Compositionality of the Runge-Kutta Method

Abstract

In Spivak [Spi16], dynamical systems are described in terms of their inputs and outputs in a pictorial way using an operad of wiring diagrams. Each dynamical system is a box with certain inputs and outputs, and multiple dynamical systems are linked together using wiring diagrams, which describe how the outputs of one dynamical system to the inputs of another. By describing dynamical systems in this way, we can decompose a large dynamical system as a collection of smaller, simpler dynamical systems linked together. Of course, this decomposition is only useful if we can work with these smaller, simpler dynamical systems instead of the larger one. In his paper, Spivak shows that we can perform Euler’s method on these smaller systems and still get the same results as working on the larger one. In this paper, we extend his results to prove that we can do something similar with the Runge-Kutta method. However, we need to modify the framework used in Spivak’s paper to account for the fact that the Runge-Kutta method requires multiple steps, unlike Euler’s method. To better describe these systems, we define wiring diagrams as objects of a double-category and dynamical systems in terms of double functors, giving a categorical description of this approximation method.

1 Introduction

Dynamical systems are everywhere, from the cells in our bodies to the computers which we work on. Each of these machines take inputs, modify an internal state, and then give an output. For a computer, the inputs could be user inputs on a keyboard, which then modify the binary state held in the hard drive and memory, and the output could be the display on the monitor. In addition, by using the outputs of and providing the inputs for other dynamical systems, dynamical systems can become interconnected through a network which as a whole represents one larger dynamical system, such as the internet, which is a dynamical system composed of many interconnected computers. When working with such a network, often it would be much easier to work with the individual dynamical systems and then combine the results instead of composing all the dynamical systems first and then working with that large system.

In Spivak’s work [Spi16], he provides a framework for describing dynamical systems and their interconnections through wiring diagrams. A dynamical system with input set $A$ and output set $B$ is described as a tuple $(S, f^{rdt}, f^{upd})$, with $S$ the state set, $f^{rdt} : S \rightarrow B$ a readout function turning the state to an output, and $f^{upd} : A \times S \rightarrow S$ an update function changing the state depending on the input. For a discrete dynamical system, we let our state set be any set, and we define $f^{upd}$ by taking an input and state and give back the next state which should result. For a continuous dynamical system, we let our state set be a real vector space and define $f^{upd}$ so that it takes the input and state and returns the infinitesimal change (like a derivative) which should result. Often, we will describe a dynamical system as a diagram

$$A \times S \longrightarrow S \longrightarrow B$$

1
with the arrows representing the update and readout functions.

In his paper, Spivak proves that Euler’s method works in a natural way when converting continuous dynamical systems to discrete dynamical systems so that we can work with the individual dynamical systems before composing instead of having to work with the much larger combined dynamical system. We would like to extend that result to the Runge-Kutta method, but unfortunately, the discrete dynamical system format above does not capture the four-step approach that Runge-Kutta uses. In every iteration of Runge-Kutta, there are four computations performed, each depending on all the previous results of the iteration, but this iterative cycle of four steps is not captured in the original definition of a discrete dynamical system, as it only goes step by step and does not remember any states besides the previous one. In addition, Spivak describes his continuous dynamical systems using the manifold structure of a real vector space, but for Runge-Kutta, it is the vector space properties which are more important. As a result, we need to describe dynamical systems differently to more accurately capture this information. In particular, we introduce a second type of morphism which describes a dynamical system working in line with another dynamical system. An iterative four-step dynamical system can be thought of as a dynamical system which lines up with the cyclic four-step dynamical system \( C_4 \) shown below.

\[
\begin{array}{c}
1 \rightarrow 2 \\
\uparrow \\
4 \leftarrow 3
\end{array}
\]

To incorporate both types of morphisms, we describe our dynamical systems using double categories and double functors and describe Runge-Kutta as a double transformation. To capture how dynamical systems can be combined, we also impose a monoidal product to describe dynamical systems being put side-by-side without interacting with each other, similar to what is described in Spivak’s work, and we prove that our double functors and double transformations preserve this monoidal structure.

The next section, Section 2, defines what double categories, double functors, and double transformations are, along with their monoidal versions. To understand these definitions, however, the reader needs to know the definitions of categories, functors, and natural transformations, in addition to the monoidal forms of each of these concepts. Spivak’s paper describes continuous systems using a functor \( CS \) and discrete dynamical systems using a functor \( DS \) from the category of wiring diagrams to the category of sets. In Section 3, we define the double category of wiring diagrams \( \mathcal{W} \) and describe our continuous dynamical systems through a double functor \( CS : \mathcal{W} \to \text{Prof} \) and our four-step dynamical systems through a double functor \( DS/C_4 : \mathcal{W} \to \text{Prof} \), where \( \text{Prof} \) is the double category of profunctors. We then describe a monoidal product to represent combining dynamical systems together and show that our double category and double functors preserve this product. Finally, in Section 4, we describe Runge-Kutta as a double transformation between \( CS \) and \( DS/C_4 \) and prove the following theorem.

**Theorem 1.** The Runge-Kutta method is a monoidal double transformation between \( CS \) and \( DS/C_4 \).
2 Definitions

We begin by defining double categories, double functors, and double transformations, using the works of Shulman [Shu10][Shu09]. As mentioned in the introduction, double categories allow us to define two types of morphisms for our objects.

Definition 2.1. A (pseudo) double category $D$ consists of a “category of objects $D_0$” and a “category of arrows” $D_1$, with structure functors

$$U : D_0 \to D_1$$

$$L, R : D_1 \to D_0$$

$$\circ : D_1 \times_{D_0} D_1 \to D_1$$

where the pullback is over

$$D_1 \xrightarrow{R} D_0 \xleftarrow{L} D_1$$

These structure functors must satisfy the relations

$$L(U_A) = A \quad R(U_A) = A$$

$$L(M \circ N) = L(M) \quad R(M \circ N) = R(N)$$

and are equipped with natural transformations

$$a : (M \circ N) \circ P \to M \circ (N \circ P)$$

$$l : U_{L(M)} \circ M \to M$$

$$r : M \circ U_{R(M)} \to M$$

such that $L(a)$, $R(a)$, $L(l)$, $R(l)$, $L(r)$, $R(r)$ are all identities and the following diagrams commute:

(Triangle)

(Triangle)

(Pentagon)

(Pentagon)
Definition 2.2. Let \( \mathcal{D} \) and \( \mathcal{E} \) be double categories. A lax double functor \( F : \mathcal{D} \to \mathcal{E} \) consists of the following:

- Functors \( F_0 : \mathcal{D}_0 \to \mathcal{E}_0 \) and \( F_1 : \mathcal{D}_1 \to \mathcal{E}_1 \) such that \( L \circ F_1 = F_0 \circ L \) and \( R \circ F_1 = F_0 \circ R \).

- Natural Transformations \( F_0 : F_1M \circ F_1N \to F_1(M \circ N) \) and \( F_U : U_{F_0} \to F_1(U_L) \), whose components are globular, and for which the following diagrams commute.

\[
\begin{align*}
(F_1(x) \circ F_1(y)) \circ F_1(z) & \xrightarrow{a} F_1(x) \circ (F_1(y) \circ F_1(z)) \\
F_1(x \circ y) \circ F_1(z) & \xrightarrow{id \circ F_0} F_1(x) \circ F_1(y \circ z) \\
F_1((x \circ y) \circ z) & \xrightarrow{F_1(a)} F_1(x \circ (y \circ z))
\end{align*}
\]

(associativity)

\[
\begin{align*}
U_{L \circ F_1 x} \circ F_1(x) & \xrightarrow{F_{A \circ id}} F_1(U_{L x}) \circ F_1 x \\
F_1(x) & \xrightarrow{F_1(l)} F_1(U_{L x} \circ x) \\
F_1(x) & \xleftarrow{F_1(r)} F_1(x \circ U_{R x})
\end{align*}
\]

If \( F_U \) is an isomorphism, then \( F \) is called a normal double functor.

Definition 2.3. A double transformation between two lax double functors \( \alpha : F \Rightarrow G : \mathcal{D} \to \mathcal{E} \) consists of natural transformations \( \alpha_0 : F_0 \Rightarrow G_0 : \mathcal{D}_0 \to \mathcal{E}_0 \) and \( \alpha_1 : F_1 \Rightarrow G_1 : \mathcal{D}_1 \to \mathcal{E}_1 \) such that \( L(\alpha_M) = \alpha_{LM} \) and \( R(\alpha_M) = \alpha_{RM} \) and

\[
\begin{align*}
FA & \xrightarrow{F_{A \circ M \circ N}} F_{A \circ (M \circ N)} \\
FA & \xrightarrow{\alpha_{A \circ M \circ N}} \alpha_{A \circ (M \circ N)} \\
GA & \xrightarrow{G_{(M \circ N)}} GC
\end{align*}
\]

\[
\begin{align*}
FA & \xleftarrow{U_{FA}} FA \\
FA & \xrightarrow{F_{U \circ A}} FA \\
GA & \xrightleftharpoons{G_{U \circ A}} GA
\end{align*}
\]

\[
\begin{align*}
FA & \xrightarrow{F_{U \circ A}} FA \\
FA & \xrightarrow{F(U_{A \circ N})} FA \\
GA & \xrightarrow{G_{U_{A \circ N}}} GA
\end{align*}
\]
We now describe the monoidal versions of these definitions. The definitions for the monoidal double functor and monoidal double transformation are the same as the definitions Shulman uses for monoidal framed bicategories.

**Definition 2.4.** A *monoidal double category* is a double category equipped with functors $\otimes : D \times D$ and $I : \ast \to D$, and invertible transformations
\[
\otimes \circ (\text{Id} \times \otimes) \cong \otimes \circ (\otimes \times \text{Id}) \\
\otimes \circ (\text{Id} \times I) \cong \text{Id} \\
\otimes \circ (I \times \text{Id}) \cong \text{Id}
\]
satisfying the usual axioms.

Unpacking this definition more explicitly, we see that a monoidal double category is a double category together with the following structure.

- $D_0$ and $D_1$ are both monoidal categories.
- If $I$ is the monoidal unit of $D_0$, then $U_I$ is the monoidal unit of $D_1$.
- The functors $L$ and $R$ are strict monoidal, i.e. $L(M \otimes N) = LM \otimes LN$ and $R(M \otimes N) = RM \otimes RN$ and $L$ and $R$ also preserve the associativity and unit constraints.
- We have globular isomorphisms
  \[
x : (M_1 \otimes N_1) \circ (M_2 \otimes N_2) \cong (M_1 \circ M_2) \otimes (N_1 \circ N_2)
\]
and
  \[
u : U_{A \otimes B} \cong (U_A \otimes U_B)
\]
such that the following diagrams commute:
Compositionality of the Runge-Kutta method

Timothy Ngotiaoco

(these arise from the constraint data from the pseudo double functor $\otimes$).

- The following diagrams commute, expressing that the associativity isomorphism for $\otimes$ is a transformation of double categories.

$$
\begin{array}{c}
((M_1 \otimes N_1) \otimes P_1) \circ ((M_2 \otimes N_2) \otimes P_2) \\
\downarrow \\
((M_1 \otimes N_1) \circ (M_2 \otimes N_2) \otimes (P_1 \circ P_2)) \\
\downarrow \\
((M_1 \otimes M_2) \otimes (N_1 \otimes N_2)) \otimes (P_1 \circ P_2) \\
\downarrow \\
(M_1 \circ M_2) \otimes ((N_1 \otimes N_2) \otimes (P_1 \circ P_2))
\end{array}
\begin{array}{c}
U_{(A \otimes B) \otimes C} \\
\downarrow \\
U_{A \otimes B} \otimes U_C \\
\downarrow \\
(U_A \otimes U_B) \otimes U_C
\end{array}
\begin{array}{c}
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow
\end{array}
\begin{array}{c}
U_{A \otimes (B \otimes C)} \\
\downarrow \\
U_A \otimes U_{B \otimes C} \\
\downarrow \\
U_A \otimes (U_B \otimes U_C)
\end{array}
$$

- The following diagrams commute, expressing that the unit isomorphisms for $\otimes$ are transformations of double categories.

$$
\begin{array}{c}
(M \otimes U_I) \circ (N \otimes U_I) \\
\downarrow \\
M \otimes N
\end{array}
\begin{array}{c}
\rightarrow \\
\rightarrow
\end{array}
\begin{array}{c}
(M \circ N) \otimes (U_I \circ U_I) \\
\downarrow \\
(M \circ N) \otimes U_I
\end{array}
\begin{array}{c}
U_{A \otimes I} \\
\downarrow \\
U_A
\end{array}
\begin{array}{c}
\rightarrow \\
\rightarrow
\end{array}
\begin{array}{c}
U_A \otimes U_I \\
\downarrow \\
U_A
\end{array}
\begin{array}{c}
(U_I \otimes M) \circ (U_I \otimes N) \\
\downarrow \\
M \otimes N
\end{array}
\begin{array}{c}
\rightarrow \\
\rightarrow
\end{array}
\begin{array}{c}
(U_I \otimes U_I) \otimes (M \circ N) \\
\downarrow \\
U_I \otimes (M \circ N)
\end{array}
\begin{array}{c}
U_{I \otimes A} \\
\downarrow \\
U_A
\end{array}
\begin{array}{c}
\rightarrow \\
\rightarrow
\end{array}
\begin{array}{c}
U_I \otimes U_A \\
\downarrow \\
U_A
\end{array}
$$

A good example of a monoidal double category is Prof, the double category of categories and profunctors. One can check that imposing the Cartesian product as the monoidal product gives us a monoidal double category.

**Definition 2.5.** A lax monoidal double functor between monoidal double categories $\mathcal{D}, \mathcal{E}$ consists of the following structure and properties.

1. A lax double functor $F : \mathcal{D} \to \mathcal{E}$.
2. The structure of a lax monoidal functor on $F_0$ and $F_1$.
3. Equalities $LF_1 = F_0L$ and $RF_1 = F_0R$ of lax monoidal functors.
The composition constraints for the lax double functor $F$ are monoidal natural transformations.

**Definition 2.6.** A monoidal double transformation is a double transformation such that $\alpha_0$ and $\alpha_1$ are monoidal natural transformations.

### 3 Dynamical Systems as a Double Functor

We can now describe our wiring boxes and wiring diagrams as a double category, allowing us to describe one morphism using wiring diagrams and another morphism using commuting diagrams which describe a wiring box acting in step with another.

**Definition 3.1.** We define $\mathcal{W}$, the double category of wiring diagrams, by the following categories $\mathcal{W}_0$ and $\mathcal{W}_1$. $\mathcal{W}_0$ is the category of wiring diagrams. The objects are ordered pairs $(A, B)$ with $A, B \in \text{Set}$ and the morphisms are wiring diagrams.

$$(A, B) \xrightarrow{\varphi} (C, D)$$

$\varphi^{\text{in}} : C \times B \to A$

$\varphi^{\text{out}} : B \to D$

The composition of wiring diagrams $\varphi$ and $\psi$

$$(A, B) \xrightarrow{\varphi} (C, D) \xrightarrow{\psi} (E, F)$$

is the wiring diagram with $(\psi \circ \varphi)^{\text{in}} : E \times B \to A$ given by

$$E \times B \xrightarrow{id \times \Delta} E \times B \times B \xrightarrow{id \times \varphi^{\text{out}} \times id} E \times D \times B \xrightarrow{\psi^{\text{in}} \times id} C \times B \xrightarrow{\varphi^{\text{in}}} A$$

and $(\psi \circ \varphi)^{\text{out}} : B \to F$ given by

$$B \xrightarrow{\varphi^{\text{out}}} D \xrightarrow{\psi^{\text{out}}} F$$

We define $\mathcal{W}_1$ to be the following category. The objects are ordered pairs $(f, g)$ where $f : A \to A'$ and $g : B \to B'$ with $L(f, g) = (A, B)$ and $R(f, g) = (A', B')$. We define a morphism $(f, g)$ to $(f', g')$ to be a pair of wiring diagrams $(\varphi_1, \varphi_2)$ such that the following diagrams commute.
The resulting 2-cell can be described by the following box.

\[
(A_1, B_1) \xrightarrow{(f,g)} (A_2, B_2)
\]

Then we let the composition of two morphisms in \( \mathcal{W}_1 \) be given by the following equality.

\[
(A_1', B_1') \xrightarrow{(f',g')} (A_2', B_2') \quad \text{and} \quad (A_1, B_1) \xrightarrow{(f,g)} (A_2, B_2) \quad \text{then} \quad (A_1'', B_1'') \xrightarrow{(f'',g'')} (A_2'', B_2'') \quad \text{such that} \quad (f'' \circ g') \circ (f \circ g) = (f'' \circ g'') \circ (f \circ g)
\]

The latter term is a valid morphism since we have the following commuting diagrams.

![Diagram](https://example.com/diagram.png)

We define horizontal composition for objects by \((f', g') \odot (f, g) = (f' \circ f, g' \circ g)\) and define horizontal composition for morphisms by taking the unique morphism having the required left and right values. One can check that the identity natural transformations work for \(a, l, r\).

Now we impose the following monoidal product to represent the combination of two wiring boxes.

\[
\left( (A_1, B_1) \xrightarrow{(f,g)} (A_2, B_2) \right) \otimes \left( (A_1', B_1') \xrightarrow{(f',g')} (A_2', B_2') \right) = (A_1 \times A_1', B_1 \times B_1') \xrightarrow{(f \times f', g \times g')} (A_2 \times A_2', B_2 \times B_2')
\]
In the second equality, the last wiring diagram $\varphi_1 \times \varphi_2$ is the product of wiring diagrams $\varphi_1$ and $\varphi_2$ and is given by the following functions.

\[
(\varphi_1 \times \varphi_2)^{out}(s_1 \times s_2) = \varphi_1^{out}(s_1) \times \varphi_2^{out}(s_2)
\]

\[
(\varphi_1 \times \varphi_2)^{in}(a_1 \times a_2, s_1 \times s_2) = \varphi_1^{in}(a_1, s_1) \times \varphi_2^{in}(a_2, s_2)
\]

Since the monoidal product of horizontal morphisms and vertical morphisms act coordinate-wise on the objects, the monoidal product of 2-cells make sense using the monoidal product of the boundaries.

We prove that $\mathcal{W}_0$ forms a monoidal category. Our product forms a functor since

\[
(\varphi_1^{out} \circ \varphi_2^{out}) \times (\psi_2^{out} \circ \psi_1^{out}) = (\varphi_2^{out} \times \psi_2^{out}) \circ (\varphi_1^{out} \times \psi_1^{out})
\]

and

\[
\varphi_1(\varphi_2(A_3, \varphi_1^{out}(B_1)), B_1) \times \psi_1(\psi_2(C_2, \psi_1^{out}(D_1)), D_1) =
\]

\[
(\varphi_1 \times \psi_1)(\varphi_2 \times \psi_2)(A_2 \times C_2, \varphi_1 \times \psi_1(B_1 \times D_1)), B_1 \times D_1
\]

The unit object is $(\ast, \ast)$ and the unit morphism is the trivial wiring diagram $(\ast, \ast) \to (\ast, \ast)$.

$\mathcal{W}_1$ also forms a monoidal category with the following product.

\[
(A_1, B_1) \xrightarrow{(f_1, g_1)} (C_1, D_1) \quad (A_1', B_1') \xrightarrow{(f_1', g_1')} (C_1', D_1')
\]

\[
(A_2, B_2) \xrightarrow{(f_2, g_2)} (C_2, D_2) \quad (A_2', B_2') \xrightarrow{(f_2', g_2')} (C_2', D_2')
\]

\[
(A_1 \times A_1', B_1 \times B_1') \xrightarrow{(f_1 \times f_1', g_1 \times g_1')} (C_1 \times C_1', D_1 \times D_1')
\]

\[
(A_2 \times A_2', B_2 \times B_2') \xrightarrow{(f_2 \times f_2', g_2 \times g_2')} (C_2 \times C_2', D_2 \times D_2')
\]

Our product forms a functor since everything works coordinate-wise, and we can take the unit object to be $(\ast, \ast) \xrightarrow{(1, 1)} (\ast, \ast)$ and the unit morphism to be the box with the unit objects at the top and bottom and unit wiring diagrams at the left and right. These products on $\mathcal{W}_0$ and $\mathcal{W}_1$ affect $L$ and $R$ in the same way so $L$ and $R$ are strict monoidal functors. Finally, we can take $x$ and $u$ to be identities and the remaining conditions can be easily checked.

As mentioned in the introduction, for continuous dynamical systems, our state set $S$ is a real vector space. We define a morphism of continuous dynamical systems to be a commuting diagram

\[
A \times S \xrightarrow{\text{id} \times f_{in}} A \times T \xrightarrow{g_{in}} A \xrightarrow{f_{out}} S \xrightarrow{m} T \xrightarrow{g_{out}} B
\]
such that \( m \) is a linear map. If \( m \) is an isomorphism, then we can say that the dynamical systems are isomorphic, and from this, we can split dynamical systems with inputs \( A \) and \( B \) into isomorphism classes.

We define the category \( \text{CS}(A, B) \), with the objects being the isomorphism classes of dynamical systems

\[
A \times S \xrightarrow{\text{fin}} S \xrightarrow{\text{fout}} B
\]

where \( S \) is a real vector space and given representatives of two isomorphism classes, say with state sets \( S \) and \( T \), we define a morphism to be a diagram

\[
\begin{array}{ccc}
A \times S & \xrightarrow{\text{fin}} & S & \xrightarrow{\text{fout}} & B \\
\downarrow{\text{id}} & & \downarrow{m} & & \downarrow{\text{id}} \\
A \times T & \xrightarrow{g\text{in}} & T & \xrightarrow{g\text{out}} & B
\end{array}
\]

with \( m : S \rightarrow T \) a linear map such that the diagram commutes.

We can now define the map \( \text{CS} : \mathcal{W} \rightarrow \text{Prof} \) to represent our continuous dynamical systems.

We define \( \text{CS}_0 \) by sending \((A, B) \in \mathcal{W}_0\) to the category \( \text{CS}(A, B) \). To show that this is a functor, we need to show that each wiring diagram \( \phi : (A, B) \rightarrow (A', B') \) gives a functor \( \text{CS}(A, B) \rightarrow \text{CS}(A', B') \).

The wiring diagram \( \phi \) act on morphisms in \( \text{CS}(A, B) \) by the following commuting diagrams.

\[
\begin{array}{ccccccc}
A' \times S & \xrightarrow{id \times \Delta} & A' \times S \times S & \xrightarrow{id \times \text{out} \times id} & A' \times B \times S & \xrightarrow{q'\text{in} \times id} & A \times S & \xrightarrow{\text{in}} & S & \xrightarrow{\text{in}} & S & \xrightarrow{\text{in}} & S \\
\downarrow{id \times m} & & \downarrow{id \times m \times m} & & \downarrow{id \times m \times m} & & \downarrow{id \times m} & & \downarrow{m} & & \downarrow{m} & & \downarrow{\text{id}} & & \downarrow{\text{id}} \\
A' \times S' & \xrightarrow{id \times \Delta} & A' \times S' \times S' & \xrightarrow{id \times \text{out} \times id} & A' \times B \times S' & \xrightarrow{q'\text{in} \times id} & A \times S' & \xrightarrow{\text{in}} & S' & \xrightarrow{\text{in}} & S' & \xrightarrow{\text{in}} & S' & \xrightarrow{\text{in}} & B & \xrightarrow{q'\text{out}} & B
\end{array}
\]

Since these diagrams respect the composition of morphisms in \( \text{CS}(A, B) \), \( \text{CS}(\phi) \) is a functor. We can also see that \( \text{CS}(\phi) \circ \text{CS}(\psi) = \text{CS}(\phi \circ \psi) \) because both sides send change the inputs and outputs of a dynamical system in the same way.

We then define \( \text{CS}_1 : \mathcal{W}_1 \rightarrow \text{Prof}_1 \) to be the map given by sending \((A, B) \xrightarrow{(f, g)} (C, D) \in \mathcal{W}_1\) to the profunctor \( \text{CS}(A, B)^{op} \times \text{CS}(C, D) \rightarrow \text{Set} \) where

\[
( A \times S \xrightarrow{\text{fin}} S \xrightarrow{\text{fout}} B, C \times T \xrightarrow{\text{fin}} T \xrightarrow{\text{fout}} D )
\]

is mapped to the set of linear maps \( m \) such that the following diagram commutes.

\[
\begin{array}{ccc}
A \times S & \xrightarrow{f\text{in} \times m} & S & \xrightarrow{\text{fout}} & B \\
\downarrow{f\times m} & & \downarrow{m} & & \downarrow{g} \\
C \times T & \xrightarrow{\text{fin}} & T & \xrightarrow{\text{fout}} & D
\end{array}
\]
We send the morphism
\[
(A, B) \xrightarrow{(f, g)} (C, D)
\]
\[
\varphi \downarrow \quad \psi \downarrow \quad \in W_1
\]
\[
(A', B') \xrightarrow{(f', g')} (C', D')
\]
to the natural transformation sending
\[
A \times S \longrightarrow S \longrightarrow B
\]
\[
f \times m \downarrow \quad m \downarrow \quad g
\]
\[
C \times T \longrightarrow T \longrightarrow D
\]
\[
f' \times m \downarrow \quad m \downarrow \quad g'
\]
\[
C' \times T \longrightarrow T \longrightarrow D
\]
by applying \(\varphi\) and \(\psi\) to the corresponding inputs and outputs. This is a valid transformation because of how the squares in \(W_1\) commute.

Since the chosen profunctors for \(CS(U(x))\) align with the hom functor of \(CS(x)\) for any \(x \in \mathcal{W}_0\), we can take \(CS_U\) to be the identity natural transformation. We define \(CS_\otimes\) by assigning to each element \((f, g) \times (f', g') \in W_1 \times \mathcal{W}_0\) the morphism in \(\text{Prof}_1\) sending
\[
A \times S \longrightarrow S \longrightarrow B
\]
\[
\downarrow \quad \downarrow f \quad \downarrow \quad \downarrow g
\]
\[
A' \times S' \longrightarrow S' \longrightarrow B'
\]
\[
\downarrow \quad \downarrow g \quad \downarrow \quad \downarrow g'
\]
\[
A'' \times S'' \longrightarrow S'' \longrightarrow B''
\]
to
\[
A \times S \longrightarrow S \longrightarrow B
\]
\[
\downarrow \quad \downarrow g \circ f \quad \downarrow \quad \downarrow g
\]
\[
A'' \times S'' \longrightarrow S'' \longrightarrow B'
\]
where each of the dynamical systems above are the representatives of their isomorphism classes. This gives us a transformation because the equivalence classes generated by profunctor composition give the same result under our dynamical system morphism composition. Associativity follows from the fact that function composition is associative and unitality follows from the fact that our profunctors and composition transformation work the same way as unit profunctors and their composition.

Now we show that \(CS\) is a lax monoidal double functor. First, we show that \(CS_0\) and \(CS_1\) are lax monoidal functors. We define \(e\) to be the untial morphism of \(CS_0\) sending the trivial category to the
Compositionality of the Runge-Kutta method

Timothy Ngotiaoco

trivial dynamical system \( \ast \times \ast \to \ast \to \ast \) and we define \( \delta \) be the unital morphism of \( CS_1 \) sending the trivial profunctor to the trivial hom from the trivial dynamical system to itself. For monoidal natural transformations, we define \( \mu \) to be the monoidal natural transformation of \( CS_0 \) which sends

\[
( A_1 \times S_1 \to S_1 \to B_1, \ A_2 \times S_2 \to S_2 \to B_2 )
\]
to

\[
( A_1 \times A_2 ) \times ( S_1 \times S_2 ) \to S_1 \times S_2 \to B_1 \times B_2 .
\]

and define \( \nu \) to be the monoidal natural transformation of \( CS_1 \) which sends an ordered pair of dynamical system morphisms to the product of the morphisms. Since everything works componentwise, the associativity and unitality constraints for our lax monoidal functors hold. We already know that \( LCS_1 = CS_0 L \) and \( RCS_1 = CS_0 R \) as functors so we just need to check the lax monoidal properties. All of these functors send ordered pairs of categories of dynamical systems to the category of product dynamical systems in the same way, so the equalities hold. Finally, we need to check that the composition constraints are monoidal natural transformations. The unital transformation is indeed monoidal since it is the identity transformation. For the composition transformation, we need to show that the following diagram commutes.

\[
\begin{array}{ccc}
(CS(M_1) \circ CS(N_1)) \circ (CS(M_2) \circ CS(N_2)) & \to & CS(M_1 \circ N_1) \circ CS(M_2 \circ N_2) \\
\downarrow & & \downarrow \\
CS(M_1 \otimes M_2) \circ CS(N_1 \otimes N_2) & \to & CS((M_1 \otimes M_2) \circ (N_1 \otimes N_2))
\end{array}
\]

This can be seen from the fact that both paths send the pair of composed dynamical system morphisms

\[
\begin{pmatrix}
A \times S & \to & S & \to & B \\
\downarrow & & \downarrow & & \downarrow \\
A' \times S' & \to & S' & \to & B' \\
\downarrow & & \downarrow & & \downarrow \\
A'' \times S'' & \to & S'' & \to & B''
\end{pmatrix}
\begin{pmatrix}
C \times T & \to & T & \to & D \\
\downarrow & & \downarrow & & \downarrow \\
C' \times T' & \to & T' & \to & D'
\end{pmatrix}
\begin{pmatrix}
C'' \times T'' & \to & T'' & \to & D''
\end{pmatrix}
\]
to the morphism

\[
(A \times C) \times (S \times T) \to S \times T \to B \times D
\]

\[
(A'' \times C'') \times (S'' \times T'') \to S'' \times T'' \to B'' \times D''
\]

One path does this by taking the product morphism first, then composing, while the other path composes the morphisms and then takes the product. Since both paths give the same result, we see that \( CS \) is a lax monoidal double functor.
We also define the category of four-step dynamical systems. A four-step dynamical system is a discrete dynamical system along with a commuting diagram

\[
\begin{array}{ccc}
A \times S & \longrightarrow & S \\
\downarrow & & \downarrow \\
* \times \{1, 2, 3, 4\} & \longrightarrow & \{1, 2, 3, 4\} & \longrightarrow & *
\end{array}
\]

where the bottom dynamical system behaves by cyclically iterating through all four elements. We will denote this bottom dynamical system by $C_4$. Another way we can portray this dynamical system is

\[
A \times (S_1 \sqcup S_2 \sqcup S_3 \sqcup S_4) \longrightarrow (S_1 \sqcup S_2 \sqcup S_3 \sqcup S_4) \longrightarrow B
\]

where each $S_i$ is mapped to $i$. A morphism between two four-step dynamical systems is a map $f : S \rightarrow T$ such that $S_i$ maps into $T_i$, and the diagram

\[
\begin{array}{ccc}
A \times S & \longrightarrow & S & \longrightarrow & B \\
\downarrow \text{f} \times m & & \downarrow m & & \downarrow g \\
C \times T & \longrightarrow & T & \longrightarrow & D
\end{array}
\]

commutes.

As in the case for CS, we describe $(DS/C_4)(A, B)$ to be the category of isomorphism classes of four-step dynamical systems with input set $A$ and output set $B$. Then we define $(DS/C_4)_0$ to be the category with objects of the form $(DS/C_4)(A, B)$ and morphisms being wiring diagrams, and we define $(DS/C_4)_1$ to be the category of profunctors describing the morphisms of dynamical system classes with different inputs or outputs. Like we did for CS, we take $(DS/C_4)_U$ to be the identity and $(DS/C_4)_\otimes$ to be the composition natural transformation. The proof that these give a lax monoidal double functor is very similar to the proof for CS.

4 Runge-Kutta as a Double Transformation

Definition 4.1. The Runge-Kutta map $\text{RK}_h(A, B) : \text{CS}(A, B) \rightarrow DS/C_4(A, B)$ is defined by sending a dynamical system

\[
A \times S \longrightarrow S \longrightarrow B
\]

to the dynamical system

\[
A \times (S \sqcup S^2 \sqcup S^3 \sqcup S^4) \longrightarrow S \sqcup S^2 \sqcup S^3 \sqcup S^4 \longrightarrow B
\]

\[
\begin{array}{ccc}
* \times \{1, 2, 3, 4\} & \longrightarrow & \{1, 2, 3, 4\} & \longrightarrow & *
\end{array}
\]
where

\[ f^{upd}(a, s, f^{out}(s)) = (a, s, f^{out}(s)) \]

\[ f^{upd}(a, s, k_1) = (a, s, k_1, f^{out}(s + \frac{h}{2}k_1)) \]

\[ f^{upd}(a, s, k_1, k_2) = (a, s, k_1, k_2, f^{out}(s + \frac{h}{2}k_2)) \]

\[ f^{upd}(a, s, k_1, k_2, k_3) = (a, s + \frac{h}{6}k_1 + \frac{h}{3}k_2 + \frac{h}{3}k_3 + \frac{h}{6}f^{out}(s + \frac{h}{3}k_3), s + \frac{h}{3}k_3) \]

and

\[ f^{r}dt(a, s) = s \]

\[ f^{r}dt(a, s, k_1) = s + \frac{h}{2}k_1 \]

\[ f^{r}dt(a, s, k_1, k_2) = s + \frac{h}{2}k_2 \]

\[ f^{r}dt(a, s, k_1, k_2, k_3) = s + h k_3 \]

**Lemma 4.1.** Let \( X \in CS(A, B) \) be a dynamical system. Then for any wiring diagram \( \phi : (A, B) \rightarrow (C, D) \), we have

\[ RK_h(\phi(X)) = \phi(RK_h(X)) \]

**Proof.** Let \( X \) be the dynamical system

\[ A \times S \xrightarrow{f^{upd}} S \xrightarrow{f^{out}} B \]

Then \( \phi(X) \) is the dynamical system \( C \times S \xrightarrow{g^{upd}} S \xrightarrow{g^{out}} B \) where \( g^{upd}(c, s) = f^{upd}(\phi^{in}(c, f^{out}(s)), s) \)
and \( g^{out} = \phi^{out}(f^{out}(s)) \). We then compute that \( RK_h(\phi(X)) \) is the dynamical system

\[ C \times S \sqcup S^2 \sqcup S^3 \sqcup S^4 \xrightarrow{h^{upd}} S \sqcup S^2 \sqcup S^3 \sqcup S^4 \xrightarrow{h^{out}} D \]

with

\[ h^{upd}_1(c, s) = (c, s, g^{upd}(c, s)) = (c, s, f^{upd}(\phi^{in}(c, f^{out}(s)), s)) \]

\[ h^{upd}_1(c, s, k_1) = (c, s, k_1, g^{upd}(c, s + \frac{h}{2}k_1)) = (c, s, k_1, f^{upd}(\phi^{in}(c, f^{out}(s + \frac{h}{2}k_1)), s + \frac{h}{2}k_1)) \]

\[ h^{upd}_1(c, s, k_1, k_2) = (c, s, k_1, k_2, g^{upd}(c, s + \frac{h}{2}k_2)) = (c, s, k_1, k_2, f^{upd}(\phi^{in}(c, f^{out}(s + \frac{h}{2}k_2)), s + \frac{h}{2}k_2)) \]

\[ h^{upd}_1(c, s, k_1, k_2, k_3) = (c, s + \frac{h}{6}k_1 + \frac{h}{3}k_2 + \frac{h}{3}k_3 + \frac{h}{6}g^{upd}(c, s + h k_3)) \]

\[ = (c, s + \frac{h}{6}k_1 + \frac{h}{3}k_2 + \frac{h}{3}k_3 + \frac{h}{6}f^{upd}(\phi^{in}(c, f^{out}(s + h k_3)), s + h k_3)) \]

\[ \square \]
Finally, we restate Theorem 1 and prove it.

**Theorem 1.** The Runge-Kutta method is a monoidal double transformation between CS and DS/C₄.

**Proof.** We first show that \((RK_0)_0\) defines a natural transformation \(CS_0 \to (DS/C_4)_0\). For this, we need to show that \(RK_h\) gives a functor from \(CS_0(A, B)\) to \((DS/C_4)(A, B)\). This fact follows from the above lemma and the fact that we are sending the CS-morphism

\[
\begin{align*}
A \times S & \xrightarrow{f_{upd}} S \xrightarrow{f_{out}} B \\
A \times S' & \xrightarrow{f'_{upd}} S' \xrightarrow{f'_{out}} B'
\end{align*}
\]

\[\varphi \times (m \sqcup m^2 \sqcup m^3 \sqcup m^4)\]

\[\begin{align*}
A \times (S \sqcup S^2 \sqcup S^3 \sqcup S^4) & \longrightarrow S \sqcup S^2 \sqcup S^3 \sqcup S^4 \longrightarrow B \\
A \times (S' \sqcup S^2 \sqcup S^3 \sqcup S^4) & \longrightarrow S' \sqcup S^2 \sqcup S^3 \sqcup S^4 \longrightarrow B'
\end{align*}
\]

Since the morphisms are component-wise, we have that the functorial properties hold. Now we just need to show that these morphisms in Prof behave naturally with CS and DS/C₄, but this follows from our above Lemma, so we can see that \((RK_0)_0\) gives us a natural transformation between \(CS_0\) and \((DS/C_4)_0\).

We then show that \((RK_1)_1\) gives a natural transformation \(CS_1\) to \((DS/C_4)_1\). This is not hard to check since we are sending

\[
\begin{align*}
A \times S & \xrightarrow{f_{upd}} S \xrightarrow{f_{out}} B \\
A' \times S' & \xrightarrow{f'_{upd}} S' \xrightarrow{f'_{out}} B'
\end{align*}
\]

\[\varphi \times (m \sqcup m^2 \sqcup m^3 \sqcup m^4)\]

\[\begin{align*}
A \times (S \sqcup S^2 \sqcup S^3 \sqcup S^4) & \longrightarrow S \sqcup S^2 \sqcup S^3 \sqcup S^4 \longrightarrow B \\
A' \times (S' \sqcup S^2 \sqcup S^3 \sqcup S^4) & \longrightarrow S' \sqcup S^2 \sqcup S^3 \sqcup S^4 \longrightarrow B'
\end{align*}
\]

and all the morphisms are component-wise. One can check the diagrams in the definition commute since our functions act component-wise and also compose component-wise.

Finally, we just need to show that these natural transformations are monoidal so that we have a lax monoidal double transformation. For this, we need the natural transformation to commute with our monoidal transformations. This is true for both natural transformations, since multiplying
the dynamical systems and then applying Runge-Kutta gives an isomorphic dynamical system as applying Runge-Kutta to two dynamical systems and then merging the two systems. The only difference is that the order of the products in the discrete sum may be different, but since our maps work component-wise, we still get isomorphic results, so our natural transformation indeed commutes with our monoidal transformations. Our last requirement, the unital requirement for the natural transformations, is easy to check, so Runge-Kutta indeed gives a monoidal double transformation.

5 Acknowledgements

The author would like to thank David Spivak for many helpful discussions.

References

[Shu09] Michael Shulman. Framed bicategories and monoidal fibrations. 2009. arXiv: 0706.1286 [math.CT]. url: https://arxiv.org/pdf/0706.1286.pdf.

[Shu10] Michael Shulman. Constructing symmetric monoidal bicategories. 2010. arXiv: 1004.0993 [math.CT]. url: https://arxiv.org/pdf/1004.0993.pdf.

[Spi16] David Spivak. The steady states of coupled dynamical systems compose according to matrix arithmetic. 2016. arXiv: 1512.00802 [math.DS]. url: https://arxiv.org/pdf/1512.00802.pdf.