Fractional
COST Action CA15225; fractional-systems.eu

cost
EUROPEAN COOPERATION IN SCIENCE & TECHNOLOGY

Lecture notes on Variable Order Derivatives

Duarte Valério

TÉCNICO LISBOA

September 7, 2017

Contents

Contents i

Introduction iii

1 Natural orders 1
1.1 Intuitive results . . . 1
1.2 Switching after differentiating . . . 1
1.3 Switching before differentiating . . . 1
1.4 Switching both before and after differentiating . . . 1
1.5 Implementing this with operational amplifiers . . . 5

2 Integer orders 7
2.1 Introduction . . . 7
2.2 Switching after differentiating . . . 7
2.3 Switching before differentiating . . . 7
2.4 Switching both before and after differentiating . . . 7
2.5 What is going on . . . 7

3 Real orders 13
3.1 Definitions for constant orders . . . 13
3.2 No memory . . . 13
3.3 How to get a memory . . . 14
3.4 Another way to get a memory . . . 14
3.5 Literature for the real case . . . 14

4 Complex orders 15
4.1 Constant complex orders . . . 15
4.2 Variable complex orders . . . 15
4.3 Literature for the complex case . . . 16

5 Matlab toolboxes 17

6 Modelling bone remodelling dynamics 19
6.1 Introduction . . . 19
6.2 Osteoclasts and osteoblasts . . . 19
6.3 Tumours and treatments . . . 19
6.4 Literature for bone remodelling with variable order derivatives . . . 20

7 Related topics 21

Bibliography 21

Introduction

Mudam-se os tempos, mudam-se as vontades,
Muda-se o ser, muda-se a confiança;
Todo o mundo é composto de mudança,
Tomando sempre novas qualidades.

Continuamente vemos novidades,
Diferentes em tudo da esperança;
Do mal ficam as mágoas na lembrança,
E do bem, se algum houve, as saudades.

O tempo cobre o chão de verde manto,
Que já coberto foi de neve fria,
E em mim converte em choro o doce canto.

E, afora este mudar-se cada dia,
Outra mudança faz de mor espanto:
Que não se muda já como soía.

Luís de Camões (1524? – †1580)

Time changes, and our desires change. What we
believe — even what we are — is ever-
changing. The world *is* change, which forever
takes on new qualities. And constantly,

we see the new and the novel overturning
the past, unexpectedly, while we retain
from evil, nothing but its terrible pain,
from good (if there's been any), only the yearning.

Time covers the ground with her cloak of green
where, once, there was freezing snow — and rearranges
my sweetest songs to sad laments. Yet even more

astonishing is yet another unseen
change within all these endless changes:
that for me, *nothing* ever changes anymore[1].

Transl. William Baer

Our objective is the study of an operator $D^{\alpha(t)}f(t)$, where $f(t)$ is the differentiated function, and $\alpha(t)$ is the time-variable differentiation order. The reader is presumed to already know what $D^{\alpha}f(t),\ \alpha \in \mathbb{R}$ is.

We will study variable order derivatives considering four cases:

- Chapter 1 — natural orders
- Chapter 2 — integer orders
- Chapter 3 — real orders
- Chapter 4 — complex orders

These lecture notes introduce all these cases, but above all present papers where the last two are discussed at length.

The following matters are also mentioned with references to corresponding published literature:

- Chapter 5 — Matlab toolboxes
- Chapter 6 — application to bone remodelling dynamics
- Chapter 7 — related topics

[1] There is, I am afraid, a mistranslation: the last verse in the original means "that changes are no longer as they used to be". This, unfortunately, spoils both metre and rhyme.

Chapter 1

Natural orders

1.1 Intuitive results

If $\alpha(t) \in \mathbb{N}, \forall t$ then its variations with time must be steps, since it can only assume values in a discrete set.

Let us consider for instance the case

$$f(t) = t^2 \tag{1.1}$$

$$\alpha(t) = \begin{cases} 2 \text{ if } t \in [0,1[\cup [2,3[\cup [4,5[\cup \dots \\ 1 \text{ if } t \in [1,2[\cup [3,4[\cup [5,6[\cup \dots \end{cases} \tag{1.2}$$

Of course,

$$D^1 f(t) = \frac{\mathrm{d}}{\mathrm{d}t} t^2 = 2t \tag{1.3}$$

$$D^2 f(t) = \frac{\mathrm{d}^2}{\mathrm{d}t^2} t^2 = 2 \tag{1.4}$$

and we can intuitively see that the reasonable result of operator $D^{\alpha(t)} f(t)$ in this case, when orders are natural, is

$$D^{\alpha(t)} f(t) = \begin{cases} 2 \text{ if } t \in [0,1[\cup [2,3[\cup [4,5[\cup \dots \\ 2t \text{ if } t \in [1,2[\cup [3,4[\cup [5,6[\cup \dots \end{cases} \tag{1.5}$$

In other words, we are just jumping from one function to the other.

1.2 Switching after differentiating

Let us think about a reasonable way of putting this in a block diagram, and see how that works in Simulink. See figure 1.1, where the signal generator is set to a square wave with amplitude 1 and frequency 0.5 Hz, and so it will activate the switch as required by (1.2). The results shown are obtained running this simulation for 10 s with a fixed time-step of 0.001 s. (Subsequent simulations will keep using these settings too.)

1.3 Switching before differentiating

We could have thought about doing as in figure 1.2 instead. Notice that we have numerical problems because we are trying to differentiate a signal with steps. Save for that, the result is the same as above.

1.4 Switching both before and after differentiating

Rather than summing the two results as in figure 1.2, we could use yet another switch, as in figure 1.3. This may seem unnecessary (and a waste of an electronic component when the circuit is implemented in hardware), but makes all sense. Results are just as those of figure 1.2 (numerical problems and all).

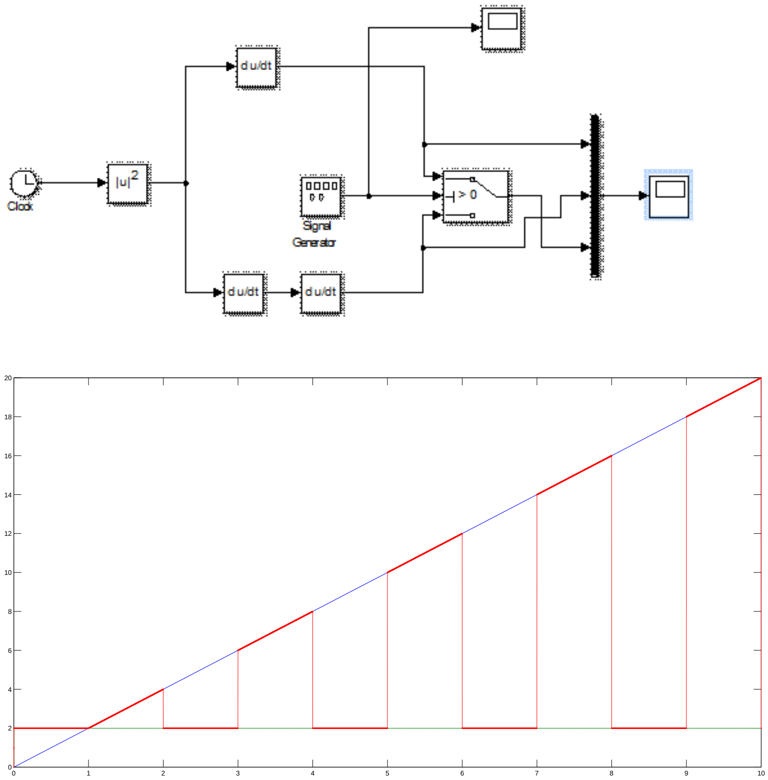


Figure 1.1: Switching after differentiating. Natural orders. Top: Simulink implementation. Bottom: results. Blue: $D^1 f(t)$; green: $D^2 f(t)$; red dots: $D^{\alpha(t)} f(t)$.

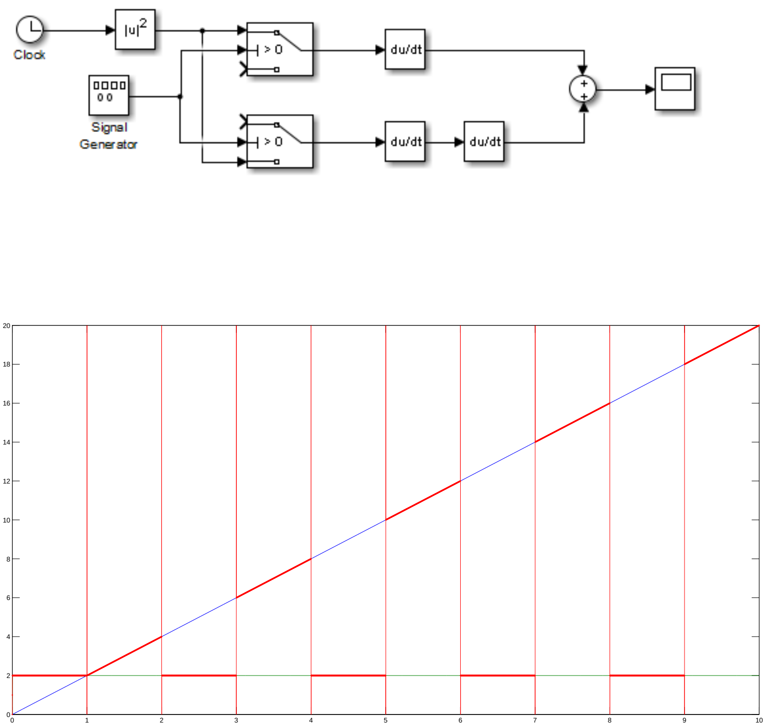
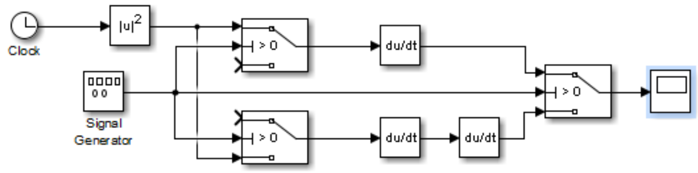


Figure 1.2: Switching before differentiating. Natural orders. Top: Simulink implementation. Bottom: results. Blue: $D^1 f(t)$; green: $D^2 f(t)$; red dots: $D^{\alpha(t)} f(t)$.



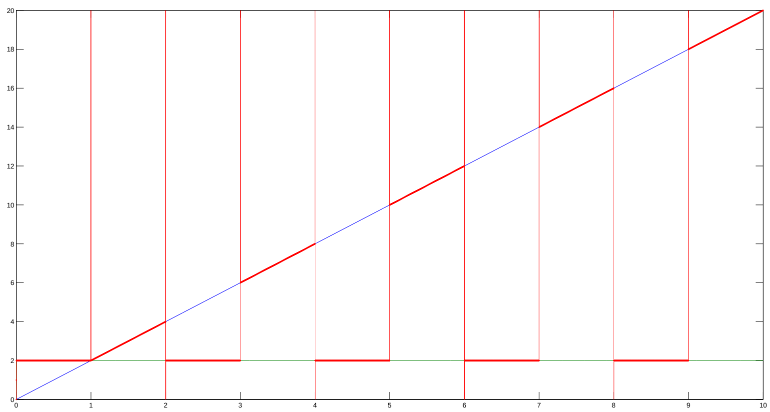


Figure 1.3: Switching both before and after differentiating. Natural orders. Top: Simulink implementation. Bottom: results. Blue: $D^1 f(t)$; green: $D^2 f(t)$; red dots: $D^{\alpha(t)} f(t)$.

1.5 Implementing this with operational amplifiers

It is left to the care of the reader to develop circuits with opamps implementing these Simulink simulations electronically. If you have some knowledge of opamp circuits this should be a trivial exercise.

Chapter 2

Integer orders

2.1 Introduction

In this case, we are still limited to steps in the order $\alpha(t)$. But when we use negative orders (integration) significant differences will appear.

Let us now make

$$f(t) = t \tag{2.1}$$

$$\alpha(t) = \begin{cases} -1 \text{ if } t \in [0,1[\cup [2,3[\cup [4,5[\cup \dots \\ 0 \text{ if } t \in [1,2[\cup [3,4[\cup [5,6[\cup \dots \end{cases} \tag{2.2}$$

2.2 Switching after differentiating

Figure 2.1 shows the corresponding Simulink simulation with the switch at the end. Of course,

$$D^{-1} f(t) = \int_0^t t \, \mathrm{d}t = \frac{1}{2} t^2 \tag{2.3}$$

$$D^0 f(t) = t \tag{2.4}$$

and we are still jumping from one function to another. Notice, consequently, that the integration goes on even when it is not being used.

But should it?

2.3 Switching before differentiating

Notice now what happens when the switch is at the beginning. See figure 2.2. Numerical problems are gone: we are no longer differentiating, so nothing goes wrong because of the steps. But the result is now completely different. Whenever we stop using the integral, its value does not grow. When we use it again, it restarts at the same value it had when we left it. And when we switch to the function itself, it is pushed up, because it is summed to the last value of the integral. That branch consists, consequently, of unaligned straight lines.

2.4 Switching both before and after differentiating

When we switch twice, as in figure 2.3, results are different again. The integral behaves in the same way as in figure 2.2, but the zero-order pieces are as in figure 2.1.

2.5 What is going on

This clearly shows that we are dealing with different things by putting switches in different positions.

In the first case, when the switch is at the end, there is no memory of the past values of the order. Whenever the order changes, everything becomes as it would have been if that order had been used ever since the beginning.

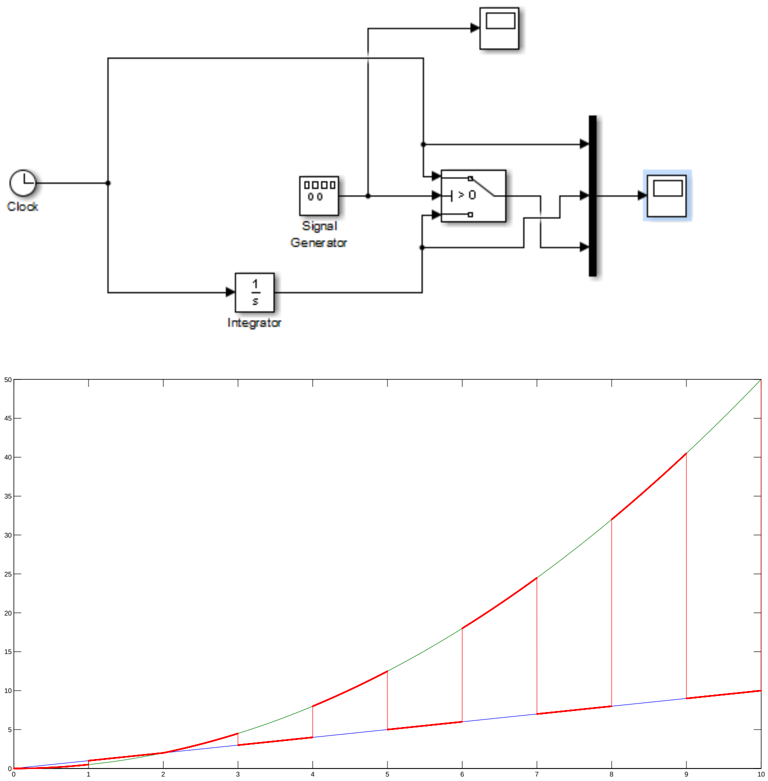


Figure 2.1: Switching after differentiating. Integer orders. Top: Simulink implementation. Bottom: results. Blue: $f(t)$; green: $D^{-1}f(t)$; red dots: $D^{\alpha(t)}f(t)$.

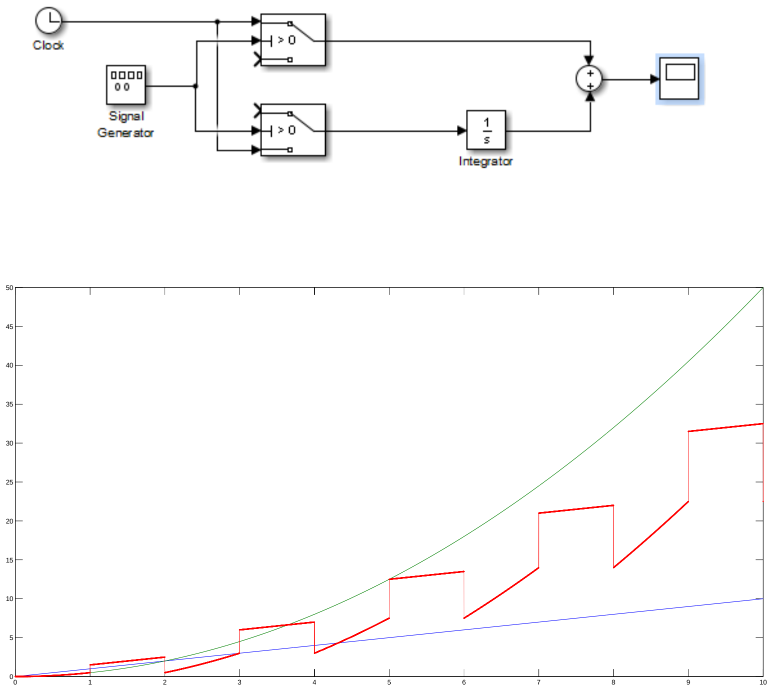


Figure 2.2: Switching before differentiating. Integer orders. Top: Simulink implementation. Bottom: results. Blue: $f(t)$; green: $D^{-1}f(t)$; red dots: $D^{\alpha(t)}f(t)$.

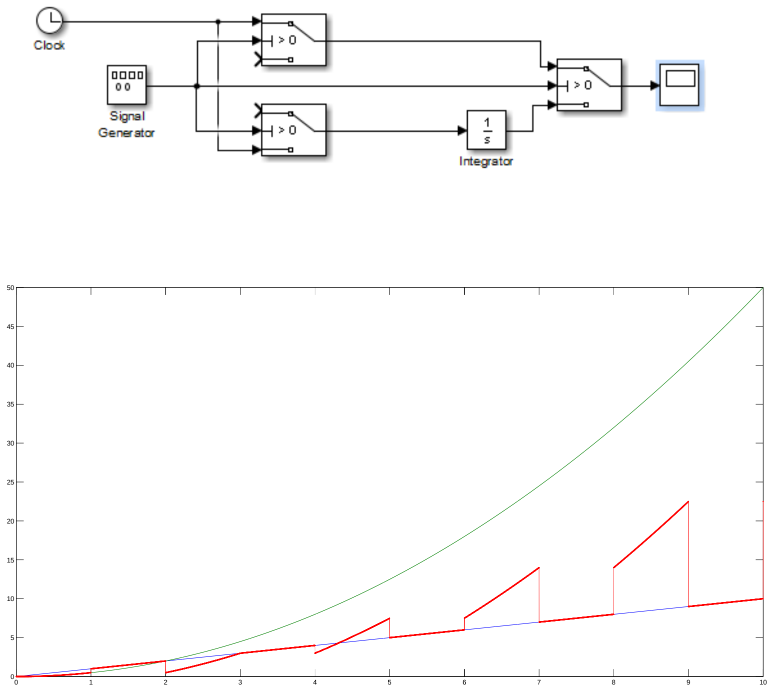


Figure 2.3: Switching both before and after differentiating. Integer orders. Top: Simulink implementation. Bottom: results. Blue: $f(t)$; green: $D^{-1}f(t)$; red dots: $D^{\alpha(t)}f(t)$.

In the second case, when the switch is at the beginning, the operator has a memory of what has been happening with the order.

When there are switches both at the beginning and at the end, there is memory, but apparently not so strong.

All options make sense. It is not that one is correct and the others foolish. They correspond to different definitions of variable order derivatives.

We will introduce them for real orders.

Chapter 3

Real orders

We are finally in a situation where $\alpha(t)$ can vary continuously (though of course steps and other discontinuities are not ruled out). Notice that while in the natural case a memory of past values of the order was irrelevant, since derivatives are local operators, and in the integer case the memory only mattered for negative orders, since integrals are not local, now, in the real case, and as the only orders corresponding to a local operator are the natural numbers, the existence or not of a memory of past values of the order is always critical.

3.1 Definitions for constant orders

Let us recall the Grünwald-Letnikoff (GL) definition of fractional derivatives:

$$ {}_cD_t^\alpha f(t) = \lim_{h\to 0^+} \frac{\sum_{k=0}^{\lfloor \frac{t-c}{h} \rfloor} (-1)^k \left(\begin{array}{c} \alpha \\ k \end{array} \right) f(t-kh)}{h^\alpha} \tag{3.1} $$

And let us recall the Riemann-Liouville (RL) definition:

$$ {}_cD_t^\alpha f(t) = \begin{cases} \int_c^t \frac{(t-\tau)^{-\alpha-1}}{\Gamma(-\alpha)} f(\tau)\,\mathrm{d}\tau, \text{ if } \alpha \in \mathbb{R}^- \\ f(t), \text{ if } \alpha = 0 \\ \frac{\mathrm{d}^{\lceil\alpha\rceil}}{\mathrm{d}t^{\lceil\alpha\rceil}} {}_cD_t^{\alpha-\lceil\alpha\rceil} f(t), \text{ if } \alpha \in \mathbb{R}^+ \end{cases} \tag{3.2} $$

The Caputo definition is a variation of the RL definition and is covered in the literature discussed below. There is no need to mention it here further. We will also not bother with the case where terminals are the other way round (fixed upper terminal c, lower terminal equal to time t).

3.2 No memory

If we want a variable order operator $D^{\alpha(t)}$ with no memory of what happened to the order before the current time instant, what we have to do is to use, to calculate the derivative in each time instant, the order that we have now, applying it to all time instants ever since c:

$$ {}_cD_t^{\alpha(t)} f(t) = \lim_{h\to 0^+} \frac{\sum_{k=0}^{\lfloor \frac{t-c}{h} \rfloor} (-1)^k \left(\begin{array}{c} \alpha(t) \\ k \end{array} \right) f(t-kh)}{h^{\alpha(t)}} \tag{3.3} $$

The corresponding RL definition will be

$$ {}_cD_t^{\alpha(t)} f(t) = \begin{cases} \int_c^t \frac{(t-\tau)^{-\alpha(t)-1}}{\Gamma(-\alpha(t))} f(\tau)\,\mathrm{d}\tau, \text{ if } \alpha(t) \in \mathbb{R}^- \\ f(t), \text{ if } \alpha(t) = 0 \\ \frac{\mathrm{d}^{\lceil\alpha(t)\rceil}}{\mathrm{d}t^{\lceil\alpha(t)\rceil}} {}_cD_t^{\alpha(t)-\lceil\alpha(t)\rceil} f(t), \text{ if } \alpha(t) \in \mathbb{R}^+ \end{cases} \tag{3.4} $$

Both definitions work just the same.

3.3 How to get a memory

It seems reasonable to think that, in the summation of (3.1), we should rather use for each time instant the value of the order that was available at that time instant.

$$ {}_cD_t^{\alpha(t)} f(t) = \lim_{h\to 0^+} \sum_{k=0}^{\lfloor \frac{t-c}{h} \rfloor} \frac{(-1)^k \begin{pmatrix} \alpha(t-kh) \\ k \end{pmatrix} f(t-kh)}{h^{\alpha(t-kh)}} \tag{3.5} $$

And it seems reasonable to expect this to result in a memory of past values of $\alpha(t)$. It is indeed so. (3.5) corresponds to the simulations with the switch before the differentiations.

Now the interesting thing is that the corresponding RL version

$$ {}_cD_t^{\alpha(t)} f(t) = \begin{cases} \displaystyle\int_c^t \frac{(t-\tau)^{-\alpha(\tau)-1}}{\Gamma(-\alpha(\tau))} f(\tau)\,\mathrm{d}\tau, \text{ if } \alpha(t) \in \mathbb{R}^- \\ f(t), \text{ if } \alpha(t) = 0 \\ \dfrac{\mathrm{d}^{\lceil \alpha(t) \rceil}}{\mathrm{d}t^{\lceil \alpha(t) \rceil}} {}_cD_t^{\alpha(t)-\lceil \alpha(t) \rceil} f(t), \text{ if } \alpha(t) \in \mathbb{R}^+ \end{cases} \tag{3.6} $$

does not provide the same result. In fact, (3.6) corresponds to the simulations with switches both before and after the differentiation.

3.4 Another way to get a memory

When we take a look at (3.2), we see that to avoid a memory of past values of the order we used t as the argument of order $\alpha(t)$, and to get a memory we used τ. The only thing we did not do yet was using $t-\tau$, that appears in the numerator of the kernel. But we could. And then this is the result:

$$ {}_cD_t^{\alpha(t)} f(t) = \begin{cases} \displaystyle\int_c^t \frac{(t-\tau)^{-\alpha(t-\tau)-1}}{\Gamma(-\alpha(t-\tau))} f(\tau)\,\mathrm{d}\tau, \text{ if } \alpha(t) \in \mathbb{R}^- \\ f(t), \text{ if } \alpha(t) = 0 \\ \dfrac{\mathrm{d}^{\lceil \alpha(t) \rceil}}{\mathrm{d}t^{\lceil \alpha(t) \rceil}} {}_cD_t^{\alpha(t)-\lceil \alpha(t) \rceil} f(t), \text{ if } \alpha(t) \in \mathbb{R}^+ \end{cases} \tag{3.7} $$

Now in the GL definition what corresponds to the difference between t (the current time) and τ (the time at which the function appears in the definition) turns out to be $t-(t-kh) = kh$. And so we end up with

$$ {}_cD_t^{\alpha(t)} f(t) = \lim_{h\to 0^+} \sum_{k=0}^{\lfloor \frac{t-c}{h} \rfloor} \frac{(-1)^k \begin{pmatrix} \alpha(kh) \\ k \end{pmatrix} f(t-kh)}{h^{\alpha(kh)}} \tag{3.8} $$

Both these definitions work, and they are different from each other and from all the other ones we have seen.

3.5 Literature for the real case

All the above definitions, for the RL case, are introduced in [2]. That paper concentrates on fractional integrals ($\alpha < 0$) and mentions fractional derivatives ($\alpha > 0$) in passing.

These definitions are considered for the GL case in [5], which presents schemes of electronic circuits to implement the operators, and experimental results.

Finally, [6] presents the results of [5] again and expands them, using recursive definitions of fractional derivatives to define variable order derivatives in yet new ways. This paper too shows diagrams of electronic circuits and photos of the hardware with which that team obtained experimental results.

Chapter 4

Complex orders

4.1 Constant complex orders

Let us see what happens when, instead of an order $\alpha \in \mathbb{R}$, we have an order $\mathfrak{z} \in \mathbb{C}$, with $\mathfrak{a} = \Re[\mathfrak{z}]$ and $\mathfrak{b} = \Im[\mathfrak{z}]$.

One of the good things of the GL definition (3.1) is that it works without any problem for complex orders. Just put the complex order there and you are good to go.

Things are not so simple with the RL definition. We will use:

- the branch for $\alpha < 0$ when $\mathfrak{a} < 0$,
- the branch for $\alpha > 0$ when $\mathfrak{a} > 0$,
- a third branch similar to that for $\alpha > 0$ when $\mathfrak{b} = \mathfrak{z} \neq 0$.

We will end up with

$$
{}_cD_t^{\mathfrak{z}} f(t) = \begin{cases} \displaystyle\int_c^t \frac{(t-\tau)^{-\mathfrak{z}-1}}{\Gamma(-\mathfrak{z})} f(\tau)\,\mathrm{d}\tau, \text{ if } \mathfrak{a} \in \mathbb{R}^- \\ f(t), \text{ if } \mathfrak{z} = 0 \\ \displaystyle\frac{\mathrm{d}}{\mathrm{d}t}\, {}_cD_t^{\mathfrak{z}-1} f(t), \text{ if } \mathfrak{a} = 0 \wedge \mathfrak{b} \neq 0 \\ \displaystyle\frac{\mathrm{d}^{\lceil \mathfrak{a} \rceil}}{\mathrm{d}t^{\lceil \mathfrak{a} \rceil}}\, {}_cD_t^{\mathfrak{z}-\lceil \mathfrak{a} \rceil} f(t), \text{ if } \mathfrak{a} \in \mathbb{R}^+ \end{cases} \tag{4.1}
$$

These two definitions are equivalent.

Notice that a complex order derivative of a real signal is complex valued.

4.2 Variable complex orders

Expanding our previous definitions of variable order derivatives to the complex case is straightforward. Here you have the expressions for the case with no memory. The others cases are left to the reader's care, and can be found in the literature anyway.

$$
{}_cD_t^{\mathfrak{z}(t)} f(t) = \begin{cases} \displaystyle\int_c^t \frac{(t-\tau)^{-\mathfrak{z}(t)-1}}{\Gamma(-\mathfrak{z}(t))} f(\tau)\,\mathrm{d}\tau, \text{ if } \mathfrak{a}(t) \in \mathbb{R}^- \\ f(t), \text{ if } \mathfrak{z}(t) = 0 \\ \displaystyle\frac{\mathrm{d}}{\mathrm{d}t}\, {}_cD_t^{\mathfrak{z}(t)-1} f(t), \text{ if } \mathfrak{a}(t) = 0 \wedge \mathfrak{b}(t) \neq 0 \\ \displaystyle\frac{\mathrm{d}^{\lceil \mathfrak{a}(t) \rceil}}{\mathrm{d}t^{\lceil \mathfrak{a}(t) \rceil}}\, {}_cD_t^{\mathfrak{z}(t)-\lceil \mathfrak{a}(t) \rceil} f(t), \text{ if } \mathfrak{a}(t) \in \mathbb{R}^+ \end{cases} \tag{4.2}
$$

$$
{}_cD_t^{\mathfrak{z}(t)} f(t) = \lim_{h \to 0^+} \frac{\displaystyle\sum_{k=0}^{\lfloor \frac{t-c}{h} \rfloor} (-1)^k \begin{pmatrix} \mathfrak{z}(t) \\ k \end{pmatrix} f(t-kh)}{h^{\mathfrak{z}(t)}} \tag{4.3}
$$

4.3 Literature for the complex case

You can find variable order complex derivatives treated in [9].

Chapter 5

Matlab toolboxes

The GL, RL and Caputo versions of the three possibilities of defining variable order derivatives are implemented in [8], which the requires the `ninteger` toolbox [7]. The functions will be integrated into [7] one day, whenever the author will have time; [7] is in bad need of a revision anyway.

Those functions are implemented to be called from the command prompt. There are Simulink implementations as well. These deal with continuous variable real orders by interpolating among some fixed orders. There are several ways of doing so and you can find (and explore) them all there. Some have no memory of past orders, others do. These ideas are covered in [9] too.

All the definitions in [6] are implemented in [4] as Simulink blocks. This is very convenient and you should compare the results of the different implementations trying to understand what is going on.

Chapter 6

Modelling bone remodelling dynamics

6.1 Introduction

Variable order derivatives can be used for many things. One of them is to model what happens to human bone remodelling dynamics when the bone is affected by a tumour. The tumour changes the dynamics of bone remodelling for its own profit. These changes can be described as a change in the order of the system of differential equation that models the process. The same applies to tumour treatments, that try to fight the tumour, regenerate the bone, and get things back to what they were.

6.2 Osteoclasts and osteoblasts

Bone tissue is not static. Just like every other part of our body, its cells are always dying and being replaced. Figure 6.1 illustrates the main actors in bone remodelling. To cut a long story short, there are some big cells destroying bone tissue, called osteoclasts, and then cells that build bone back, called osteoblasts. Let C be the number of osteoclasts and B the number of osteoblasts. The presence of osteoblasts influences the rate of increase of osteoclasts — this is called a paracrine effect —, and the number of osteoclasts also influences their own evolution — this is called an autocrine effect. It turns out that we can write

$$D^1 C(t) = \alpha_C C(t)^{g_{CC}} B(t)^{g_{BC}} - \beta_C C(t) \tag{6.1}$$

Here α_C is a scaling constant, and the exponents g_{CC} and g_{BC} correspond to the autocrine and paracrine effects respectively. The last term is there because osteoclasts die, like any other cell, and β_C is their death rate (or apoptosis rate, to use a fancier name).

A rather similar equation can be written for osteoblasts:

$$D^1 B(t) = \alpha_B C(t)^{g_{CB}} B(t)^{g_{BB}} - \beta_B B(t) \tag{6.2}$$

As to the bone mass itself z, it decreases when there are more osteoclasts than usual, and increases when there are more osteoblasts than usual. It is, in fact, presumed that when below their average level there will be no significant effect in z. So we write

$$D^1 z(t) = -\kappa_C \max\left[0, C(t) - C_{ss}\right] + \kappa_B \max\left[0, B(t) - B_{ss}\right] \tag{6.3}$$

κ_C and κ_B are proportionality constants; C_{ss} and B_{ss} are steady state values.

6.3 Tumours and treatments

The changes in dynamic behaviour when there is a tumour can be modelled changing the parameters of the autocrine and paracrine effects. Then, treatments change those changes. In the end, (6.1) and (6.2) must be

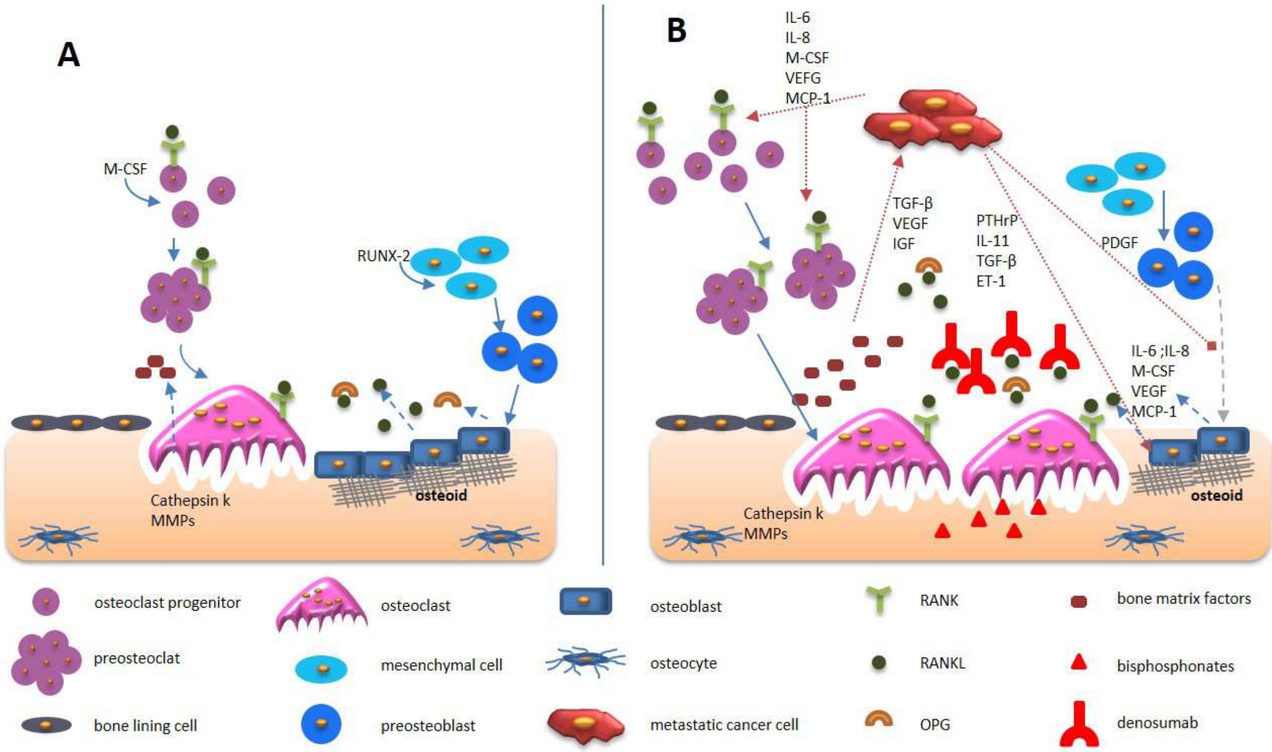


Figure 6.1: Bone dynamics (taken from an upcoming paper).

replaced with

$$D^1C(t) = \alpha_C C(t)^{g_{CC}\left(1+r_{CC}\frac{T(t)}{L_T}\right)} B(t)^{g_{BC}\left(1+r_{BC}\frac{T(t)}{L_T}\right)\left(1+K_{s_1}d_1(t)\right)} - \left(1+K_{s_2}d_2(t)\right)\beta_C C(t) \tag{6.4}$$

$$D^1B(t) = \alpha_B C(t)^{\left(\frac{g_{CB}}{1+r_{CB}\frac{T(t)}{L_T}}\right)} B(t)^{\left(g_{BB}-r_{BB}\frac{T(t)}{L_T}\right)} - \beta_B B(t) \tag{6.5}$$

$$D^1T(t) = \left(1-K_{i_{34}}d_{c_{34}}(t)\right)\gamma_T T(t)\log\left(\frac{L_T}{T(t)}\right) \tag{6.6}$$

There is no need to explain all the new variables. It just suffices to say that the same effects can be obtained much more simply by changing the differentiation order of (6.1) and (6.2).

6.4 Literature for bone remodelling with variable order derivatives

[3] covers the results obtained for local bone remodelling models when a tumour appears. Cancer treatments, and models including diffusion of cells along the bone, are found in upcoming papers.

Chapter 7

Related topics

Variable order controllers make use of variable order derivatives. You may find several examples in [10].

The order may change with time not because it depends directly on time but because it depends on something else. You can find in [1] an example of a mechanical problem where there is a fractional derivative $D^{q(\xi(t))}$ that models a viscoelastic effect with an order q that depends on the position ξ of an oscillator. This position, of course, changes with time, and the way it changes with time depends on the effect of the variable order derivative. Not very different things turn up in models of anomalous diffusion.

Bibliography

[1] Carlos F. M. Coimbra. Mechanics with variable-order differential operators. *Annalen der Physik*, 12(11-12):692–703, 2003.

[2] Carl F. Lorenzo and Tom T. Hartley. Variable order and distributed order fractional operators. *Nonlinear Dynamics*, 29:57–98, 2002.

[3] Joana Neto, Rui Moura Coelho, Duarte Valério, Susana Vinga, Dominik Sierociuk, Wiktor Malesza, Michal Macias, and Andrzej Dzielinski. Variable order differential models of bone remodelling. In *IFAC World Congress*, 2017. Accepted.

[4] Dominik Sierociuk. Fractional variable order derivative simulink toolkit. `https://www.mathworks.com/matlabcentral/fileexchange/38801-fractional-variable-order-derivative-si`, 2015.

[5] Dominik Sierociuk, Wiktor Malesza, and Michal Macias. Derivation, interpretation, and analog modelling of fractional variable order derivative definition. *Applied Mathematical Modelling*, 39:3876–3888, 2015.

[6] Dominik Sierociuk, Wiktor Malesza, and Michal Macias. On the recursive fractional variable-order derivative: Equivalent switching strategy, duality, and analog modeling. *Circuits, Systems and Signal Processing*, 34:1077–1113, 2015.

[7] Duarte Valério. ninteger. `http://www.mathworks.com/matlabcentral/fileexchange/8312-ninteger`, 2005.

[8] Duarte Valério. Variable order derivatives. `https://www.mathworks.com/matlabcentral/fileexchange/24444-vari`, 2010.

[9] Duarte Valério and José Sá da Costa. Variable-order fractional derivatives and their numerical approximations. *Signal Processing*, 91(3):470–483, 2011.

[10] Duarte Valério and José Sá da Costa. Variable order fractional controllers. *Asian Journal of Control*, 15(3):648–657, 2013.