Title: Tutorial: Coarse-graining of Spin Foams

Date: Jun 21, 2017 03:45 PM

URL: http://pirsa.org/17060080

Abstract:

Tutorial: Renormalization of hypercuboidal spin foams

Sebastian Steinhaus

sebastian.steinhaus@desy.de

II. Institute f. Theoretical Physics
University of Hamburg

Make quantum gravity computable
@ Perimeter Institute, Waterloo

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Emmy Noether-Programm
DFG

Steinhaus_numerical.pdf - Adobe Acrobat Reader DC

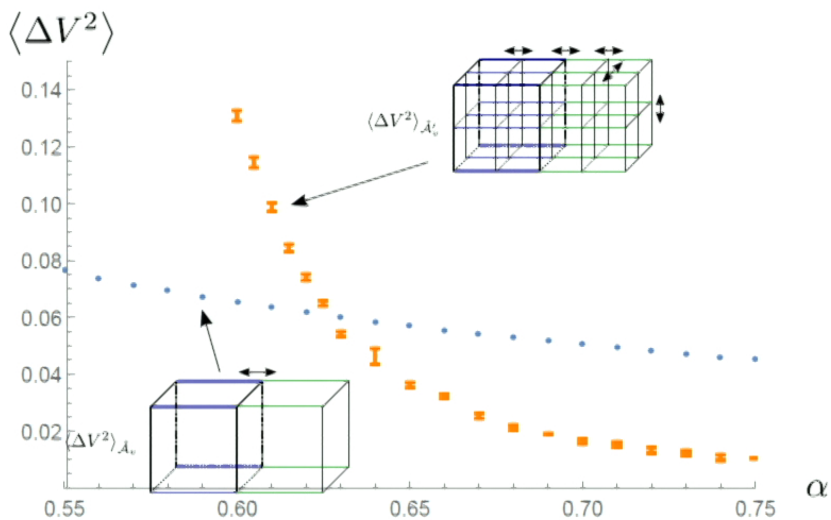
Datei Bearbeiten Anzeige Fenster Hilfe

Start Werkzeuge Steinhaus_numeric... × Anmelden

2 (2 von 9) 154%

What we want to compute today

- Compare observables for 32 'quantum cuboids' and 2 'quantum cuboids' [Bahr, Steinhaus, PRL '16 (1605.07649), PRD '17 (1701.02311)]



Goal of today: Compute expectation values of observable $\Delta \mathcal{V}_1^2$, find the fixed point and identify whether it is attractive or repulsive.

DEU 15:46 21.06.2017

Steinhaus_numerical.pdf - Adobe Acrobat Reader DC

Hypercuboid spin foams

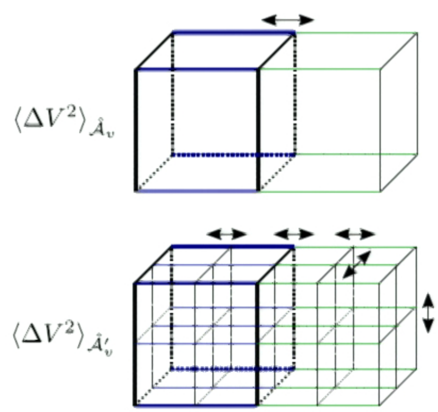
- Consider a 4D hypercubic lattice
- Geometry of hypercuboid encoded in spin foam amplitude $\mathcal{A}(\alpha, x, y, z, t)$ [Bahr, Steinhaus PRD '15 (1508.07961)]
  - α single parameter of the model
  - x, y, z, t are the edge lengths of the hypercuboid.
- $\mathcal{A} \sim$ 'probability' of cuboid with lengths x, y, z, t.
- Partition function Z:

$$Z = \int \prod_{i=0}^{L} (dx_i\, dy_i\, dz_i\, dt_i) \prod_v \mathcal{A}_v(\alpha, x_i, y_i, z_i, t_i)$$

- Observables $\mathcal{O}$:

$$\langle \mathcal{O} \rangle = \frac{1}{Z} \int \prod_{i=0}^{L} (dx_i\, dy_i\, dz_i\, dt_i) \mathcal{O}(\{x_i, y_i, \dots\}) \prod_v \mathcal{A}_v(\alpha, x_i, y_i, z_i, t_i)$$

Finite lattice with boundary



- Consider two finite lattices
- 'Coarse': 2 hypercubes glued in 'time' direction
- 'Fine': 32 hypercubes glued in 'time' direction
- Boundary data: X, Y, Z, T

$$\sum \text{Short length} \overset{!}{=} \text{Boundary length}$$

Exercise 1: Partition functions

What is the partition function for these two lattices with fixed boundary data? How many edge lengths? How many constraints?

Steinhaus_numerical.pdf - Adobe Acrobat Reader DC

Observables: Volume of one hypercuboid

$$Z_2 = \int dt_1\, \mathcal{A}_2(X, Y, Z, T, t_1)$$

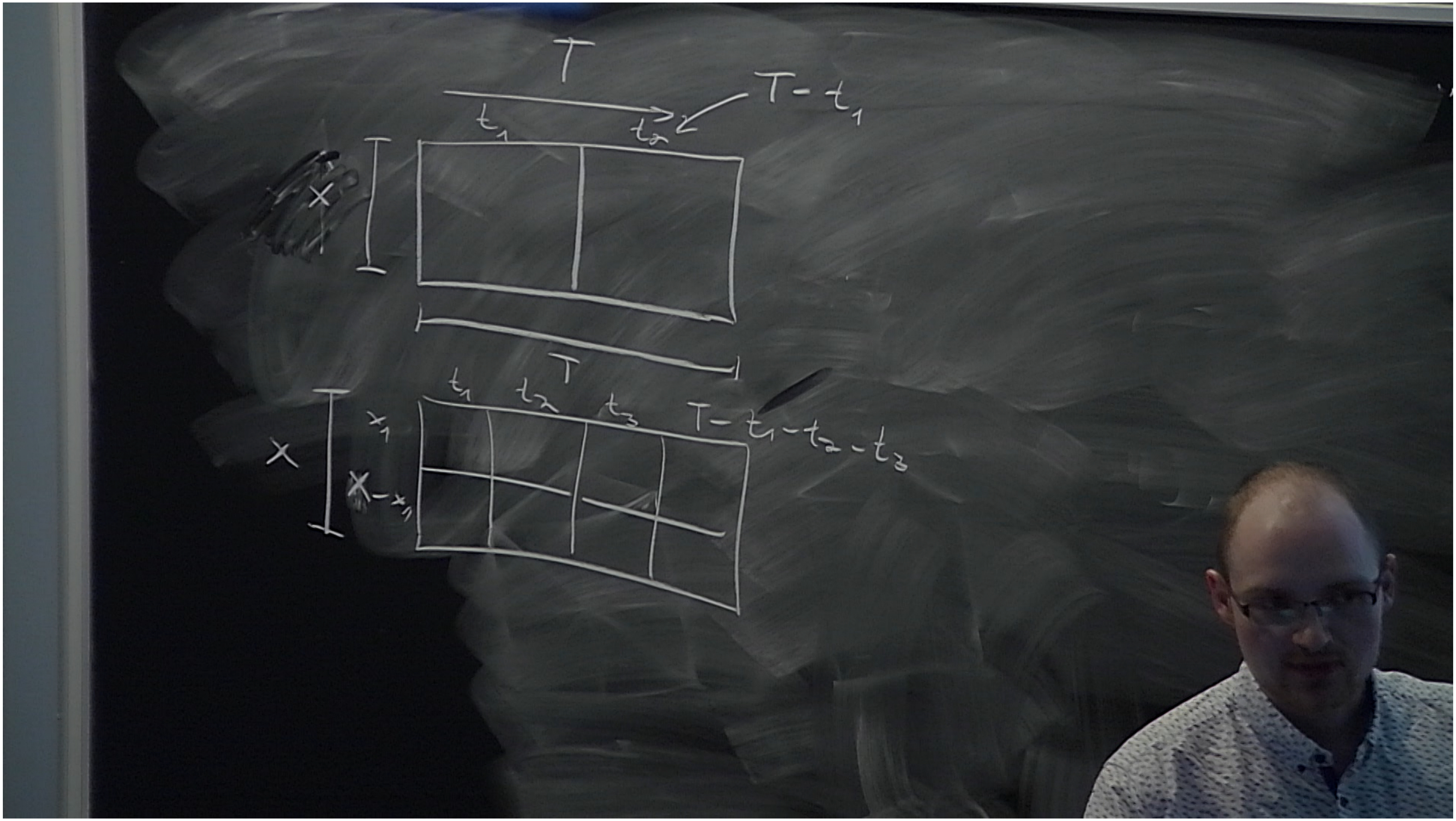
$$Z_{32} = \int dx_1\, dy_1\, dz_1\, dt_1\, dt_2\, dt_3\, \mathcal{A}_{32}(X, Y, Z, T, x_1, y_1, z_1, t_1, t_2, t_3)$$

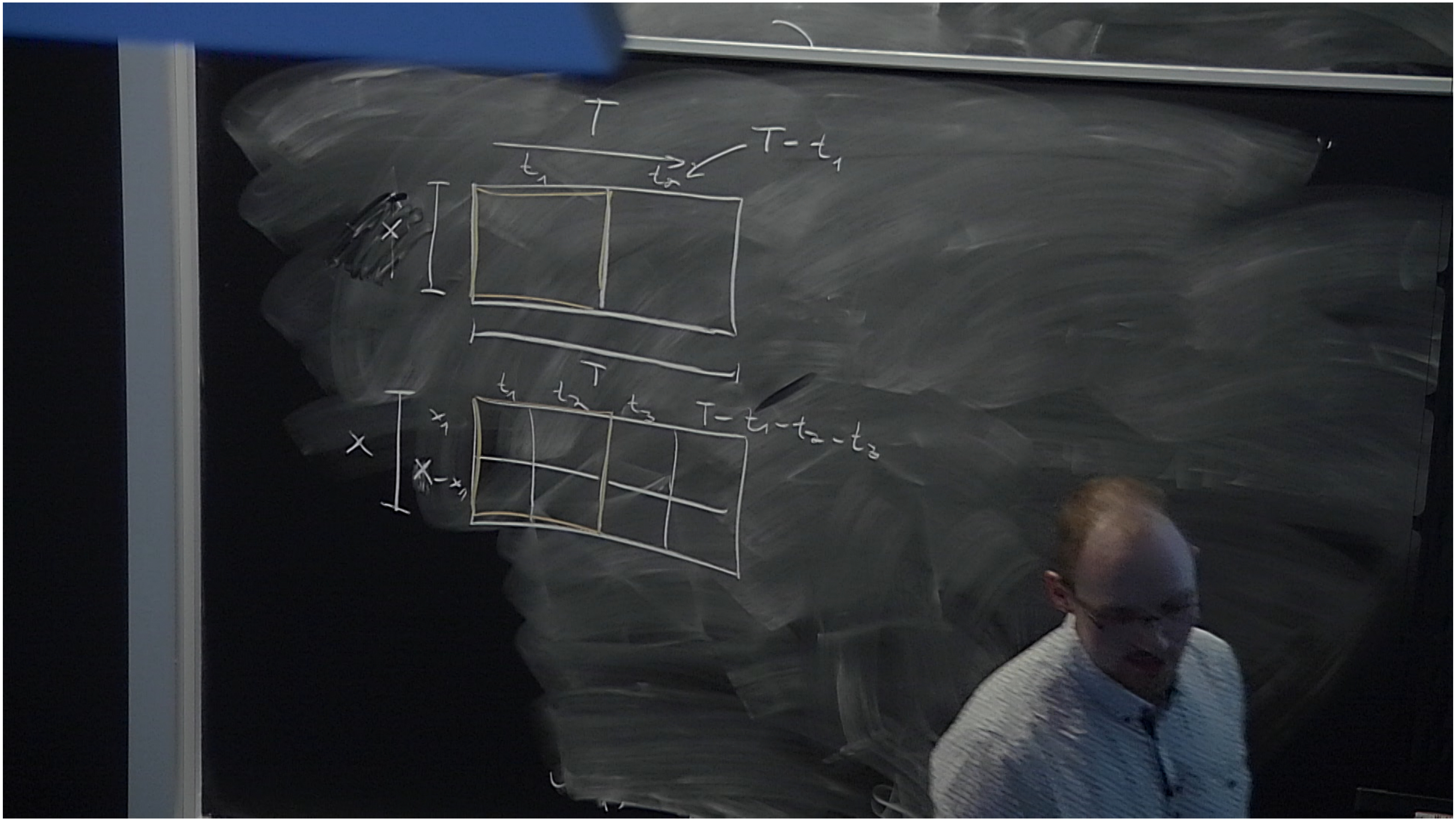
- What is an interesting observable?

Exercise 2: Volume of one hypercuboid

Define the observable 'volume of one hypercuboid' $\mathcal{V}_1$ (half of the lattice)?

Do you have an expectation for $\langle \mathcal{V}_1 \rangle$?


T
T − t1
t1
t2
X
T
t1
t2
t3
T − t1 − t2 − t3
x1
X
X − x1



T
$T-t_1$
t_1
t_2
X
T
t_1
t_2
t_3
$T-t_1-t_2-t_3$
X
x_1
$X-x_1$

Steinhaus_numerical.pdf - Adobe Acrobat Reader DC

Actual amplitudes

- Look at file `Quantum-Cuboid-renormalization`.
- Hypercuboid amplitude: $\mathcal{A}$ as `Am(...)`
- Definition of integrands:
  - $\mathcal{A}_2$ as `Am2(...)`
  - $\mathcal{A}_{32}$ as `Am32(...)`

WarmUp: Volume

Write a function returning the volume of a 4D cuboid.

Question: How can we perform the integrations numerically?

Apply simple Monte Carlo methods.

- Random sampling
- Importance sampling

localhost:8888/notebooks/AppData/Local/Julia-0.5.2/Quantum-Cuboid-renormalization.ipynb

jupyter Quantum-Cuboid-renormalization Last Checkpoint: 8 hours ago (autosaved)

Tutorial: Quantum cuboid renormalisation

Basic definitions

In [ ]:

```
function DE(j1::Float64,j2::Float64,j3::Float64,j4::Float64,j5::Float64,j6::Float64)
    res = (-2) * (j1^2 * (j2 + j4) + j2 * j4 * (j2 + j4) + j1 * (j2^2 + (1 + 1im) * j2 * j4 + j4^2))
    res *= (j1^2 * (j3 + j5) + j3 * j5 * (j3 + j5) + j1 * (j3^2 + (1 + 1im) * j3 * j5 + j5^2))
    res *= (j3 * j4 * j5 + j2 * (j4 * j5 + j3 * (j4 + j5)))
    res *= (j2^2 * (j3 + j6) + j3 * j6 * (j3 + j6) + j2 * (j3^2 + (1 + 1im) * j3 * j6 + j6^2))
    res *= (j4^2 * (j5 + j6) + j5 * j6 * (j5 + j6) + j4 * (j5^2 + (1 + 1im) * j5 * j6 + j6^2))
    res *= (j3 * j4 * j6 + j1 * (j4 * j6 + j3 * (j4 + j6)))
    res *= (j2 * j5 * j6 + j1 * (j5 * j6 + j2 * (j5 + j6)))

    res
end

function DF(j1::Float64,j2::Float64,j3::Float64,j4::Float64,j5::Float64,j6::Float64)
    res = (-2) * (j1^2 * (j2 + j4) + j2 * j4 * (j2 + j4) + j1 * (j2^2 + (1 - 1im) * j2 * j4 + j4^2))
    res *= (j1^2 * (j3 + j5) + j3 * j5 * (j3 + j5) + j1 * (j3^2 + (1 - 1im) * j3 * j5 + j5^2))
    res *= (j3 * j4 * j5 + j2 * (j4 * j5 + j3 * (j4 + j5)))
    res *= (j2^2 * (j3 + j6) + j3 * j6 * (j3 + j6) + j2 * (j3^2 + (1 - 1im) * j3 * j6 + j6^2))
    res *= (j4^2 * (j5 + j6) + j5 * j6 * (j5 + j6) + j4 * (j5^2 + (1 - 1im) * j5 * j6 + j6^2))
    res *= (j3 * j4 * j6 + j1 * (j4 * j6 + j3 * (j4 + j6)))
    res *= (j2 * j5 * j6 + j1 * (j5 * j6 + j2 * (j5 + j6)))

    res
end

function Ampl(α::Float64,j1::Float64,j2::Float64,j3::Float64,j4::Float64,j5::Float64,j6::Float64)
    res = 1/((2. * pi)^3) * 2.^(24)
    res *= (1/(sqrt(-DE(j1,j2,j3,j4,j5,j6))) + 1/(sqrt(-DF(j1,j2,j3,j4,j5,j6))))
    res *= (1/(sqrt(-DE(j1,j2,j3,j4,j5,j6))) + 1/(sqrt(-DF(j1,j2,j3,j4,j5,j6))))
```

localhost:8888/notebooks/AppData/Local/Julia-0.5.2/Quantum-Cuboid-renormalization.ipynb

jupyter Quantum-Cuboid-renormalization Last Checkpoint: 8 hours ago (autosaved)

File Edit View Insert Cell Kernel Help Julia 0.5.0

```
if t1 > 0.001 && t2 > 0.001 && t3 > 0.001 && T-t1-t2-t3 > 0.001 && x > 0.001 &&
    y > 0.001 && z > 0.001 && X-x > 0.001 && Y-y > 0.001 && Z-z > 0.001

    result = 1

    result *= Am(a, x, y, z, t1)*Am(a, x, y, z, t2)*Am(a, x, y, z, t3)*
    Am(a, x, y, z, T-t1-t2-t3)

    result *= Am(a, X-x, y, z, t1)*Am(a, X-x, y, z, t2)*Am(a, X-x, y, z, t3)*
    Am(a, X-x, y, z, T-t1-t2-t3)

    result *= Am(a, x, Y-y, z, t1)*Am(a, x, Y-y, z, t2)*Am(a, x, Y-y, z, t3)*
    Am(a, x, Y-y, z, T-t1-t2-t3)

    result *= Am(a, x, y, Z-z, t1)*Am(a, x, y, Z-z, t2)*Am(a, x, y, Z-z, t3)*
    Am(a, x, y, Z-z, T-t1-t2-t3)

    result *= Am(a, X-x, Y-y, z, t1)*Am(a, X-x, Y-y, z, t2)*Am(a, X-x, Y-y, z, t3)*
    Am(a, X-x, Y-y, z, T-t1-t2-t3)

    result *= Am(a, X-x, y, Z-z, t1)*Am(a, X-x, y, Z-z, t2)*Am(a, X-x, y, Z-z, t3)*
    Am(a, X-x, y,Z-z, T-t1-t2-t3)

    result *= Am(a, x, Y-y, Z-z, t1)*Am(a, x, Y-y, Z-z, t2)*Am(a, x, Y-y, Z-z, t3)*
    Am(a, x, Y-y, Z-z, T-t1-t2-t3)

    result *= Am(a,X-x, Y-y, Z-z, t1)*Am(a, X-x, Y-y, Z-z, t2)*Am(a, X-x, Y-y, Z-z, t3)*
    Am(a, X-x,Y-y, Z-z, T-t1-t2-t3)

    # The following line incorporates the "embedding map", which is actually a combinatorial factor in
    # our case. It is specific to the case of coarse graining 32 hypercubes into two hypercubes.

    result *= Cemb(X*Z,Y*Z,X*Y) * Cemb(X*(t1+t2),Y*(t1+t2),X*Y) * Cemb(X*(t1+t2),Z*(t1+t2),X*Z) *
    Cemb(Y*(t1+t2),Z*(t1+t2),Y*Z)

    result *= Cemb(X*Z,Y*Z,X*Y) * Cemb(X*(T-t1-t2),Y*(T-t1-t2),X*Y) * Cemb(X*(T-t1-t2),Z*(T-t1-t2),X*Z) *
    Cemb(Y*(T-t1-t2),Z*(T-t1-t2),Y*Z)

end
```

AppData/Local/Julia-0.5. × Quantum-Cuboid-renorr × Quantum-Cuboid-renorr ×

localhost:8888/notebooks/AppData/Local/Julia-0.5.2/Quantum-Cuboid-renormalization.ipynb

jupyter Quantum-Cuboid-renormalization Last Checkpoint: 8 hours ago (autosaved)

File Edit View Insert Cell Kernel Help Julia 0.5.0

Markdown CellToolbar

```
end
```

Write a function to compute the volume of a 4D hypercuboid

In [ ]:

Write a random sampling algorithm to compute Z, < V1 > and < \Delta V1^2 > for 1D integral.

In [ ]:

Generalize the same idea to the 6D integral

In [ ]:

Is random sampling working well for the 6D integral?

Write an importance sampling algorithm. Use the amplitude itself as "weights".

In [ ]:

Generalize this to the 6D integral.

In [ ]:

DEU 15:51 21.06.2017

Actual amplitudes

- Look at file `Quantum-Cuboid-renormalization`.
- Hypercuboid amplitude: $\mathcal{A}$ as `Am(...)`
- Definition of integrands:
  - $\mathcal{A}_2$ as `Am2(...)`
  - $\mathcal{A}_{32}$ as `Am32(...)`

WarmUp: Volume

Write a function returning the volume of a 4D cuboid.

Question: How can we perform the integrations numerically?

Apply simple Monte Carlo methods.

- Random sampling
- Importance sampling

Observables: Volume of one hypercuboid

$$Z_2 = \int dt_1 \, \mathcal{A}_2(X, Y, Z, T, t_1)$$

$$Z_{32} = \int dx_1 \, dy_1 \, dz_1 \, dt_1 \, dt_2 \, dt_3 \, \mathcal{A}_{32}(X, Y, Z, T, x_1, y_1, z_1, t_1, t_2, t_3)$$

- What is an interesting observable?

Exercise 2: Volume of one hypercuboid

Define the observable 'volume of one hypercuboid' $\mathcal{V}_1$ (half of the lattice)?

Do you have an expectation for $\langle \mathcal{V}_1 \rangle$?

Steinhaus_numerical.pdf - Adobe Acrobat Reader DC

Datei Bearbeiten Anzeige Fenster Hilfe

Start Werkzeuge Steinhaus_numeric... Anmelden

6 (6 von 9) 154%

Actual amplitudes

- Look at file `Quantum-Cuboid-renormalization`.
- Hypercuboid amplitude: $\mathcal{A}$ as `Am(...)`
- Definition of integrands:
  - $\mathcal{A}_2$ as `Am2(...)`
  - $\mathcal{A}_{32}$ as `Am32(...)`

WarmUp: Volume

Write a function returning the volume of a 4D cuboid.

Question: How can we perform the integrations numerically?

Apply simple Monte Carlo methods.

- Random sampling
- Importance sampling

DEU 15:51 21.06.2017

Random sampling

- Simplest Monte Carlo method

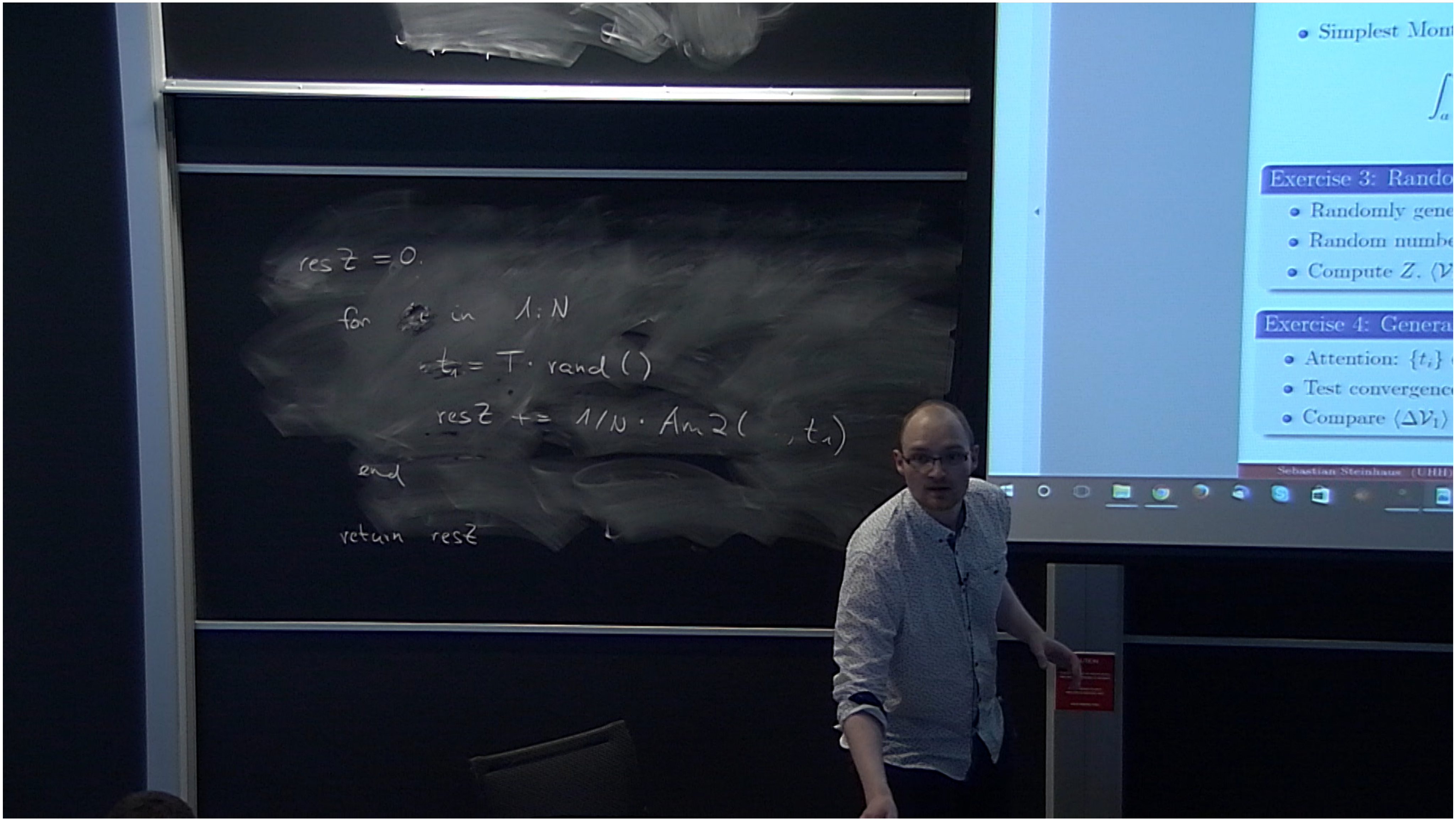
$$\int_a^b dx\, f(x) \approx \frac{(b-a)}{N} \sum_{i=1}^{N} f(x_i) \quad \text{with } x_i \in [a,b]\,\forall\, i$$

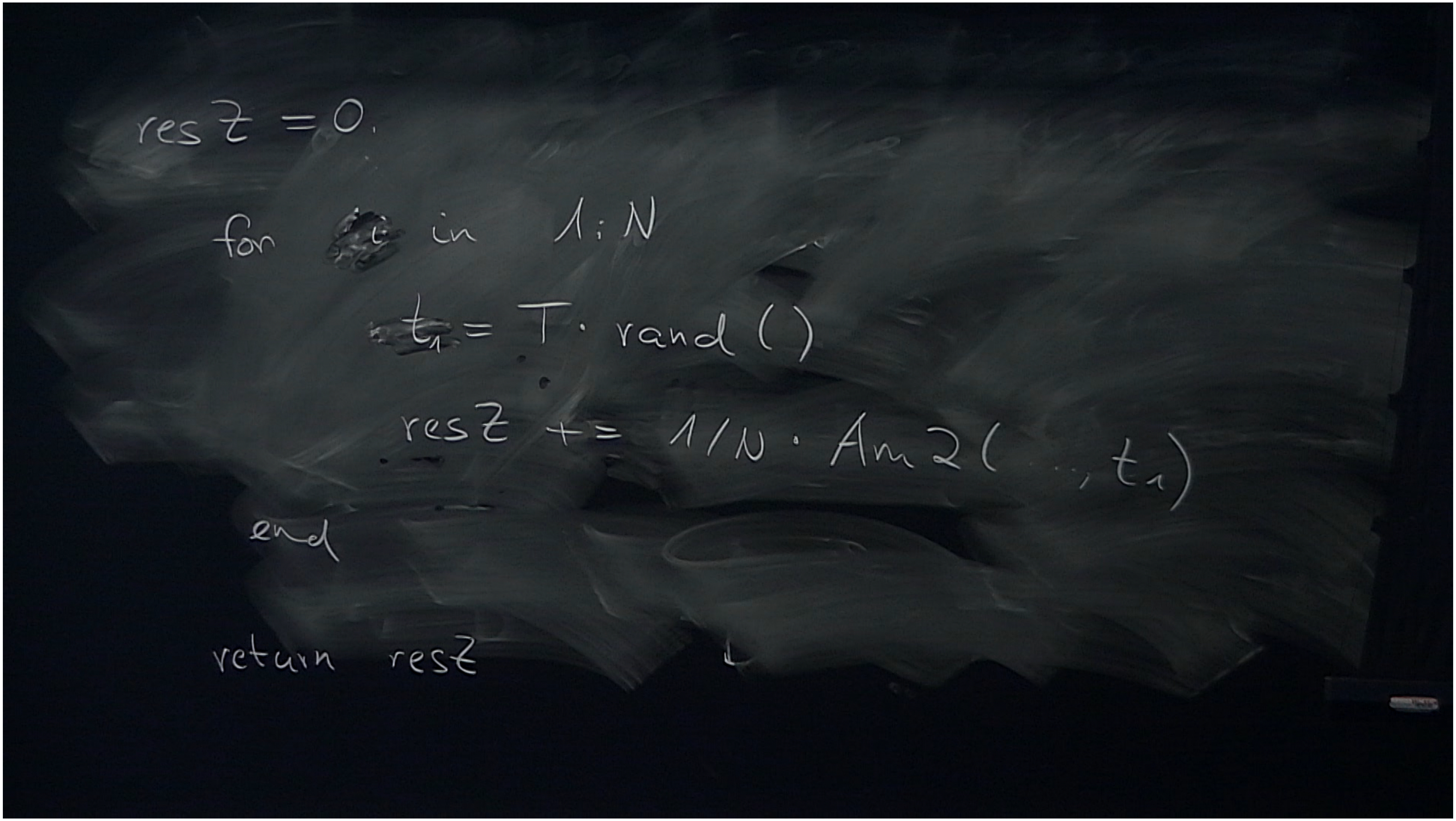
Exercise 3: Random sampling for 1D integral

- Randomly generate N samples for $t_1 \in [0, T]$.
- Random number in Julia: `rand()` (between `0` and `1`)
- Compute Z, $\langle \mathcal{V}_1 \rangle$ and $\langle \Delta\mathcal{V}_1 \rangle = \langle \mathcal{V}_1^2 \rangle - \langle \mathcal{V}_1 \rangle^2$ for various α.

Exercise 4: Generalize to 6D integral

- Attention: $\{t_i\} \in [0, T]$, but $\sum_{i=1}^4 t_i = T$.
- Test convergence of the integral.
- Compare $\langle \Delta\mathcal{V}_1 \rangle$ in both cases.


resZ = 0.
for in 1:N
t1 = T · rand()
resZ += 1/N · Am2(, t1)
end
return resZ
Simplest Mon
Exercise 3: Rando
Randomly gene
Random numbe
Compute Z.
Exercise 4: Genera
Attention:
Test convergenc
Compare
Sebastian Steinhaus (UHH)


resZ = 0.
for i in 1:N
t1 = T · rand()
resZ += 1/N · Am2(..., t1)
end
return resZ