SCHOONSCHIP, THE LARGEST TIME EQUATION AND THE CONTINUOUS DIMENSIONAL REGULARISATION

Ettore Remiddi

Dipartimento di Fisica e Astronomia, University of Bologna, Italy
ettore.remiddi@bo.infn.it

(Received May 11, 2021; accepted May 21, 2021)

I will recall three results of Martinus J.G. Veltman, which had a substantial importance in my scientific activity and which gave me the occasion of meeting him and of appreciating the great human and scientific gifts of his unforgettable personality.

DOI:10.5506/APhysPolB.52.513

1. Introduction

Tini Veltman had a highly positive, strong influence on my whole professional activity, although I was not a student of his and I never coauthored a paper with him or under his supervision. He was ten years older than me, taller — and with an imposing beard. The first time we exchanged letters I addressed him as "Dear Tiny, ...", as I had never heard anyone calling him Martinus; in his answer, he replied: "Do I look tiny to you?" (but I think that happened also to other people).

I got acquainted with him in 1968 at CERN. In those years, he was still young, but already rather authoritative, at the first sight I felt somewhat intimidated, but soon he became for me a kind of "senior friend", ready to help and always to be listened to with attention.

In this paper, I will recall in the three sections which follow, three of his many scientific results, for the importance that they had on my whole professional life, and for the occasion which they gave me of knowing him and of entering in contact with his unforgettable personality. As it is obvious from the titles, the three sections refer to his Computer Algebra Program SCHOONSCHIP, to the *Largest Time Equation* and the related cutting rules, and to the use of the *Continuous Dimensional Regularisation* in the derivation of differential equations for Feynman graph amplitudes.

2. SCHOONSCHIP

As a graduate student in Pisa, I had written an extremely primitive Fortran program for evaluating SU(3) Clebsch–Gordan coefficients (the results were printed in single precision arithmetics, so that, for instance, the number 0.8660254 in the output was to be interpreted as $\sqrt{3}/2$).

In 1966, I got a fellowship for the CERN-TH Division, and I had in mind to write a similar program (but my ideas were really confusing, if not simply wrong) for the traces of Dirac's gamma-matrices, whose evaluation was still considered, in those years, the very first serious challenge encountered in working out Feynman graphs.

I talked of my intentions to a common friend of me and Tini, Daniele Amati, who told me immediately that Tini Veltman had already solved the problem, and suggested that I look for him, before risking wasting effort and time in something which could turn out to be for me too demanding. So I started looking around for Tini.

At that time Tini was almost regularly commuting between CERN and Utrecht (with a French car which I still remember); at his next coming to CERN I plucked up courage, introduced myself and asked him about the program.

To my surprise, contrary to my fears and to his somewhat gruff external appearance, he was extremely kind; he brought me into some office of the CERN-TH Division (keys were not in use in those years), opened a cabinet with four cardboxes filled with punched cards (the SCHOONSCHIP source code!), told me that I could use it at my will, and gave me rudimentary but *foolproof* verbal instructions on how to use the cards, adding, of course, the ritual threats for any injury that the cards could ever suffer in my hands; and shortly afterwards left for Utrecht.

At that time, I was working with my friend Goyi, the late Juan Alberto Mignaco (1940–2001). As the source code of SCHOONSCHIP exceeded the maximal size for the jobs processed during the normal shifts of the Computer Service, we booked the first available night time at the CDC-6400 (the smaller CERN machine, which you could use without an attending operator), took the cardboxes from the cabinet and slipped the source cards into the fast card reader of the computer to obtain the compiled binary code.

But the punched cards, perhaps because kept too long in the cardboxes without an appropriate pressure, were somewhat bent and the fast card reader started to slam-smash them in a rolled-up shape in the exit tray — a nightmare. Really scared, I had to stop everything, to copy-repunch with great attention a bunch of cards which could not be properly unrolled (belonging to a subroutine whose name I will indicate here with XXX, for privacy reasons which will appear in a moment), and then to restart reading

again all the cards from scratch, but a few cards at a time. After which, fortunately, the program compiled correctly, we got the punched cards with the desired binary code, and eventually we could bring the cards back in their cabinet.

Next time Tini came to CERN, I reported to him that the source code was still safe, adding *"despite my problems with XXX"*, making him immediately and briskly react — because, I learned, XXX was his wife's name.

The binary code was, of course, much more compact (you could hold it with one single hand), so that a job containing the binary code of SCHOONSCHIP and the user input to be processed by SCHOONSCHIP itself fulfilled the requirements of the fast jobs queue of the Computer Service. Thanks also to a few hints by Tini, I quickly became one of the first intensive and expert users of SCHOONSCHIP even if, due perhaps to the drawbacks of my Italian origin, I never succeeded (as many other people, indeed) in pronouncing correctly the word SCHOONSCHIP (apparently the choice of that name by the author was an intentional kind of *Dutch pride*).

Notwithstanding my pronunciation (and other linguistic) problems we soon became good friends, and Tini continued to assist me in the use of SCHOONSCHIP, but also to help me in many other ways in the subsequent years. I started acknowledging him with gratitude in my papers, but soon he almost forbade me to do that. Later on, he said that he did not want to share any responsibility in endorsing my results; I hope he was joking, but surely he did not like formalities and compliments.

Once Tini explained to me that in human relationships he was contrary to what he called *the posture of the cyclist* (or something similar), namely bowing your head to people above you and kicking down to those below (an attitude which I fully endorsed, even if it implied an obvious certification of my inferior condition compared to him).

I asked him several times to write recommendation letters to support my applications to various appointments. I remember in particular one occasion (the application for an important position at the Italian University) in which I did not get the job, and long after I was told (but I could not check) that his letter had a negative effect on the members of the Committee, because it was too disrespectful towards them.

Back to SCHOONSCHIP. He kept giving me the new versions of the program (each new release bearing the date April 1 of the current year), and implemented also, on my request and in the proposed format, the new command `Ratio` for carrying out the partial fractioning of two or more factors depending on the same variable. Another request which he satisfied was the introduction in SCHOONSCHIP of (long) variable names within square brackets, such as, for instance, `[x+a]`, corresponding, in the user's intentions, to $(x+a)$. Writing simply `(x+a)` in the input code of SCHOONSCHIP

would cause the program to open the brackets at its first occurrence, while it could be convenient to keep it in the compact form `[x+a]` for a while, leaving to the user the possibility of replacing `[x+a]` by `(x+a)` only when it fits within the following algorithm.

Let me add here a short digression on computer algebra. SCHOONSCHIP did not know anything about the calculus machinery, such as standard functions, differentiation *etc.*, but it was very easy for the user to implement all the necessary relations, when useful to him, as user-supplied instructions in the input code. That can be particularly helpful when, trying to figure out a new algorithm, it is essential to keep full control of each step.

Another feature of SCHOONSCHIP, of huge convenience for the user, was the possibility of printing a large output (in particular: an intermediate large output, usually much larger than the final result) suitably bracketed, with the most relevant variables out of brackets for ease of reading (and figuring out what to do next). I will not enter in any detailed comparison of SCHOONSCHIP with the more modern computer algebra programs now in use; let me, however, refer to [1] for the possible influence of Veltman and SCHOONSCHIP on Mathematica.

According to Tini's Nobel Lecture, he started planning SCHOONSCHIP at CERN in the early sixties of the last century, for processing with the help of a computer the algebra of the Feynman graphs arising in a model of weak interactions mediated by a vector boson at which he was working.

In 1963, two other people were also at CERN, with similar ideas, Michael Levine and Tony Hearn. Tony Hearn, the most computer-science oriented, decided to write his program, REDUCE, in LISP language, while Michael, more practical, had already a working program written in Fortran (with some Assembler insertions as well: compilers at that time were somewhat rudimental and allowed for the mixture), which he was using for his $(g-2)$ QED calculations. That code, later called ASHMEDAI [2], was the Fortran version of a 1961 assembly language code originally written for his Ph.D.-1963 Thesis [3] (which, coincidentally, also involved vector boson mediated weak interaction).

The CERN mainframe at that time was an IBM-7094, and Tini told me that he decided to avoid the inefficiency of the existing compilers by writing SCHOONSCHIP in IBM Assembler language. Such a choice may sound quite odd nowadays, but it must be remembered that in those years, one of the strongest limitations of computers, including the largest available mainframes, was the memory (for lack of space, only upper case letter were allowed), and Tini perceived compilers as responsible for a blameful waste of memory and slowdown of execution speed.

Shortly afterwards, CERN got a CDC-6600 (and later, the smaller CDC-6400 referred to above) as the mainframe of the laboratory — and Tini, as he told me, converted immediately his code to the Assembler of the new computer. To recall again the shortage of memory at that time, these machines had just 64 kilowords of 60 (not 64) bit, 8 registers, instructions of 15 bit (so that a word could contain up to four instructions) and a particularly fast cache memory of 8 words, in which Tini was proudly able to squeeze critical parts of the code to speed up the execution time. The CDC had also a fabulous (for the time) arithmetic unit. It was kind of moral duty to exploit it at the best, so Tini expressed the numerical coefficients of SCHOONSCHIP as double precision (2 words of memory) floating numbers. For "obvious aesthetical reasons" the numbers were however printed as rational numbers (ratios of integers); the conversion of format was performed by using the first 20 digits of the floating format for guessing the continuous fraction expansion of the number, and the remaining digits as a check.

Quickly, I became a SCHOONSCHIP addict, unable to do anything without it; for a short while, Tini gave me also the prestigious (for me) assignment to distribute SCHOONSCHIP, on his directions, to the interested people (one of them was his student Peter van Nieuwenhuizen). As a matter of fact, I was somewhat surprised at that time that it took so long for computer algebra to become a common tool in Elementary Particle Physics.

In 1969, at the expiration of my CERN fellowship, I moved to Pisa, where Riccardo Barbieri, known to me from his undergraduate years, joined the collaboration with Goyi Mignaco; immediately, I introduced Riccardo to the use of the program (and shortly afterwards, to Tini as well).

However, in Pisa the use of SCHOONSCHIP became more difficult. The problem was that SCHOONSCHIP, written in CDC Assembler, could run only on the very good, but relatively rare, CDC machines. The nearest available CDC-6600 was at the CINECA, a computing center located in Bologna. Nothing like remote terminals to mainframes existed at that time, and we had to drive periodically from Pisa to Bologna for running our jobs on that machine. In 1970, I moved to Bologna; since then, the access to the CDC computer became easier, and I proposed it as a standard tool to all my subsequent collaborators, starting from Michele Caffo and Sandro Turrini.

At the beginning of the 70s, a student of Tini, Hugo Strubbe, got a job at CERN-DD, the Computer Division of CERN, for implementing a version of SCHOONSCHIP for IBM mainframes. The idea was to translate directly the source CDC-Assembler code into IBM-Assembler by means of a suitable program written in PL-1 Language [4]. In 1977, the IBM version was working, and I used it in a few different places (I still remember the torture of the JCL commands of the IBM operating system).

CERN had at hand the obvious option of supporting SCHOONSCHIP more or less officially, and of proposing it as a general-purpose computer algebra program for the scientific community. On the contrary, for reasons unknown to me (a Tini's failure in *scientific diplomacy*?), CERN decided to *discontinue* Strubbe's appointment. To my knowledge, Strubbe quit CERN to join PRIME (more exactly: PR1ME), a newborn company then producing minicomputers (small mainframes, fully replaced in the 90s by the PC's we have now).

In retaliation, Tini *discontinued* the availability of SCHOONSCHIP on all CERN's CDC computers. That was somewhat later discovered by Michele Caffo. He was at the University of Geneva on sabbatical, did not find SCHOONSCHIP on the CERN CDC, and installed the copy we were using at the Bologna's CDC. Some other people used it, till eventually also Tini (then at Ann Arbor) learned of it — and called me, urging us to remove *his* SCHOONSCHIP from CERN's CDC, which was done immediately.

In 1976, I spent a month at the CMU, Pittsburgh, for collaborating with Michael Levine. We used intensively his program ASHMEDAI. As the source code was in Fortran, Michael Levine had no problem in letting it run on the department computer, which at that time was a UNIVAC-1108. ASHMEDAI, presumably the very first working computer algebra program, was written by the author for his personal use in a "no frill" style. I remember that it allowed products, but not powers; given a variable, say `X`, it was quite natural to call `IX` its inverse, and to provide the program with the instruction `SUBST X*IX=1`. All that may sound clumsy but, in fact, ASHMEDAI was very robust and efficient, allowing for user-defined macros and recursive insertion of files, which speeded up remarkably the writing of input code and the execution time.

By then, it became evident (to me; presumably to everybody) that the non-portability of SCHOONSCHIP, written in CDC Assembler, was a serious shortcoming. Maybe also Tini realised it, and in 1983 he decided to rewrite once more SCHOONSCHIP for the Motorola-68k line of microprocessor — but again in machine code. To be more precise, he started by writing himself an Assembler for that CPU, for the obvious reason that he considered unsatisfactory the already available Assemblers. His Assembler, he told me, was a success, to the point that he could *sell* it to some software company. After which he quickly wrote the 68k version of SCHOONSCHIP. For completeness, he wired also inside a small wooden box a personal PC with a Motorola CPU (the box needed, however, external display and keyboard), on which SCHOONSCHIP was running. I still remember him proudly carrying the box on his arm along the CERN corridors, showing around his solution to the *software portability* problem.

More details on the M68k version of SCHOONSCHIP [5], together with the picture of a later version of Tini's personal PC, can be found at the link http://www-personal.umich.edu/~williams/archive/schoonschip/

In the late 80s, my Bologna's group bought the 68k version of SCHOONSCHIP and an Atari ST, with the Motorola CPU and 1 MB of RAM memory (a record at that time) to run it. Everything worked properly — but due to the strong input–output and storage limitations of the hardware, the overall usefulness of the tool turned out to be somewhat limited (later on, we had access to a Sun workstation, also with a 68k CPU, with similar results).

In 1983–84, Michael Levine invited me again at CMU for a sabbatical. At that time the CMU Physics and Chemistry Departments had already a pair of VAX-780 computers (in fact, one of them was the very first), and again we used the Fortran-coded ASHMEDAI on that machine without any problem. As a matter of fact, shortly afterwards, Michael gave me also a binary copy of his program, which I used later on many VAX machines as they started spreading around Italy in the research centers of the INFN, the Italian High Energy Physics Agency.

For a while, I asked Tini to convert the 68k code of SCHOONSCHIP into VAX code, more or less along the lines of the work done years before by Hugho Strubbe for the conversion from CDC to IBM, but he explained to me, with great clarity and patience, how difficult, if not impossible at all, was that conversion, because one of the code was big-endian, while the other was small-endian (a technicality related to the order, different in the two CPU's, of the bits within a byte and of the bytes within the words assigned to a variable). As C was spreading as a kind of universal programming language, I then started to ask Tini to rewrite again SCHOONSCHIP in C. Needless to say, he always rejected my proposal, illustrating to me every time the unacceptable inefficiency of compilers.

Jos Vermaseren shared essentially my opinion that SCHOONSCHIP had to be rewritten in a machine-independent language (not in Assembler code), but instead of just asking or complaining, he decided to do himself the job, and started the job of rewriting SCHOONSCHIP, *without changes* (or almost), at first (1984) in Fortran-77, but afterwards and until completion (1989) in C. In the course of the work, the result, rather than a strict conversion, evolved into a kind of *free translation* into a new, increasingly independent program, which Jos called FORM [6].

One Sunday afternoon (end of summer 1989?), I was invited to attend in Tini's office at CERN an outspoken discussion between him and Jos on the transition from SCHOONSCHIP to FORM, presumably on the compliance between the two programs. The conversation began in English, but quickly became a quite heated discussion in Dutch, whose details I could not catch (my impression was that Tini did not agree with the undue autonomy of Jos).

At any rate, shortly afterwards (1989) FORM was available as a natural successor of SCHOONSCHIP, running practically on any machine. From the user point of view, I moved from SCHOONSCHIP to FORM almost without realising it, and thanks also to the kindness of Jos for always providing me with the latest versions, since then I started using FORM immediately on all the VAX, VAX Station Minicomputers, *etc.*, to which I had access in the following years, up to the PC's of the present time.

3. Largest time equation and cutting rules

In 1971 Tini, was invited to a workshop in Marseille organized by Andrea Visconti. As by some reason he could not go, he gave me (I felt highly honored) the assignment of presenting his contribution in his stead. The contribution, which I will discuss in this section, was the development (I guess) of previous ideas already contained in his Ph.D. Thesis [7], concerning the derivation of the Cutkosky cutting rules [8] with the Largest Time Equation.

Let $D(m, x_1 - x_2)$ be, in the coordinate representation, the scalar propagator of mass m from the point x_1 to x_2, with

$$D(m,x) = \int \mathrm{d}\vec{k}\ \mathrm{e}^{i\vec{k}\vec{x}} \int \mathrm{d}k_0\ \mathrm{e}^{-ik_0x_0} \frac{-i}{\vec{k}^2 - k_0^2 + m^2 - i\epsilon}\,, \tag{1}$$

where the number of dimensions of $\vec{k}$, which does not play any role, is not specified (and the overall (2π) factors omitted). One has the customary formulas

$$D(m,x) = \theta(x_0)D^+(m,x) + \theta(-x_0)D^-(m,x)\,, \tag{2}$$

where $D^+(m,x), D^-(m,x)$ are the positive and negative frequency parts of the propagator

$$\begin{aligned} D^+(m,x) &= \int \mathrm{d}\vec{k}\ \mathrm{e}^{i\vec{k}\vec{x}} \int \mathrm{d}k_0\ \mathrm{e}^{-ik_0x_0}\ \theta(+k_0)\delta\left(\vec{k}^{\,2} - k_0^2 + m^2\right)\,, \\ D^-(m,x) &= \int \mathrm{d}\vec{k}\ \mathrm{e}^{i\vec{k}\vec{x}} \int \mathrm{d}k_0\ \mathrm{e}^{-ik_0x_0}\ \theta(-k_0)\delta\left(\vec{k}^{\,2} - k_0^2 + m^2\right)\,, \end{aligned} \tag{3}$$

so that

$$\begin{aligned} D^+(m,x) &= D^-(m,-x) = (D^-(m,x))^*\,, \\ D(m,x) &= D(m,-x)\,, \\ D^*(m,x) &= \theta(x_0)D^-(m,x) + \theta(-x_0)D^+(m,x)\,. \end{aligned} \tag{4}$$

For the present discussion, an n-points (or n-vertices) Feynman amplitude $A(x_1, x_2, \ldots, x_n)$ is just the product of the factor $(i)^n$ times a suitable set of propagators $D(m_k, x_i - x_j)$, where m_k is a mass and x_i, x_j a pair of two of the n points (such an amplitude can then be multiplied by the wave functions of the external particles and integrated over all the points).

A simple 2-point amplitude is, for instance, (apart from an overall sign)

$$A(x_1, x_2) = D(M, x_1 - x_2)\ D(m, x_2 - x_1)\,, \tag{5}$$

the so-called *bubble* of masses M, m.

Veltman's idea was to consider the sum of all the associated amplitudes (or graphs) obtained from the original graph by underlining any combination of the points (or circling the vertices); in the above example, such a sum reads

$$\Sigma = A(x_1, x_2) + A(\underline{x_1}, x_2) + A(x_1, \underline{x_2}) + A(\underline{x_1}, \underline{x_2})\,, \tag{6}$$

with a total of 2^n terms, in general, for an n-point amplitude. The amplitudes with underlined points are obtained from the original one by multiplying it by a factor (-1) for each underlined point and then underlining the corresponding coordinates within the propagators. In the previous example, that gives

$$\begin{aligned} A(\underline{x_1}, x_2) &= -D(M, \underline{x_1} - x_2)\ D(m, x_2 - \underline{x_1})\,, \\ A(x_1, \underline{x_2}) &= -D(M, x_1 - \underline{x_2})\ D(m, \underline{x_2} - x_1)\,, \\ A(\underline{x_1}, \underline{x_2}) &= +D(M, \underline{x_1} - \underline{x_2})\ D(m, \underline{x_1} - \underline{x_2})\,. \end{aligned} \tag{7}$$

In turn, the definition of the propagators with underlined arguments is

$$\begin{aligned} D(M, \underline{x_1} - x_2) &= D^+(M, x_1 - x_2)\,, \\ D(M, x_1 - \underline{x_2}) &= D^-(M, x_1 - x_2)\,, \\ D(M, \underline{x_1} - \underline{x_2}) &= D^*(M, x_1 - x_2)\,. \end{aligned} \tag{8}$$

The associated amplitude in which all the points are underlined is then the complex conjugate of the original amplitude, so that for $A(\underline{x_1}, \underline{x_2})$ appearing in Eqs. (6) and (7), one has

$$A(\underline{x_1}, \underline{x_2}) = A^*(x_1, x_2)\,. \tag{9}$$

In a Feynman graph, a propagator $D(M, x_1 - x_2)$ is usually represented as a continuous line joining x_1 to x_2, or which is the same x_2 to x_1 (the propagator line is not oriented); similarly, $D(M, \underline{x_1} - x_2)$ can be represented by a *cut* line, oriented from x_2 to x_1, $D(M, x_1 - \underline{x_2})$ by a cut line oriented from x_1 to x_2, and $D(M, \underline{x_1} - \underline{x_2})$, finally, by a continuous non-oriented line, like $D(M, x_1 - x_2)$.

Consider now Eq. (6) in the case in which x_{10} is the largest of the two times x_{10}, x_{20}. According to Eqs. (5), (7) and (8), one has in general

$$\begin{aligned} A(x_1,x_2)+A(\underline{x_1},x_2) \;=\; & [D(M,x_1-x_2)\; D(m,x_2-x_1)] \\ & +[-D^+(M,x_1-x_2)D^-(m,x_1-x_2)]\,, \end{aligned}$$

but if $x_{10}>x_{20}$, Eq. (2) gives

$$\begin{aligned} D(M,x_1-x_2)&=D^+(M,x_1-x_2)\,,\\ D(m,x_2-x_1)&=D^-(m,x_2-x_1)\,, \end{aligned}$$

so that, for $x_{10}>x_{20}$,

$$A(x_1,x_2)+A(\underline{x_1},x_2)=0\,.$$

Similarly, one finds

$$A(x_1,\underline{x_2})+A(\underline{x_1},\underline{x_2})=0\,,$$

so that, for $x_{10}>x_{20}$,

$$\Sigma=0\,.$$

Similarly, for $x_{20}>x_{10}$, one finds

$$\begin{aligned} A(x_1,x_2)+A(x_1,\underline{x_2})&=0\,,\\ A(\underline{x_1},x_2)+A(\underline{x_1},\underline{x_2})&=0\,, \end{aligned}$$

so that, again

$$\Sigma=0\,,$$

i.e. the equation $\Sigma=0$ always holds.

From the very definition of Σ, Eq. (6), one has

$$A(x_1,x_2)+A(\underline{x_1},\underline{x_2})=-A(x_1,\underline{x_2})-A(\underline{x_1},x_2)\,. \tag{10}$$

According to Eq. (9),

$$A(x_1,x_2)+A(\underline{x_1},\underline{x_2})=2\mathrm{Re}A(x_1,x_2)\,,$$

so that Eq. (10) expresses the real part of $A(x_1,x_2)$ in terms of suitably underlined amplitudes, containing suitably cut propagators.

If one defines $T=iA$, the real part of $A(x_1,x_2)$ is, obviously, the imaginary part of T. Equation (10) then gives the imaginary part of T (*i.e.* the real part of A) in terms of an expression containing *cut propagator lines*,

which can be represented as Feynman graphs where some of the propagator lines are suitably *cut* (in the general case, the terms contributing to the imaginary part contain, besides the *cut propagators*, uncut propagators as well).

That is the Cutkosky *cutting rule*, for the *imaginary part* of the amplitude T of the graph. In Veltman's derivation, the *cutting rules* are obtained by a few simple algebraic steps rather than from the analyticity properties of the amplitudes.

As a further comment, the imaginary part of a graph is related to its discontinuity and to the unitarity of the theory, so that those cuts are also called *unitarity cuts*, and the cut propagator corresponds to a particle on the mass shell with a defined sign of the energy, fixed by the orientation of the cut propagator line, according to Eqs. (3).

As a digression, let me give another elementary derivation of Eq. (10) in the momentum representation.

In the c.m.s. of the external vector $p = (E, \vec{0})$, the two propagators of mass m, M can be written as

$$\begin{aligned}
\frac{-i}{D(k,m) - i\epsilon} &= \frac{-i}{\vec{k}^2 - k_0^2 + m^2 - i\epsilon} \\
&= \frac{i}{2A}\left(\frac{1}{k_0 - A + i\epsilon} - \frac{1}{k_0 + A - i\epsilon}\right), \\
\frac{-i}{D(p-k,M) - i\epsilon} &= \frac{-i}{\vec{k}^2 - (E - k_0)^2 + M^2 - i\epsilon} \\
&= \frac{i}{2B}\left(\frac{1}{k_0 - E - B + i\epsilon} - \frac{1}{k_0 - E + B - i\epsilon}\right),
\end{aligned}$$

with

$$A = \sqrt{\vec{k}^2 + m^2}\,, \qquad B = \sqrt{\vec{k}^2 + M^2}\,,$$

so that the considered amplitude is

$$\begin{aligned}
T\left(E^2\right) &= i \int \mathrm{d}\vec{k}\mathrm{d}k_0 \, \frac{1}{D(k,m) - i\epsilon} \frac{1}{D(p-k,M) - i\epsilon} \\
&= i \int \mathrm{d}\vec{k} \frac{1}{4AB} \mathrm{d}k_0 \left(\frac{1}{k_0 - A + i\epsilon} - \frac{1}{k_0 + A - i\epsilon}\right) \\
&\quad \times \left(\frac{1}{k_0 - E - B + i\epsilon} - \frac{1}{k_0 - E + B - i\epsilon}\right). \qquad (11)
\end{aligned}$$

By contour integration considerations in the complex k_0 plane, one has the *integration by contour identities*

$$\int \mathrm{d}k_0 \; \frac{1}{k_0 - A + i\epsilon} \; \frac{1}{k_0 - E - B + i\epsilon} = 0\,,$$
$$\int \mathrm{d}k_0 \; \frac{1}{k_0 + A - i\epsilon} \; \frac{1}{k_0 - E + B - i\epsilon} = 0\,, \tag{12}$$

because the integrands have their singularities on the same side of the real axis; similarly, one has also

$$\int \mathrm{d}k_0 \; \frac{1}{k_0 - A + i\epsilon} \; \frac{1}{k_0 - E + B + i\epsilon} = 0\,,$$
$$\int \mathrm{d}k_0 \; \frac{1}{k_0 + A - i\epsilon} \; \frac{1}{k_0 - E - B - i\epsilon} = 0\,. \tag{13}$$

Quite in general,

$$\frac{1}{x - i\epsilon} = \frac{x}{x^2 + \epsilon^2} + i\frac{\epsilon}{x^2 + \epsilon^2} = P\left(\frac{1}{x}\right) + i\pi\delta(x)\,, \tag{14}$$

where $P(1/x)$ is the principal value, and the parting of the real and imaginary parts is made explicit. One can insert Eq. (14) into the previous equations, and then use Eqs. (12) and (13) for eliminating the products of two principal values from Eq. (11). The result is

$$\begin{aligned} T\left(E^2\right) \;=\; & \int \mathrm{d}\vec{k}\,\mathrm{d}k_0 \; \frac{1}{2AB} \\ & \times \Bigg[i\pi^2 \Big(\delta(k_0 - A)\delta(k_0 - E + B) + \delta(k_0 + A)\delta(k_0 - E - B) \Big) \\ & + P\left(\frac{1}{k_0 + A}\right) \delta(k_0 - E - B) - P\left(\frac{1}{k_0 - A}\right) \delta(k_0 - E + B) \Bigg]\,, \end{aligned} \tag{15}$$

from which, in particular, one obtains (with minor aesthetical polishing)

$$\begin{aligned} \mathrm{Im}T\left(E^2\right) \;=\; & 2\pi^2 \int \mathrm{d}\vec{k}\,\mathrm{d}k_0 \\ & \times \Big[\theta(+k_0)\delta\Big(\vec{k}^2 - k_0^2 + m^2\Big)\theta(E - k_0)\delta\left(\vec{k}^2 - (E - k_0)^2 + \; M^2\right) \\ & + \theta(-k_0)\delta\Big(\vec{k}^2 - k_0^2 + m^2\Big)\theta(k_0 - E)\delta\left(\vec{k}^2 - (E - k_0)^2 + \; M^2\right) \Big]\,. \end{aligned} \tag{16}$$

End of the digression.

In 1972, Tini was invited by Luigi A. Radicati to give a course on Feynman graphs at the Scuola Normale of Pisa, which I could attend only once in a while as I had already moved to Bologna. Riccardo Barbieri took the notes of the course, later used by Tini and G. 't Hooft for the CERN Yellow Report DIAGRAMMAR [9] (Acknowledgements read: "The authors are deeply indebted to Dr. R. Barbieri, who helped in writing a first approximation to this report."). The yellow report contains, among many other subjects, a paragraph on the Largest Time Equation and a Chapter on Renormalisation and the Continuous Dimensional Regularisation, to which I will refer in the next section.

The "private lecture" on the Largest Time Equation which Tini taught me in 1971, gave me a growing confidence in using intensively cutting rules in all my subsequent radiative correction calculations.

In 1981, as a kind of self-pedagogical paper [10], I wrote at last with some details of what I had learned in that 1971 conversation (concerning also a related algebraic derivation of dispersion relations). To my great (and pleasant) surprise, on 11/07/2005, I received a mail from Tini:

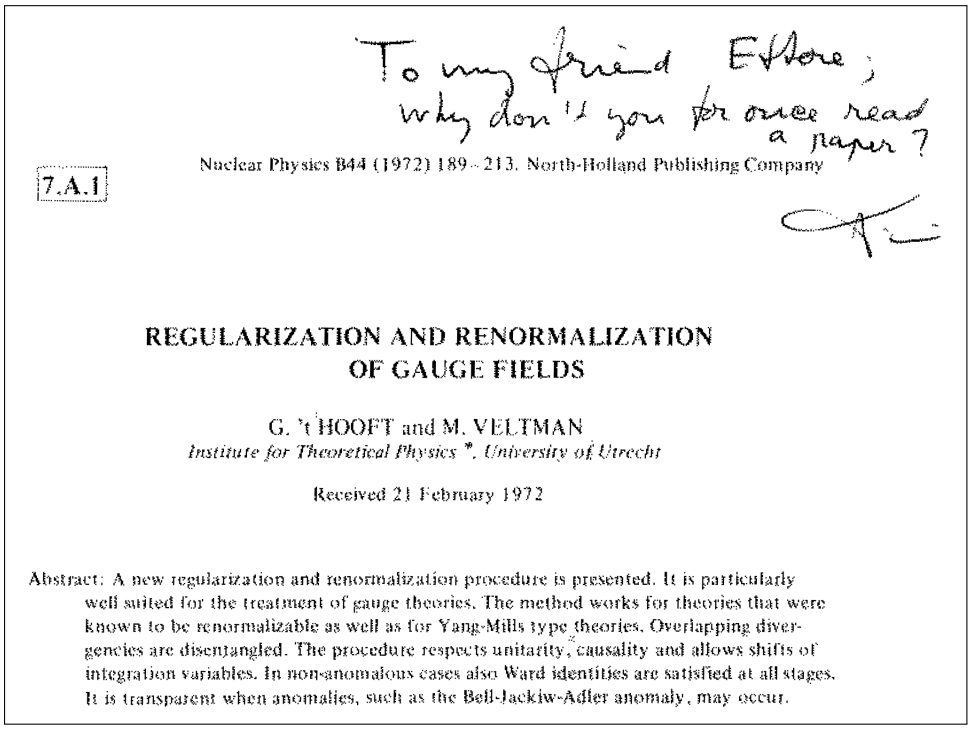
> *Dear Ettore.*
> *I am giving lectures in Columbus, Ohio and I would like to make available a pdf version of your paper in Helv. Phys. Acta of 1981 ... Is that ok with you?*
> *In addition, I know that there are some misprints. Do you know any that I should add to the pdf file ? ...*
> *Tini*

Fortunately, the misprints were just misprints — and I consider that mail the highest recognition I ever received in my whole scientific life.

4. The continuous dimensional regularisation

The paper by Tini and G. 't Hooft introducing the *continuous dimensional regularisation* appeared at the beginning of 1972 [11], just before the Pisa lectures referred to in the previous section. Tini was so kind to send me a reprint of his paper *with a dedication*, see Fig. 1.

Despite that dedication, I did not pay too much attention to it, because I had learned how to use the Pauli–Villars regularisation with dispersion relations (and cut propagators) in the evaluation of Feynman graphs, and I did not want to change (just out of laziness, perhaps). Furthermore, I was disturbed by the fact that in the new scheme, the same parameter d (the continuous number of dimensions) was used for regularising both the ultraviolet divergences with $d < 4$ and the infrared divergences with $d > 4$. (Only many years later, I learned that you can start with spacelike


To my friend Ettore;
why don't you for once read
a paper?

Nuclear Physics B44 (1972) 189–213. North-Holland Publishing Company

7.A.1

REGULARIZATION AND RENORMALIZATION
OF GAUGE FIELDS

G. 't HOOFT and M. VELTMAN
*Institute for Theoretical Physics *, University of Utrecht*

Received 21 February 1972

Abstract: A new regularization and renormalization procedure is presented. It is particularly well suited for the treatment of gauge theories. The method works for theories that were known to be renormalizable as well as for Yang-Mills type theories. Overlapping divergencies are disentangled. The procedure respects unitarity, causality and allows shifts of integration variables. In non-anomalous cases also Ward identities are satisfied at all stages. It is transparent when anomalies, such as the Bell-Jackiw-Adler anomaly, may occur.

Fig. 1. The continuous dimensional regularisation paper.

Mandelstam variables and small d, regularise the UV divergences, continue analytically up to large enough $d > 4$, switch then to physical Mandelstam variables and fix at last the IR divergences arising for $d \to 4$.)

In 1992 David Broadhurst introduced me to the *integration by parts identities* (ibp-id's) [12]; again, I did not attach too much attention to them, with the (wrong) perception that they were useful only in the massless case, while at that time I was working with Stefano Laporta on the $(g-2)$ of the (non-massless!) electron at 3 loops in QED.

Let me recall here some feature of the ibp-id's for a generic Feynman graph with n external particles of momenta p_i, $(i = 1, \ldots, n)$, l loop d-dimensional internal momenta q_j, $(j = 1, \ldots l)$ and N propagator lines, with denominators $D_1, D_2, \ldots, D_N$. One can form nl scalar products of the form of $(p_i \cdot q_j)$ and $l(l+1)/2$ of the form of $(q_j \cdot q_{j'})$; the N denominators contain N different linear combinations of those scalar products, so that there is a set of $s = l(2n + l + 1)/2 - N$ *irreducible* scalar products, say $S_k, (k = 1, \ldots, s)$, which cannot be expressed as a linear combination of the N scalar products contained in the N denominators.

The most general scalar integral associated with the graph can be written as

$$I(p,a,b) = \int \mathrm{d}q_1 \ldots \mathrm{d}q_l \; A(p,q,a,b)\,, \tag{17}$$

with

$$A(p,q,a,b) = \frac{(S_1)^{a_1} \ldots (S_s)^{a_s}}{(D_1)^{b_1} \ldots (D_N)^{b_N}}\,, \tag{18}$$

where the arguments p, q of A stand for the two sets of the vectors p_i, q_j (in d continuous dimensions), a is any set of non-negative integers $(a_1, \ldots, a_s)$, and b any set of positive integers $(b_1, \ldots, b_N)$ (the elements of a can vanish, while those of b are all positive). For any $A(p,q,a,b)$, one has then the $l(n+l)$ ibp-id's

$$\int \mathrm{d}q_1 \ldots \mathrm{d}q_l \; \frac{\partial}{\partial q_{j\mu}} \Big(v_{r\mu} A(p,q,a,b) \Big) = 0\,, \tag{19}$$

where the vector v_r is any of the $(n+l)$ external or loop momenta p, q, and the identities hold for the "magic" properties of the continuous dimensional regularisation (which allow one to integrate by parts disregarding the vanishing of the endpoint contributions).

Assume now that you want to evaluate the N propagator integral of Eq. (17), related to an N propagator graph, for some given sets of the indices a, b, say, for definiteness, with all the elements of a equal to zero and all the elements of b equal to 1, and you want to use Eq. (19) with the chosen sets of the indices (a, b) to get a hint. As a consequence of the differentiation of the denominators, each of the $l(n+l)$ identities which you obtain involves, besides your original integral, several new integrals, whose integrands are typically equal to the original integrand times one more scalar product (to be expressed in terms of the irreducible scalar products and of the denominators) and divided by one of the denominators. (If a numerator is also present, you can get from its differentiation some additional terms, to be treated in a similar way). After minor algebraic rearrangements, one obtains, in general, a combination of integrals of the same family but with some of the indices a or b increased or decreased by one unit with respect to the integral originally considered.

Let us observe here that if in one of the new integrals one of the b indices decreases from 1 to zero, that integral is an amplitude related to a *subtopology* of the original graph, *i.e.* to a graph with $N-1$ propagators where one propagator is missing, to be considered simpler (and already known) in a bottom-up approach to the original N propagator problem.

Summarising, one obtains $l(n+l)$ relations between several amplitudes of the form of Eq. (17), one of which may be the original integral, while all the others are of the same family but with different sets of indices a, b.

In particularly simple cases, it may happen that all the newly-introduced amplitudes are related to subtopologies, so that the identities can be used to fully express the original amplitude in terms of subtopology amplitudes but, in general, that is not the case; the number of the newly-introduced integrals is larger, in general, than the number $l(n+l)$ of the identities, and one ends up by expressing the original unknown integral in terms of many equally unknown new integrals, of similar or even more complicated form.

By repeating the procedure for the newly appearing integrals, one keeps obtaining more and more new unknown integrals, in a kind of *run away* pattern ...

But fortunately, it is not so! As Stefano Laporta explained to me, if you do not give up, and keep writing new identities for the new appearing integrals, after a while (which, however, depends on the problem you are dealing with!), the number of the identities becomes larger than the number of the unknown integrals (that is essentially the *Laporta algorithm* [13]). At that point, you have just to solve the identities (a highly demanding task, usually) to find that *all* the N propagator amplitudes are linear combinations of a small number (a few units, typically) of N-propagators *Master Integrals* (MIs) and of the MIs of the subtopologies.

Quite in general, the coefficients of the linear combinations are ratios of two polynomials (with integer coefficients) in the Mandelstam variables of the problem and in the continuous dimension d. As a minor comment, let me repeat that the system of the identities can be rather large and its actual solution extremely demanding from the computational point of view, but it becomes conceptually *trivial* in the limit of *infinite* computer algebra power. Further, the actual choice of the MIs is not unique, as different sets of MIs can be more appropriate for accomplishing different purposes (such as analytic or numerical evaluation).

We quickly realised that, although the ibp-id's were of no direct help for the actual evaluation of the integrals, they were a powerful tool for checking results or, better, for strongly reducing the number of the independent integrals to be separately evaluated. In a kind of *reverse engineering* performed after the completion of the analytic evaluation of the 3 loops $(g-2)$ of the electron, Stefano Laporta found indeed that the whole calculation required, besides some *trivial algebra*, the actual evaluation of 18 MIs only.

But the existence of the MIs can be used also in other ways (1998). Consider a graph depending, for simplicity, on a single external momentum p only, and therefore on a single Mandelstam variable $s = -p^2$, and assume that it has n master integrals $M_i(s), (i = 1, \ldots, n)$, given by the Feynman amplitudes

$$M_i(s) = \int \mathrm{d}q \; A_i(p, q) \,, \tag{20}$$

where dq stands for all the d-dimensional loop integrations (the number of loops is irrelevant here) and the $A_i(p,q)$ are amplitudes of the kind introduced in Eq. (18). One has

$$s\frac{\mathrm{d}}{\mathrm{d}s}M_i(s) = \frac{1}{2}p_\mu\frac{\partial}{\partial p_\mu}M_i(s)\,,$$

that is

$$s\frac{\mathrm{d}}{\mathrm{d}s}M_i(s) = \frac{1}{2}p_\mu\int \mathrm{d}q\ \frac{\partial}{\partial p_\mu}A_i(p,q)\,. \tag{21}$$

However, the integrand of the of r.h.s. is also a combination of amplitudes of the kind of Eq. (18), so that the integral is a combination of integrals like those of Eq. (17), which, in turn, are combinations of the MIs of the problem, Eq. (20).

The previous relation can then be written as

$$s\frac{\mathrm{d}}{\mathrm{d}s}M_i(d,s) = \sum_j^n C_{ij}(d,s)M_j(s) + N_i(d,s)\,, \tag{22}$$

where the dependence on the continuous dimension d is made explicit, the coefficients $C_{ij}(d,s)$ are ratios of two polynomial in s and d with integer coefficients, and the $N_i(d,s)$ are the contributions to the r.h.s. of Eq. (21) from the subtopologies of the graph (supposedly known in a bottom-up systematic approach to the problem).

Equation (22) is a first-order linear system of n inhomogeneous differential equations in the n unknown functions $M_i(d,s)$ with inhomogeneous (but supposedly known) terms $N_i(d,s)$.

The equations, which for convenience can be expanded in d around $d=2$ or $d=4$, can then be solved by the Euler method, which, in turn, relies on the solutions of the homogeneous part of the system involving the MIs only.

As usual, from the $n\times n$ system for the MIs, one can derive an equivalent n-order homogeneous differential equation for just one of the MIs, say (arbitrarily) for the *main* Master Integral.

To my great (and pleasant) surprise, in many cases of interest, that homogeneous equation was rather simple (sometimes shrinking to a trivial first order differential equation). Starting from that solution and then implementing Euler's algorithm, it is then relatively easy to express the solution of the inhomogeneous equation, *i.e.* the Feynman graph amplitude under consideration, in a closed analytic form, usually as the repeated integration of simple rational factors times Euler's polylogarithms and their generalisations (harmonic polylogarithms or, generally, repeated Kummer–Goncharov integrals).

However, it is not always so. In the case of the equal mass sunrise (a subtopology which appears in the 2-loops self-mass of the electron in QED, studied by Sabry [14] in 1962), the homogeneous equation for the main MI, say $M(s)$, is of second order, and cannot be solved in terms of the above polylogarithms.

It happens, however, that the inhomogeneous part of the differential equation for $M(s)$ is always real, while $M(s)$ can develop an imaginary part if $s > 9m^2$, where m is the electron mass; therefore, $\mathrm{Im}M(s)$ satisfies the associated homogeneous equation.

That imaginary part can be safely evaluated by the cutting rules related to the Largest Time equation discussed in Section 3; the result (apart from overall constants anyhow irrelevant when a homogeneous equation is considered) is nothing but the physical three body phase space at energy $\sqrt{s}$, which, in turn, is equal to a kinematical factor times the complete elliptic integral of the first kind.

That approach to the analytic evaluation of Feynman graph amplitudes (through the ibp-id's, the differential equation for the MIs, the solution of the associated homogenous equations and then the Euler's method for solving the full inhomogeneous equations) is by now a kind of accepted standard (see, for instance, [15] and [16] for more details and bibliography).

An essential step is, therefore, the evaluation of the solutions of the homogeneous equations. The trick used for the sunrise does not work in general, because in general also the subtopologies can develop an imaginary part, so that the imaginary part of the *main* MI does not satisfy the homogeneous equation. A more general, direct derivation of the homogeneous part of the equations would be therefore of great use.

To that aim, let us go back to the (scalar) Feynman propagator and write, as in Eq. (14)

$$\frac{1}{D - i\epsilon} = \frac{D}{D^2 + \epsilon^2} + i\pi \frac{\epsilon}{D^2 + \epsilon^2} = P\left(\frac{1}{D}\right) + i\pi\delta(D)\,, \tag{23}$$

where $P(1/D)$ is the principal value.

The $\delta(D)$ terms played an increasingly important role in the study of the properties of Feynman amplitudes (see, for instance, [17, 18]) and were called *generalised unitarity cuts* for the close analogy with the *unitarity cuts* of Section 3, with the important difference that those cuts select only one definite sign of the energy, while the *generalised unitarity cuts* include both signs.

By using Eq. (23), any graph with N propagators can be decomposed in 2^N terms, whose sum is, of course, equal to the considered graph. The term corresponding to the product of N δ-functions, in which namely all the propagators are replaced by the corresponding δ's, is called the *maximally*

cut amplitude, or, for short, the *maxcut* amplitude or simply the *maxcut*. (A first use of the maximal cut for differential equations purposes can be found in [19].)

It happens that the ibp-id's for a *maxcut* amplitude are equal to the ibp-id's valid for the original amplitude, where however all the inhomogeneous terms (coming from the subtopologies) are set to zero, because $D/(D - i\epsilon) = 1$, while $D\,\delta(D) = 0$.

However, there is a problem. In the case of the sunrise seen above, the *maxcut* is just a generalisation of the cuts giving the imaginary part, but in more general cases, it is found that the *maxcuts* give simply a vanishing result, because the integration region in the loop variables shrinks to the empty set, due to all the constraints imposed by the arguments of the δ-functions.

Loosely speaking, time and space components of the vectors give contributions of different sign to the arguments of the δ's, with just *a single* time component giving, say, a negative contribution against *many* space components, all giving positive contributions.

However, that imbalance can be modified. In a Feynman graph referring to a scattering amplitude, the physical momenta of the external particles span one time and only two space dimensions, say x and y, while the loop momenta span in addition also the third space component z. That z component can therefore be Wick-rotated from Euclidean to Minkowskian, without affecting the physical components and the value of the Feynman amplitude, but the arguments of the propagators *do* change, and each of the 2^N terms obtained with the decomposition Eq. (23) can change as well. In particular, the *maxcut* is no longer forced to vanish.

At any rate, under any allowed Wick rotation, the *integration by parts identities* remain valid, and the same applies to the differential equations, which rely on the ibp-id's.

Rather than Wick-rotating a single component, one can lump together that component and the usual Euclidean regularising components into $(d-3)$ Minkowskian continuous regularising components, and then evaluate the no longer vanishing *maxcut*; one obtains in that way the required solution of the homogeneous part of the differential equation [20].

That is one more extension of the use of Tini's *cut* graphs and *continuous dimensions* to the evaluation of Feynman graph amplitudes.

It is the author's pleasure to thank Michael Levine, Michele Caffo and Stefano Laporta who helped him in rearranging the dating of his memories.

REFERENCES

[1] S. Wolfram, «Tini Veltman 1931–2021, from Assembly Language to a Nobel Prize», in: «Stephen Wolfram Writings», 2021, vol. 2021/01.

[2] M. Levine, «ASHMEDAI: a symbol manipulation program», in: «Proceedings of the Third International Colloquium in Advanced Computing Methods in Theoretical Physics», 1973, Report Number CNRS-CPT-73-P-550(VOL.1).

[3] M.J. Levine, «Neutrino processes of significance in stars», Ph.D. Thesis, CalTech, 1963.

[4] H. Strubbe, «Converting SCHOONSHIP to IBM», Internal report DD/77/1, CERN, 1977.

[5] M.J.G. Veltman, D.N. Williams, «SCHOONSHIP '91», Tech. Rep., Michigan TU. Dept. Theor. Phys., Ann Arbor, MI, October, 1991.

[6] J. Vermaseren, «New features of FORM», arXiv:math-ph/0010025.

[7] M.J.G. Veltman, «Unitarity and causality in a renormalizable field the with unstable particles», *Physica* **29**, 186 (1963).

[8] R.E. Cutkosky, «Singularities and Discontinuities of Feynman Amplitudes», *J. Math. Phys.* **1**, 429 (1960).

[9] G. 't Hooft, M.J.G. Veltman, «DIAGRAMMAR», *NATO Sci. Ser. B* **4**, 177 (1974).

[10] E. Remiddi, «Dispersion Relations for Feynman Graphs», *Helv. Phys. Acta* **54**, 364 (1982).

[11] G. 't Hooft, M. Veltman, «Regularization and Renormalization of Gauge Fields», *Nucl. Phys. B* **44**, 189 (1972).

[12] K. Chetyrkin, F. Tkachov, «Integration by parts: The algorithm to calculate β-functions in 4 loops», *Nucl. Phys. B* **192**, 159 (1981).

[13] S. Laporta, «High-Precision Calculation of Multiloop Feynman Integrals by Difference Equations», *Int. J. Mod. Phys. A* **15**, 5087 (2000), arXiv:hep-ph/0102033.

[14] A. Sabry, «Fourth order spectral functions for the electron propagator», *Nucl. Phys.* **33**, 401 (1962).

[15] M. Argeri, P. Mastrolia, «Feynman Diagrams and Differential Equations», *Int. J. Mod. Phys. A* **22**, 4375 (2007), arXiv:0707.4037 [hep-ph].

[16] J.M. Henn, «Lectures on differential equations for Feynman integrals», *J. Phys. A* **48**, 153001 (2015), arXiv:1412.2296 [hep-ph].

[17] Z. Bern, L.J. Dixon, D.C. Dunbar, D.A. Kosower, «Fusing gauge theory tree amplitudes into loop amplitudes», *Nucl. Phys. B* **435**, 59 (1995), arXiv:hep-ph/9409265.

[18] R. Britto, F. Cachazo, B. Feng, «Generalized unitarity and one-loop amplitudes in $N = 4$ super-Yang–Mills», *Nucl. Phys. B* **725**, 275 (2005), arXiv:hep-th/0412103.

[19] A. Primo, L. Tancredi, «On the maximal cut of Feynman integrals and the solution of their differential equations», *Nucl. Phys. B* **916**, 94 (2017), arXiv:1610.08397 [hep-ph].

[20] E. Remiddi, «Maximal Cuts and Wick Rotations», in: «MathemAmplitudes: Intersection Theory & Feynman Integrals», 2021, vol. MA2019.