamma^\mu{}_{rr}$$

$$\partial_\theta \Gamma^\theta{}_{\phi\phi} - \partial_\phi \Gamma^\theta{}_{\theta\phi} + \Gamma^\theta{}_{\theta\mu}\Gamma^\mu{}_{\phi\phi} - \Gamma^\theta{}_{\phi\mu}\Gamma^\mu{}_{\theta\phi} = -\partial_\phi \Gamma^\theta{}_{\theta\phi} - \partial_\theta \Gamma^\theta{}_{\phi\phi} + \Gamma^\theta{}_{\phi\mu}\Gamma^\mu{}_{\theta\phi}$$

$$\partial_\theta \Gamma^\phi{}_{\phi\theta} - \partial_\phi \Gamma^\phi{}_{\theta\theta} + \Gamma^\phi{}_{\theta\mu}\Gamma^\mu{}_{\phi\theta} - \Gamma^\phi{}_{\phi\mu}\Gamma^\mu{}_{\theta\theta} = -\partial_\phi \Gamma^\phi{}_{\theta\theta} - \partial_\theta \Gamma^\phi{}_{\phi\theta} + \Gamma^\phi{}_{\phi\mu}\Gamma^\mu{}_{\theta\theta}$$

[42]
```
Schwarzschild.riemann(rho, sigma, mu, nu) @ Schwarzschild.riemann(rho, sigma, mu, nu)
```
22.4s — Python

[ ] — Python

Launchpad ⊗ 0 ⚠ 0 | 0 | 10.69% | 19.14/63.68 GB | Spaces: 4 LF Cell 41 of 41

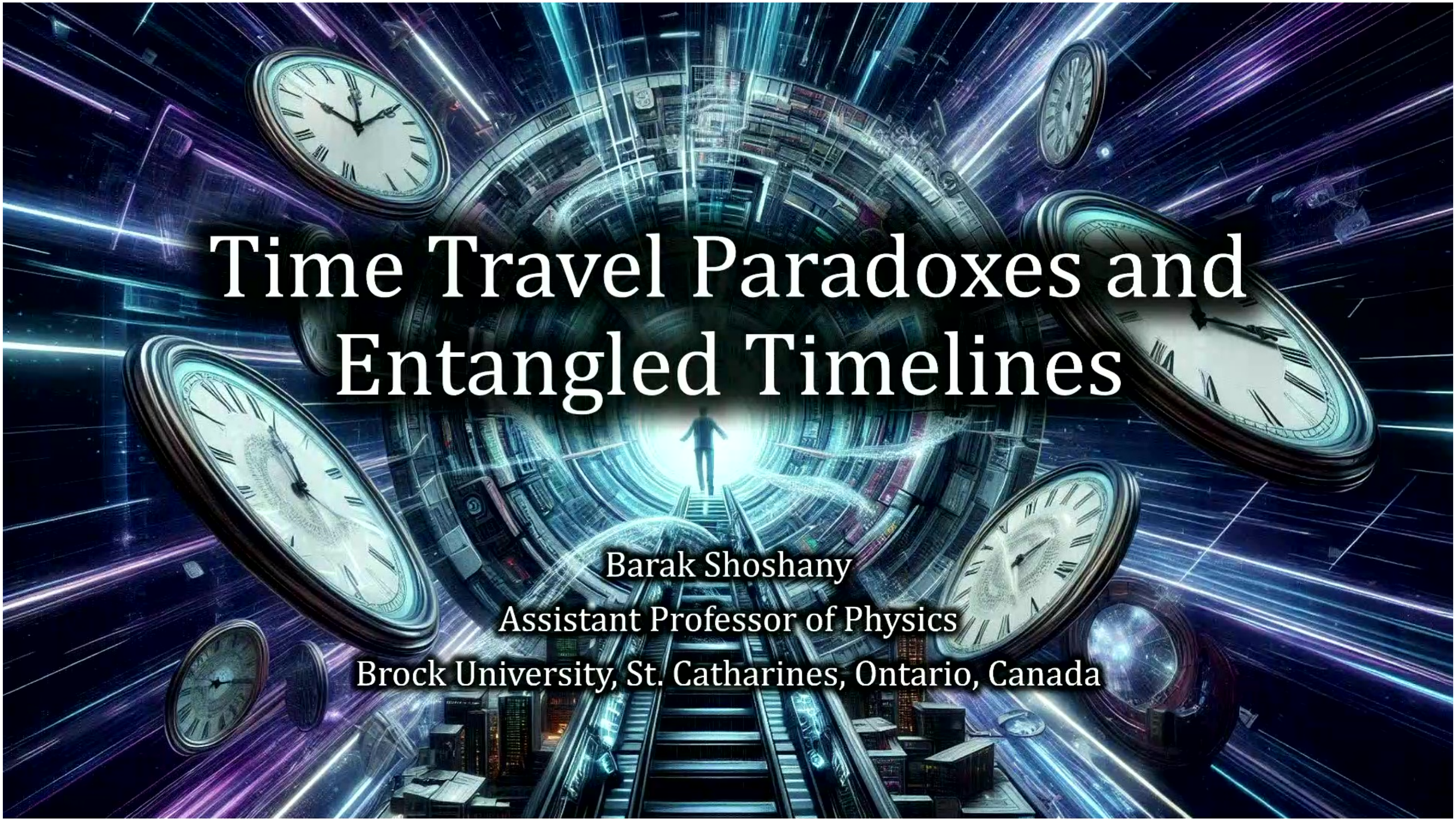
$$R_{rr} = \partial_{t^2}aa + 2\partial_t a^2$$
$$R_{\theta\theta} = r^2\left(\partial_{t^2}aa + 2\partial_t a^2\right)$$
$$R_{\phi\phi} = r^2\left(\partial_{t^2}aa + 2\partial_t a^2\right)\sin^2(\theta)$$

[45]
```python
FLRW.ricci_tensor(mu, mu)
```
✓ 0.0s

$$R_\mu{}^\mu\Big|_{(t,r,\theta,\phi)} = \frac{6\left(\partial_{t^2}aa + \partial_t a^2\right)}{a^2}$$

[46]
```python
FLRW.ricci_scalar()
```
✓ 0.0s

$$R\Big|_{(t,r,\theta,\phi)} = \frac{6\left(\partial_{t^2}aa + \partial_t a^2\right)}{a^2}$$


Time Travel Paradoxes and Entangled Timelines
Barak Shoshany
Assistant Professor of Physics
Brock University, St. Catharines, Ontario, Canada

Motivation

- Time travel may be possible
  - GR solutions with CTCs/CCCs
  - FTL: wormholes/warp drives → time travel?
- If possible: paradoxes?
  - Consistency: grandfather
  - Bootstrap: time loops
  - Inconsistent with known physics
- Proposed solutions
  - Hawking chronology protection: boring
  - Novikov self-consistency: problematic
  - **Parallel timelines: solve all paradoxes, but no concrete models**

Parallel timeline models

- General relativity: branching spacetimes
  - Non-Hausdorff manifold? (Hausdorff = points have disjoint neighborhoods)
  - Non-locally-Euclidean? (Locally Euclidean = every point has a neighborhood homeomorphic to $\mathbb{R}^n$)
  - Mathematically intractable, no branching mechanism
- Quantum mechanics
  - Deutsch D-CTCs: Everett / "many-worlds" interpretation (MWI)
  - Uses reduced/mixed states, destroys information about timelines
  - **Our new model: "entangled timelines" (E-CTCs) [with Zipora Stober]**

The generic paradox

- Independent of geometry/topology or specific physical system
- Time machine $\mathcal{H}_{\mathrm{CTC}}$: occupation number
  - $|0\rangle =$ empty (time travel has not occurred)
  - $|1\rangle =$ not empty (time travel has occurred)
- External system $\mathcal{H}_{\mathrm{ex}}$ (human, billiard ball, particle): control bit
  - $|0\rangle =$ won't go in (time travel will not occur)
  - $|1\rangle =$ will go in (time travel will occur)
- Logical, not physical states

The generic paradox

- $|\Psi(t)\rangle = \mathcal{H}_{\text{CTC}} \otimes \mathcal{H}_{\text{ex}}$
- True consistency paradox: time travel iff no time travel
- Cyclic
- Novikov does not apply; no way to make this consistent

Time	State/Event
$t = 0$	$\lvert 0\rangle \otimes \lvert 1\rangle$
🕒	Nothing happens
$t = 1$	$\lvert 0\rangle \otimes \lvert 1\rangle$
🌀	Time travel occurs
$t = 0$	$\lvert 1\rangle \otimes \lvert 1\rangle$
🕒	Time travel is prevented
$t = 1$	$\lvert 1\rangle \otimes \lvert 0\rangle$
🌀	Time travel does not occur
$t = 0$	$\lvert 0\rangle \otimes \lvert 1\rangle$

Alice and the bomb

- At $t = 1$, Alice puts bomb inside time machine, sends to $t = 0$
- At $t = 0$, Alice opens the time machine door and is killed
- Now Alice cannot send bomb at $t = 1$, so she's alive at $t = 0$
- Classic consistency paradox
- Macroscopic: cannot be precise, issues of free will, etc…
- Treat Alice's states as equivalence class of states with same macroscopic behavior

Alice and the bomb

- Formulate using generic paradox.
- For $\mathcal{H}_{\mathrm{CTC}}$: (time machine)
  - $|0\rangle \equiv |\text{empty}\rangle$
  - $|1\rangle \equiv |\text{bomb}\rangle$
- For $\mathcal{H}_{\mathrm{ex}}$: (Alice)
  - $|0\rangle \equiv |\text{dead}\rangle$
  - $|1\rangle \equiv |\text{alive}\rangle$

Time	State/Event
$t = 0$	$\|\text{empty}\rangle \otimes \|\text{alive}\rangle$
	Nothing happens
$t = 1$	$\|\text{empty}\rangle \otimes \|\text{alive}\rangle$
	Alice sends a bomb back in time
$t = 0$	$\|\text{bomb}\rangle \otimes \|\text{alive}\rangle$
	Bomb explodes, Alice dies
$t = 1$	$\|\text{bomb}\rangle \otimes \|\text{dead}\rangle$
	Dead Alice cannot send a bomb
$t = 0$	$\|\text{empty}\rangle \otimes \|\text{alive}\rangle$

Particle annihilation

- Microscopic (simpler)
- For $\mathcal{H}_{\text{CTC}}$: (time machine)
  - $|0\rangle \equiv |\text{empty}\rangle$
  - $|1\rangle \equiv |\text{particle}\rangle$
- For $\mathcal{H}_{\text{ex}}$: (particle)
  - $|0\rangle \equiv |\text{annihilated}\rangle$
  - $|1\rangle \equiv |\text{not annihilated}\rangle$

Time	State/Event
$t = 0$	$|\text{empty}\rangle \otimes |\text{not annihilated}\rangle$
🕒	**Nothing happens**
$t = 1$	$|\text{empty}\rangle \otimes |\text{not annihilated}\rangle$
🌀	**Particle goes into time machine**
$t = 0$	$|\text{particle}\rangle \otimes |\text{not annihilated}\rangle$
🕒	**Past and future particles annihilate**
$t = 1$	$|\text{particle}\rangle \otimes |\text{annihilated}\rangle$
🌀	**Particle doesn't go into time machine**
$t = 0$	$|\text{empty}\rangle \otimes |\text{not annihilated}\rangle$

Note: This means a particle has traveled through, not that it still exists

Classical timelines: generic

Time	Timeline $h = 0$	Timeline $h = 1$
$t = 0$	$\|0\rangle \otimes \|1\rangle$	$\|1\rangle \otimes \|1\rangle$
🕒	Nothing happens	Time travel is prevented
$t = 1$	$\|0\rangle \otimes \|1\rangle$	$\|1\rangle \otimes \|0\rangle$
🌀	Time travel occurs	Time travel does not occur

Note: $\mathcal{H}_{\mathrm{CTC}}$ state $|0\rangle$ or $|1\rangle$ indicates the timeline

Classical timelines: macroscopic

Time	Timeline $h = 0$	Timeline $h = 1$
$t = 0$	$\lvert\text{empty}\rangle \otimes \lvert\text{alive}\rangle$	$\lvert\text{bomb}\rangle \otimes \lvert\text{alive}\rangle$
	Nothing happens	**Bomb explodes, Alice dies**
$t = 1$	$\lvert\text{empty}\rangle \otimes \lvert\text{alive}\rangle$	$\lvert\text{bomb}\rangle \otimes \lvert\text{dead}\rangle$
	Alice sends a bomb back in time	**Dead Alice cannot send a bomb**

Classical timelines: microscopic

Time	Timeline $h = 0$	Timeline $h = 1$
$t = 0$	$\lvert\text{empty}\rangle \otimes \lvert\text{not annihilated}\rangle$	$\lvert\text{particle}\rangle \otimes \lvert\text{not annihilated}\rangle$
🕒	**Nothing happens**	**Past and future particles annihilate**
$t = 1$	$\lvert\text{empty}\rangle \otimes \lvert\text{not annihilated}\rangle$	$\lvert\text{particle}\rangle \otimes \lvert\text{annihilated}\rangle$
🌀	**Particle goes into time machine**	**Particle doesn't go into time machine**

Quantum superposition

- $\mathcal{H}_{\text{ex}}$ always has initial condition $|1\rangle$, so no superposition at $t = 0$
- But $\mathcal{H}_{\text{CTC}}$ can be in superposition:
$$|\Psi(0)\rangle = (\alpha|0\rangle + \beta|1\rangle) \otimes |1\rangle, \qquad |\alpha|^2 + |\beta|^2 = 1$$
- Unitary evolution operator U from $t = 0$ to $t = 1$:
$$U(|0\rangle \otimes |1\rangle) = |0\rangle \otimes |1\rangle, \qquad U(|1\rangle \otimes |1\rangle) = |1\rangle \otimes |0\rangle$$
- Just CNOT gate:
$$U(|x\rangle \otimes |y\rangle) = |x\rangle \otimes |x \dot{+} y\rangle$$

($x, y \in \mathbb{Z}_2$ and $\dot{+}$ is addition mod 2)
$$|\Psi(1)\rangle = U|\Psi(0)\rangle = \alpha|0\rangle \otimes |1\rangle + \beta|1\rangle \otimes |0\rangle$$
- State became entangled!

Quantum superposition

- Define timeline correlation operator T between $t = 1$ and $t = 0$ at different h:

$$T(|x\rangle \otimes |y\rangle) = |y\rangle \otimes |1\rangle$$

- Not unitary; lost info about $|x\rangle$, as it does not affect time travel
- Not evolution; just correlation

$$T|\Psi(1)\rangle = \alpha|1\rangle \otimes |1\rangle + \beta|0\rangle \otimes |1\rangle$$

- Compare to initial state:

$$|\Psi(0)\rangle = \alpha|0\rangle \otimes |1\rangle + \beta|1\rangle \otimes |1\rangle$$

- So $\alpha = \beta = 1/\sqrt{2}$ (up to phase)

Collapse interpretation

- State at $t = 1$:

$$|\Psi(1)\rangle = \frac{1}{\sqrt{2}}(|0\rangle \otimes |1\rangle + |1\rangle \otimes |0\rangle)$$

- Collapses to $|0\rangle \otimes |1\rangle$ or $|1\rangle \otimes |0\rangle$ with 50% probability
- Correlates with either $|1\rangle \otimes |1\rangle$ or $|0\rangle \otimes |1\rangle$ at $t = 0$
- Collapse will destroy superposition, so paradox reappears
- Quantum superposition solution doesn't work in collapse models!

Many-worlds interpretation

- Alice has a qubit:

$$|\text{qubit}\rangle = a|0\rangle + b|1\rangle$$

- Measurement: collapses to 0 or 1, **non-unitary** evolution
- Solution: consider Alice's state too

$$|\Psi(0)\rangle = (a|0\rangle + b|1\rangle) \otimes |\text{Alice}\rangle$$
$$|\Psi(1)\rangle = a|0\rangle \otimes |\text{Alice saw 0}\rangle + b|1\rangle \otimes |\text{Alice saw 1}\rangle$$

- Unitary evolution (similar to CNOT gate)
- MWI = "unmodified" (purely unitary) QM

Entangled worlds

- Recall:

$$|\Psi(1)\rangle = a|0\rangle \otimes |\text{Alice saw } 0\rangle + b|1\rangle \otimes |\text{Alice saw } 1\rangle$$

- Alice and qubit are now entangled; Alice "branches" into two
- Each Alice sees "collapse" from her own perspective
- Superposition $\neq$ measurement outcomes or knowledge
- Interpret as multiplicity of "worlds"
- Not physically distinct universes! Just 2 different terms in universal quantum state

Spreading of branches

- Consider Bob. Before measurement:
$$|\Phi(0)\rangle = (a|0\rangle + b|1\rangle) \otimes |\text{Alice}\rangle \otimes |\text{Bob}\rangle$$
- Alice measures:
$$|\Phi(1)\rangle = (a|0\rangle \otimes |\text{Alice saw } 0\rangle + b|1\rangle \otimes |\text{Alice saw } 1\rangle) \otimes |\text{Bob}\rangle$$
- Alice tells Bob what she measured:
$$|\Phi(2)\rangle = a|0\rangle \otimes |\text{Alice saw } 0\rangle \otimes |\text{Bob heard } 0\rangle$$
$$+b|1\rangle \otimes |\text{Alice saw } 1\rangle) \otimes |\text{Bob heard } 1\rangle$$
- All 3 systems now entangled; branching spread further
- At $t \rightarrow \infty$ branching will spread to the entire universe (causal future)
- Branching is not global; it's local and spreads causally

Common misconceptions

- **Misconception 1:** branching happens upon measurement
- Correction: branching happens upon interaction between any two systems (e.g. Alice tells Bob the result)
- **Misconception 2:** branching instantaneously creates entire new parallel universes from scratch
- Correction: branching is just gradual and causal spreading of entanglement to more systems within a single universe
- Better name for MWI: "entangled worlds" or "entangled histories"?

Entangled timelines

- Back to time travel. At $t = 0$, state is separable:
$$|\Psi(0)\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |1\rangle$$
- Time machine ($\mathcal{H}_{\text{CTC}}$) has 2 timelines; external system ($\mathcal{H}_{\text{ex}}$) has only 1 timeline
- At $t = 1$, state is entangled:
$$|\Psi(1)\rangle = \frac{1}{\sqrt{2}}(|0\rangle \otimes |1\rangle + |1\rangle \otimes |0\rangle)$$
- Both systems have 2 timelines
- Systems entangled; can't talk about one without the other, so both share the same timelines

Spreading timelines

- Timelines spread locally like branches in MWI
- Macroscopic: Bob opens door to lab, finds Alice alive or dead
- Microscopic: detector detects products of annihilation or not
- "Parallel timelines" or "parallel universes" are just local branching
- Start by interacting with time machine, then spread out

Conclusions

- QM with the MWI provides a simple and natural way to resolve time travel paradoxes
- The abstract generic paradox qubit model can be mapped onto more complicated macroscopic/microscopic models
- No need to worry about mechanisms for creating new physically distinct universes
- There is only one universe; timelines are emergent structures resulting from entanglement between systems
- Timelines propagate locally, gradually, and causally

References

- Barak Shoshany, "Lectures on Faster-Than-Light Travel and Time Travel", arXiv:1907.04178
- Jacob Hauser and Barak Shoshany, "Time Travel Paradoxes and Multiple Histories", arXiv:1911.11590
- Barak Shoshany and Jared Wogan, "Wormhole Time Machines and Multiple Histories", arXiv:2110.02448
- Barak Shoshany and Ben Snodgrass, "Warp Drives and Closed Timelike Curves", arXiv:2309.10072
- Barak Shoshany and Zipora Stober, "Time Travel Paradoxes and Entangled Timelines", arXiv:2303.07635