Chapter 3 Understanding the PDE State Definition U(τ)

Equation Under Discussion

The PDE-level state definition used in Guide III is:

$$U(\tau) = (X\_i(x,\tau), g\_ij(x,\tau), N\_eff(\tau))$$

This expression defines the canonical state at the PDE-expanded level.

It does not yet give the evolution equations.

It tells us what the evolving state contains before the PDE engines are written explicitly.

In Guide I, the canonical state was written in a more readable form:

$$U(\tau) = \{X(\tau), g(\tau), N\_eff(\tau)\}$$

Guide III expands this readable canonical bundle into a more detailed PDE-level state structure.

The state-side component becomes:

$$X\_i(x,\tau)$$

The geometry-side component becomes:

$$g\_ij(x,\tau)$$

The effective degree-of-freedom component remains:

$$N\_eff(\tau)$$

This chapter explains why the state is written this way.

Opening Question

Why does Guide III write the canonical state as:

$$U(\tau) = (X\_i(x,\tau), g\_ij(x,\tau), N\_eff(\tau))$$

instead of the simpler form used in Guide I?

Guide I introduced the canonical state in a reader-friendly way:

$$U(\tau) = \{X(\tau), g(\tau), N\_eff(\tau)\}$$

That form was sufficient for the Human-Readable Gateway Form.

It allowed the reader to understand that CUWF evolution involves three major components:

a state-side component,

a geometry-side component,

and an effective degree-of-freedom component.

At that level, the purpose was conceptual clarity.

The reader did not yet need to know how each component varied across a domain.

The reader did not yet need indices.

The reader did not yet need local field structure.

The reader did not yet need PDE notation.

Guide III has a different purpose.

Guide III is not only introducing the idea of canonical evolution.

It is opening the operational engine.

To do that, the state must be written with enough detail to support explicit evolution equations.

This is why the canonical state is now written as:

$$U(\tau) = (X_i(x,\tau), g_{ij}(x,\tau), N_{eff}(\tau))$$

The notation becomes heavier because the task becomes heavier.

Guide I needed a readable bundle.

Guide III needs a state definition that can support PDE-level evolution.

Opening Perspective

The expression:

$$U(\tau) = (X_i(x,\tau), g_{ij}(x,\tau), N_{eff}(\tau))$$

should be understood as a state definition, not as an evolution equation.

It does not yet say how X_i changes.

It does not yet say how g_{ij} changes.

It does not yet say how N_{eff} changes.

It simply identifies the main components that will evolve in the PDE-expanded form.

This distinction is important.

A state definition answers the question:

What is the system made of at a given evolution order?

An evolution equation answers the question:

How does that system change from one evolution order to the next?

Chapter 3 is about the first question.

Later chapters will address the second question.

The PDE-level state definition tells us that, at evolution order τ, the canonical state contains:

$X_i(x,\tau)$, the state-side or collapse-field component,

$g_{ij}(x,\tau)$, the geometry-side or metric-like component,

and

$N_{eff}(\tau)$, the effective degree-of-freedom component.

The notation also tells us that the state has internal structure.

It is not merely a single variable.

It contains field-like quantities over a domain.

It contains indexed components.

It contains geometry-like relational structure.

It contains an effective global measure of structural complexity.

This is why the PDE-expanded form begins with a more detailed state definition.

Physical Motivation

The physical motivation behind the PDE-level state definition is that CUWF does not treat reality as a single featureless object.

Reality is structured.

It has state-side content.

It has geometry-side organization.

It has effective structural complexity.

It has coupling structure.

It evolves across evolution order.

If we want to describe this evolution operationally, we need a state definition that can carry this structure.

The simple notation:

$$U(\tau) = \{X(\tau), g(\tau), N\_eff(\tau)\}$$

is useful as a conceptual doorway.

But it hides the fact that X may have many components.

It hides the fact that X may vary across a domain.

It hides the fact that geometry is not merely one scalar quantity.

It hides the fact that geometry is relational and tensor-like.

It hides the fact that effective degrees of freedom may summarize the activity of the whole system rather than act as a local field.

The PDE-level state definition begins to reveal these differences.

It says that X should be treated as a field-like component:

$$X\_i(x,\tau)$$

It says that geometry should be treated as a metric-like or tensor-like structure:

$$g\_ij(x,\tau)$$

It says that effective structural complexity should be treated as an evolution-order dependent effective quantity:

$$N\_eff(\tau)$$

This makes the state definition more precise.

It prepares CUWF for explicit evolution equations.

Core Concept

The core concept of this chapter is the following:

Guide III does not change what $U(\tau)$ means. It increases the resolution of what $U(\tau)$ contains.

In Guide I, $U(\tau)$ was written as:

$$U(\tau) = \{X(\tau), g(\tau), N_{eff}(\tau)\}$$

This was a readable canonical bundle.

In Guide III, the same canonical idea is written at PDE level:

$$U(\tau) = (X_i(x,\tau), g_{ij}(x,\tau), N_{eff}(\tau))$$

This is not a different theory.

It is a higher-resolution representation.

The symbol $U(\tau)$ still refers to the canonical state at evolution order τ.

But now the internal components of that state are written with enough detail to support PDE-level analysis.

The state-side part becomes field-like.

The geometry-side part becomes tensor-like.

The effective degree-of-freedom part becomes an effective global flow.

Thus, the transition from Guide I to Guide III is not a change from one state to another.

It is a change from a readable state notation to an operational state notation.

Why U Is Written as $U(\tau)$, Not $U(x)$

A common question arises immediately:

If X_i and g_{ij} depend on x, why is the whole state written as $U(\tau)$ rather than $U(x)$?

The answer is that $U(\tau)$ represents the whole canonical state at evolution order τ.

It is not merely the value of the system at one spatial point.

The coordinate x labels positions inside the field configuration.

The parameter τ labels the evolution order of the whole state.

This distinction is essential.

When CUWF writes:

$$X_i(x,\tau)$$

it refers to a field-like component distributed across the domain.

When CUWF writes:

$$g_{ij}(x,\tau)$$

it refers to a geometry-like structure distributed across the domain.

But when CUWF writes:

$$U(\tau)$$

it refers to the whole configuration at evolution order τ.

In other words:

x labels positions inside the state.

τ labels the evolution of the whole state.

Therefore, $U(\tau)$ should not be read as a local value at one point.

It should be read as the full canonical configuration at a given evolution order.

The state contains x-dependent fields, but the state itself is indexed by τ because the whole configuration evolves through τ.

Note on the Meaning of X_i(x,τ), g_ij(x,τ), and N_eff(τ)

The notation:

$$U(\tau) = (X_i(x,\tau), g_{ij}(x,\tau), N_{eff}(\tau))$$

should be read carefully.

It does not mean that U is only a value at one spatial point x.

Rather, $U(\tau)$ represents the whole PDE-level canonical state at evolution order τ.

The coordinate x labels positions inside the field configuration.

The parameter τ labels the evolution order of the whole state.

The quantity $X_i(x,\tau)$ represents the state-side or collapse-field component.

Here, x labels position in the domain, while i labels the internal component of the X field.

Thus, $X_i(x,\tau)$ means the i-th component of the state-side field at position x and evolution order τ.

The quantity $g_{ij}(x,\tau)$ represents the geometry-side or metric-like structure.

Here, the two indices i and j indicate that g is tensor-like.

It does not merely describe one component of a field.

It describes relational geometric structure between directions, components, or degrees of variation.

This is why g carries two indices rather than one.

The quantity $N_{eff}(\tau)$ is different.

It is not written as $N_{eff}(x,\tau)$ in this guide because it is not treated as a local field over x.

Instead, N_eff(τ) is treated as an effective global measure of the dynamically active degrees of freedom of the system at evolution order τ.

This does not mean that N_eff is disconnected from the field structure.

Rather, N_eff summarizes the effective structural complexity of the whole system after coupling, renormalization, and regime-dependent activity are taken into account.

In short:

x labels positions inside the field configuration.

i labels a component of the state-side field X.

i and j together label the tensor-like geometric relation in g.

τ labels the evolution order of the whole canonical state.

N_eff(τ) represents an effective global structural quantity, not a local field at one point.

Understanding X_i(x,τ)

The first component of the PDE-level state is:

$$X_i(x,\tau)$$

This is the state-side or collapse-field component.

The symbol X already appeared in Guide I as the state function.

At the Guide I level, it was written simply as:

$$X(\tau)$$

This was sufficient for the Human-Readable Gateway Form.

However, the PDE-expanded form must describe how the state-side component evolves in a more detailed way.

For this reason, X becomes:

$$X_i(x,\tau)$$

The coordinate x indicates that X is distributed across a domain.

The parameter τ indicates that X evolves across evolution order.

The index i indicates that X may have multiple internal components.

Thus, $X_i(x,\tau)$ should be read as:

the i-th component of the state-side field at position x and evolution order τ.

This does not mean that we are only looking at one point.

It means that the field has a value at each position x, and the full field configuration belongs to $U(\tau)$.

This is important because collapse-field evolution is not merely a change of one number.

It is a change of a distributed state-side structure.

The PDE-expanded form must therefore be able to describe how X_i changes across the domain and across evolution order.

Why X Has an Index i

The index i in:

$$X_i(x,\tau)$$

is a component index.

It should not be confused with the spatial coordinate x.

The coordinate x tells us where in the domain the field is being evaluated.

The index i tells us which internal component of the field is being described.

For example, a field may have multiple components.

One component may represent one state-side direction of variation.

Another component may represent another.

The notation X_i allows CUWF to write this component structure compactly.

This is common in field-like systems.

A field does not always have only one scalar value at each point.

It may have several components.

The index i allows the PDE-expanded form to describe those components without writing a separate symbol for each one.

In CUWF, this matters because collapse-field evolution may not be one-dimensional or featureless.

It may involve multiple state-side components evolving together.

The notation X_i(x,τ) keeps this possibility open.

Understanding g_ij(x,τ)

The second component of the PDE-level state is:

$$g_{ij}(x,\tau)$$

This is the geometry-side or metric-like component.

In Guide I, geometry was written simply as:

$$g(\tau)$$

This was enough for a conceptual explanation.

But geometry is not merely a single number.

At the PDE-expanded level, geometry must be able to describe relational structure.

It must be able to encode how directions, components, or degrees of variation are related.

For this reason, the geometry term is written with two indices:

$$g_{ij}(x,\tau)$$

The coordinate x indicates that geometry may vary across the domain.

The parameter τ indicates that geometry evolves across evolution order.

The indices i and j indicate that geometry is tensor-like or metric-like.

It describes a relation between directions or components, not merely one isolated component.

This is why g_ij carries two indices.

It is not just “the i-th geometry component.”

It is a relational object.

It tells us how the i-direction and j-direction are geometrically connected or weighted at position x and evolution order τ.

This makes g_ij fundamentally different from X_i.

X_i represents field components.

g_ij represents geometric relations among components or directions.

Why g Has Two Indices i and j

The two indices in:

$g_{ij}(x,\tau)$

reflect the metric-like role of geometry.

A metric-like object does not merely assign a value to one component.

It describes how pairs of directions relate.

In ordinary geometry, a metric tells us how distances, angles, or inner products are measured.

In CUWF, g_{ij} plays a geometry-side role within the canonical state.

It carries information about the relational structure through which state evolution is geometrically organized.

This is why one index is not enough.

A single index would label one component.

Two indices allow the object to describe relations between components or directions.

In matrix-like form, g_{ij} may be imagined as a structured array:

$g_{11}, g_{12}, g_{13}, \ldots$

$g_{21}, g_{22}, g_{23}, \ldots$

$g_{31}, g_{32}, g_{33}, \ldots$

and so on.

The diagonal components may describe self-relations or direction-specific weights.

The off-diagonal components may describe cross-relations or couplings between directions.

The exact interpretation depends on the regime and formal development, but the key point is general:

g_{ij} is tensor-like because geometry is relational.

This is why the geometry component of the PDE-level state is written as $g_{ij}(x,\tau)$.

Understanding N_eff(τ)

The third component of the PDE-level state is:

$$N_{eff}(\tau)$$

This is the effective degree-of-freedom component.

In Guide I, N_eff(τ) was introduced as part of the canonical state:

$$U(\tau) = \{X(\tau), g(\tau), N_{eff}(\tau)\}$$

In Guide III, N_eff remains:

$$N_{eff}(\tau)$$

It is not written as:

$$N_{eff}(x,\tau)$$

This is intentional.

In the present guide, N_eff is not treated as a local field over x.

Instead, it is treated as an effective global measure of the dynamically active degrees of freedom of the system.

It summarizes structural complexity at the level of the whole system or regime.

This does not mean that N_eff is unrelated to the local field structure.

It may depend on the behavior of X, g, coupling, renormalization, and regime-dependent activity.

But its role is different from X_i and g_ij.

X_i and g_ij are written as field-like or tensor-like structures over x.

N_eff is written as an effective flow over τ.

It changes as the system evolves, but it is not assigned a separate value at every point x in this formulation.

Why N_eff Is Not Written as N_eff(x,τ)

One might ask:

If X and g depend on x, why does N_eff not also depend on x?

The answer is that N_eff plays a different role.

It is not a local field component.

It is an effective measure of active structural complexity.

In many systems, the number of effective degrees of freedom is not simply a local quantity.

It is a measure of how many degrees of freedom are dynamically active after constraints, coupling, coarse-graining, renormalization, and regime selection are taken into account.

For this reason, N_eff is better understood as an effective global or regime-level quantity.

It may summarize the result of many local and nonlocal processes.

It may change as the system evolves.

It may respond to coupling and renormalization.

But it is not necessarily meaningful to assign a separate independent N_eff to each point x.

This is why the PDE-level state definition writes:

$$N_{eff}(\tau)$$

rather than:

$$N_{eff}(x,\tau)$$

This choice keeps the form conceptually controlled.

It allows N_eff to represent effective structural complexity without turning it into a separate local field.

Spatial Dependence and Evolution-Order Dependence

The expression:

$$X_i(x,\tau)$$

contains two types of dependence.

It depends on x.

It also depends on τ.

These dependencies mean different things.

The coordinate x tells us where the field is being evaluated inside the domain.

The parameter τ tells us which evolution order the field belongs to.

The same is true for:

$$g_{ij}(x,\tau)$$

The geometry structure may vary across x.

It may also evolve across τ.

This means that the PDE-expanded form is not merely describing how a number changes.

It is describing how entire field configurations change.

At each τ, there is a field configuration over x.

As τ changes, that entire configuration evolves.

This is why partial derivatives appear in later chapters.

For example:

$$\partial X\_i(x,\tau)/\partial\tau$$

asks how the X_i field changes with respect to evolution order while still being a function over x.

Similarly:

$$\partial g\_ij(x,\tau)/\partial\tau$$

asks how the geometry-side structure changes with respect to evolution order while still being distributed across the domain.

This is the PDE character of the expanded form.

It does not come from the state definition alone.

It comes from the evolution equations written for field-like components.

Why This Is a PDE-Level State Definition

The expression:

$$U(\tau) = (X\_i(x,\tau), g\_ij(x,\tau), N\_eff(\tau))$$

is called a PDE-level state definition because it prepares the system for partial differential evolution equations.

It is not itself a PDE.

It is the state structure on which PDEs will act.

The PDE character enters because X_i and g_ij depend on both x and τ.

When their τ-evolution is written, the result involves partial derivatives such as:

$$\partial X\_i(x,\tau)/\partial\tau$$

and

$$\partial g\_ij(x,\tau)/\partial\tau$$

These are partial derivatives because X_i and g_ij depend on more than one variable.

They depend on position x and evolution order τ.

The evolution equation describes how they change with respect to τ while retaining their spatial structure over x.

This is why Guide III must introduce the PDE-level state definition before writing the PDE engines in detail.

The reader must first understand what kind of object is evolving.

Only then can the evolution equations be understood.

From Guide I State Bundle to Guide III State Structure

Guide I used:

$$U(\tau) = \{X(\tau), g(\tau), N\_eff(\tau)\}$$

This form was readable and conceptually useful.

It allowed the reader to understand that CUWF evolution involves a coupled canonical state.

But Guide III requires more structure.

The PDE-expanded form must be able to describe:

field variation,

geometry variation,

component structure,

relational geometry,

effective degrees of freedom,

and evolution across τ.

Therefore, Guide III writes:

$$U(\tau) = (X\_i(x,\tau), g\_ij(x,\tau), N\_eff(\tau))$$

The transition may be summarized as follows:

$X(\tau)$ becomes $X\_i(x,\tau)$.

$g(\tau)$ becomes $g\_ij(x,\tau)$.

$N\_eff(\tau)$ remains $N\_eff(\tau)$, but is now understood as an effective global flow.

This is not a contradiction.

It is a refinement.

The Guide I notation gave the readable outline.

The Guide III notation gives the operational structure.

Relation to the Three Main Engines

The PDE-level state definition prepares the three main engines of Guide III.

The first engine evolves $X\_i(x,\tau)$.

This will be the collapse-field evolution equation.

The second engine evolves $g\_ij(x,\tau)$.

This will be the geometry or curvature evolution equation.

The third engine evolves $N\_eff(\tau)$.

This will be the renormalization / effective degree-of-freedom flow equation.

Thus, the state definition tells us what the engines will act on.

The later equations tell us how those components change.

This relationship is important.

The PDE-expanded form is not a random collection of equations.

It is organized around the components of the canonical state.

The state definition gives the object.

The PDE engines give the motion.

Together, they form the operational structure of Guide III.

The Role of Ξ_eff Relative to the State Definition

Although this chapter focuses on:

$$U(\tau) = (X_i(x,\tau), g_{ij}(x,\tau), N_{eff}(\tau))$$

it is also important to remember the role of Ξ_eff.

In Guide II, the generator functional was written as:

$$G(X, g, N_{eff}, \Xi_{eff})$$

This means that Ξ_eff belongs to the generator structure.

However, in Guide III, Ξ_eff is not treated as a fourth independent engine coordinate inside the state vector.

Instead, it functions as an effective coupling / entanglement field.

It shapes how X, g, and N_eff evolve.

It appears in the collapse-field equation through nonlocal coupling.

It appears in the geometry equation through $F_{ij}(\Xi_{eff})$.

It appears in the renormalization / DOF flow through $R(..., \Xi_{eff})$.

Therefore, Ξ_{eff} is not absent.

It is present as coupling structure inside the engine.

The state definition identifies the main evolved components.

Ξ_{eff} identifies an effective coupling structure that influences their evolution.

This distinction helps prevent a misunderstanding.

Guide III has three main engines, not four.

But those three engines are coupled through structures that include Ξ_{eff}.

Physical Interpretation

Physically, the PDE-level state definition says that CUWF evolution is not only the evolution of one abstract variable.

It is the evolution of a structured canonical state.

The state-side content is field-like.

The geometry-side structure is tensor-like.

The effective degree-of-freedom structure is global or regime-level.

The whole configuration evolves across τ.

This gives CUWF a richer operational structure.

It allows the theory to ask how collapse fields change.

It allows the theory to ask how geometry responds.

It allows the theory to ask how effective structural complexity flows.

It also allows the theory to ask how coupling structures influence these changes.

Thus, the PDE-level state definition is not merely notation.

It is the foundation for the operational engine.

Without this state definition, the later PDE equations would have no clear object to act on.

What This Chapter Does Not Do

This chapter does not yet explain the collapse-field evolution equation.

It does not yet explain the nonlocal kernel.

It does not yet explain the full role of Ξ_eff in the integral term.

It does not yet explain the geometry evolution equation.

It does not yet explain the renormalization / DOF flow.

Those tasks belong to later chapters.

The purpose of this chapter is narrower.

It explains what the PDE-level state is.

It explains why x appears.

It explains why indices i and j appear.

It explains why N_eff is written as an effective global quantity.

It prepares the reader to understand the evolution equations that follow.

Final Definition

The PDE-level state definition:

$$U(\tau) = (X\_i(x,\tau),\ g\_ij(x,\tau),\ N\_eff(\tau))$$

defines the canonical state at evolution order τ as a structured configuration composed of a state-side field component, a geometry-side tensor-like component, and an effective global degree-of-freedom component.

More simply:

U(τ) is the whole canonical state at τ.

X_i(x,τ) gives the state-side field structure.

g_ij(x,τ) gives the geometry-side relational structure.

N_eff(τ) gives the effective structural complexity of the system.

One-Sentence Summary

Guide III writes U(τ) as (X_i(x,τ), g_ij(x,τ), N_eff(τ)) because the PDE-expanded form must represent the canonical state as a field-structured, geometry-structured, and effective-degree-of-freedom structure capable of supporting explicit evolution equations.

Looking Ahead

This chapter explained the PDE-level state definition.

We saw why Guide III expands the readable Guide I state bundle into a more detailed operational structure.

We saw why x appears inside X_i and g_ij.

We saw why i labels a component of the state-side field.

We saw why i and j together label tensor-like geometric relations.

We saw why N_eff is treated as an effective global flow rather than a local field.

The next chapter will begin opening the first major engine of the PDE-expanded form:

the collapse-field evolution equation.

It will ask how $X_i(x,\tau)$ changes with respect to τ, why local descent appears, and why nonlocal coupling must enter the state-side evolution.