Order of Lenin
KELDYSH INSTITUTE OF APPLIED MATHEMATICS
Russian Academy of Science

A.W. Niukkanen
I.B. Shchenkov, G.B. Efimov

OPERATOR FACTORIZATION TECHNIQUE OF FORMULA DERIVATION IN THE THEORY OF SIMPLE AND MULTIPLE HYPERGEOMETRIC FUNCTIONS OF ONE AND SEVERAL VARIABLES

Moscow – 2003

A.W. Niukkanen, I.B. Shchenkov, G.B. Efimov. **Operator factorization technique of formula derivation in the theory of simple and multiple hypergeometric functions of one and several variables**

Abstract

It is shown that computation technique of the operator factorization method provides a simple and universal foundation for a new theory of hypergeometric series in any number of variables. Examples showing how the method works in practice are given. We also discuss the prospects of the method including the necessary modernization of Santra 2 system. We also give a preliminary analysis of the potentialities of using the superstructure Santra 3 over the Refal language as a basis for computer implementation of a globally universal program capable to perform the complete set of operations inherent in the factorization method.

The work was supported by Russian Foundation for Basic Research under grant 03-01-00708.

Contents

1. Introduction [1]

Mathematical models which play the role of theoretical substitutes for underlying natural or technological phenomena including those under supervision of computer–aided control systems give us an information about the phenomena in terms of functions having specialized structures. The vast majority, up to 95%, of the special functions involved in description of fundamentally important processes are connected with hypergeometric series which thereby give us a key to a substantial part of applied mathematics really needed by numerous users.

There is a further simple motive for our interest to the hypergeometric series. Following elementary functions these series present the most important class of functions naturally arising from simple operations over elementary functions. It suggests that along with elementary (EL) functions hypergeometric (HYP) series will inevitably turn into an obligatory component of modern software especially for the coming computer generation with prevailing role of "intellectual" interactive symbolic (SYMB) analysis over purely numerical (NUM) computations. Looking at the vertices (EL, NUM), (HYP, NUM), (EL, SYMB) and (HYP, SYMB) in functions-methods space one can say, loosely, that the first vertex pertains to the past, the next two symbolize the present and the last one relates to the future. It is just the future the present work is aimed at. For better visualization of what was said above we remind that Lozier and Olver maintain that even when one moves from (EL, NUM) to (HYP, NUM) "enormous gaps remain for functions having variable parameters in addition to the argument". On our move from (EL, SYMB) to (HYP, SYMB) the gaps would have been "twice as enormous" if we had not had the operator factorization method in our disposal.

The operator factorization method [?]–[?] greatly facilitating solution of many mathematical problems including the study of multiple hypergeometric series is the main object of our interest aimed at verification of possibility to use the main operations of the method for construction

[1]The work was supported by Russian Foundation of Basic Research under Grant 03-01-00708

a superstructure over the system of analytical transformations SANTRA [?]–[?].

The check–up is based on a detailed overveiw of the formal structure of the method. We show that the method uses a **new fundamental operation** over power series, **new analytical technique** and **new form of presenting results**. There is a strong evidence that the set of the basic operations of the method is complete within a wide range of problems including analysis of multiple hypergeometric series.

In Sec.2 we formulate some general statements which being trivial on their own can help us to better understand the position occupied by this method amongst other scientific methods.

In Sec.3 we describe a compact and transparent notational system that allows the structure of any multiple hypergeometric series to be comprehended easily in full detail.

In Sec.4 a brief overveiw of the operator factorization method is given. An Ω-multiplication operation, factorization formulas and general concepts underlying the method are discussed in brief. It is shown that the method permits us to use different analytical schemes as a basis for working out different computational algorithms.

The main factor determining the value of the method is that it makes use of **a limited set of operations** (see Sec.2 in Ref.[?]) sufficient to derive any formula in the theory of simple and multiple hypergeometric series. Moreover, as compared with indefinitely large number of conventional approaches this gives us the most simple and direct way to the desired result. Technically, there are 3 computational modes to reach the goal. First, we can use the set of operations manually, with pen and paper, to cover all the way to the final result without use of computer (see an example in Sec.5). Second, we can make an attempt to use the manual mode to obtain a set of macro–operations playing the role of basis relations for a given class of formulas and then generate all relations belonging to the class by computer–aided combining of the basis macro–operations. An example of such "bounded–universal" approach is given in Sec.6. Up to now the set of main operations was not implemented in the form of computer commands. Numerous examples of successful applications of the first and the second computational modes suggest that an interactive program using the complete set of operations of factorization method may give us a globally universal computer–aided "formula constructor" playing the role of a central core with respect to the partially universal programs and allowing the researcher to obtain the desired relation without any cease of the user session. Preliminary examination of feasibility of the globally universal computational mode with the help of the system of analytical transformations SANTRA is one of the main goal of the present work (see Sec.7).

2. General outline of the problem

2.1 Researcher and computer

The main body of scientific knowledge had been built before computer became a full member of scientific process. That's why scientific knowledge generated by centuries of human intellectual activity bears an indelible imprint of human ingenuity and human failings. Both these extremes put obstacles in our way to computerizable scientific knowledge.

Putting aside innumerable minor human drawbacks we stress that **results** are most important from **antropocentric** standpoint. On the contrary the **derivation rules** play the primary role for **computercentric** science. This difference serves as the main obstacle for teaching computer human tricks.

2.2. Sturm und Drang?

After principles governing a given domain of science begin to work their way there comes the time of *Sturm und Drang* epoch. The **negative results of this impetuous activity** being out of any reasonable control are not at all less than positive results.

The most grave consequence of the rush is that the domain being allegedly conquered by science is in fact conquered by **narrow layer of elite** capable to discern the main constitutive features of a new theory uprising from the primary intellectual chaos. As for the scientific community as a whole and even for the most part of the "elite" the domain transforms from a ”blank spot” into a ”black spot” of intellectual disorder. Only deep systematic revision of the domain may give us a hope to rectify the situation. A good theory is a simple theory requiring a will and industry rather than great abilities.

2.3. What kind of knowledge do we need

Specialization divides knowledge into scores of subfields all having methods of their own. The knowledge we need is that which helps different fields to combine and merge rather than come apart and disintegrate.

Mathematics is the language of Science. Analysis is the heart of mathematics and the concept of function is the heart of analysis. Therefore the **mathematical reference data** containing the most commonly used knowledge seems to present primary interest for researchers.

Computer-aided accumulating, processing and generating of mathematical knowledge varies from passive data bases to sophisticated knowledge bases which can produce information not present explicitly in the bases. Unfortunately, the use of computer algebra methods which can help us to build effective knowledge bases is limited by the fact that we meet a severe want of new sufficiently simple and universal analytical methods which would give us an effective basis for **symbolic manipulations**.

2.4. The NIST project: what is about computer algebra

Several years ago the National Institute of Standards and Technology (NIST, USA) initiated the project of a Digital Library of Mathematical Functions (DLMF) [**?**]. In an earlier web description of the project (http://math.nist.gov/DigitalMathLib//publications/nistir6297/) there had been stressed that ”standardization of mathematical knowledge requires growing use of **symbolic** and numerical software”, ”in the intervening decades **computer algebra and symbolics** have come into wide use” and ”many users will want easy-to-use support ... in numerical and **symbolic** computation”.

Then it is said that ”the role of symbolic computation in the DLMF is still being discussed” (!). The statement of P. Paule (DLMF associate editor) that ”the goal is a presentation of computer algebra concepts” with ”a new chapter on computer algebra", in view of what was said above, sounds quite unexpectedly. **Where has the computer algebra gone?**

2.5. Underestimated role of symbolic computing

Without **intensive practical use of computer algebra** methods any reference database would lose most of its value. The less objects are included in the base (34 in DLMF!) the less interest it presents for user. On the contrary, the giant book-like site would be crammed with misprints and authors' mistakes. Data level in the field of multiple hypergeometric series may exceed the capacity of a medium-size website. The giant site would confront with cross-referencing and search difficulties.

The knowledge created by computer at will of user is what we should aim at.

2.6. Misuse of symbolic manipulation

An attempt to attack the problem of symbolic manipulation of **simple** hypergeometric series by brute force method has been undertaken by C. Krattenthaler in his HYP and HYPQ packages. The results do not seem to completely justify the efforts. The formal combination of occasional formulas can hardly serve us as a reliable way to *interesting* new results. Moreover the efforts needed to obtain the new formulas seem to be rather tedious. "The philosophy of this package is: *Do it yourself*!" The disappointing idea implies that "you should be able to *control each step* in a series of manipulations by yourself"

What we really need is **an integrated complete set of commands** rather than a disorderly collection of **occasional tools**.

3. Hypergeometric series

3.1. Where the series arise

Hypergeometric series is probably what an applied mathematician or a theoretical physicist **more often runs into** when making his calculation work. Sometimes it happens **even without knowing it**!

The hypergeometric series are ubiquitous. They appear, under different guises, in elementary functions, differential equations, heat conduction, solid state dynamics, hydraulics, atomic and molecular physics, quantum mechanics, elementary particles (Feynman diagrams), combinatorics, group representations, algebraic geometry, etc. The late corresponding member of Academy of Science of the USSR, professor K.I. Babenko counted up to 1500 special function of hypergeometric type.

I.M. Gelfand and his co-authors remark that **hypergeometric series play an outstanding unifying role in science** and assert that "impetuous development of the theory of hypergeometric functions begun in the eighties of present century". In course of time the role of hypergeometric functions will become more significant due to their state–of–the–art and, especially, potential importance for applied sciences and engineering.

3.2. Why we can not use "standard" methods

The present level of standard methods in the theory of hypergeometric series can be clearly seen from Richard Askey's statement made in his preface to "Special Functions" (Reidel, 1984). He wrote "There are **many examples** (of special functions - A.N.) **and no single way** of looking at them that can illuminate all examples or even all the important properties of a single example of a special function".

The Askey's statement does not hold any more. The operator **factorization method just gives us a single way** of looking at scores of thousands of special functions and multiple hypergeometric series. Moreover **it allows us to "computerize" the theory** of these functions in a two-fold way. Our main goal is to discuss these ways and their relation to the existing computer - aided approaches to accumulating, processing and generating of scientific knowledge. By the example of the operator factorization method one can see one of the ways making the structure of knowledge easily accessible to computer.

3.3. Hypergeometric series: notation

1. Generalized hypergeometric series in one variable of type $A//B$:

$$F\left[\begin{matrix} a^1, \ldots, a^A \, ; \, x \\ b^1, \ldots, b^B \end{matrix}\right] = \sum_{i=0}^{\infty} \frac{(a^1, i) \, \ldots \, (a^A, i)}{(b^1, i) \, \ldots \, (b^B, i)} \, \frac{x^i}{i\,!}$$

where Pochhammer symbol (a, i) is

$$(a, i) = a(a+1)\cdots(a+i-1) = \Gamma(a+i)/\Gamma(a).$$

2. Contracted notation

$$F_B^A[\mathbf{d}; x] = \sum_{i=0}^{\infty}(\mathbf{d}, i)\,\frac{x^i}{i\,!} = \sum_{i=0}^{\infty}\frac{(\mathbf{a}, i)}{(\mathbf{b}, i)}\,\frac{x^i}{i\,!}.$$

Numerator and denominator sets of parameters are $\mathbf{a} = [a^1, \ldots, a^A]$, $\mathbf{b} = [b^1, \ldots, b^B]$ and $\mathbf{d} = \mathbf{a}//\mathbf{b}$ is a double set of parameters. Except of the above definitions explicit summations are not used in the method.

3. Multiple series: conventions

a. Complex parameter $< \alpha \mid m_1, \ldots m_N >$ corresponds to complex Pochhammer symbol $(\alpha, m_1 i_1 + \ldots + m_N i_N) \equiv (\alpha, \mathbf{m}\cdot\mathbf{i})$ in the coefficient of multiple series.

b. Glueing complex parameter $\langle \alpha \mid m_1, \ldots m_N \rangle$ contains **several** non-zero **integer spectral components** m_n.

c. Individual "complex" parameter $\langle \alpha | 0, \cdots, 0, m_n, 0, \cdots, 0 \rangle$ contains only one non-zero spectral number.

4. Multiple series: more conventions

d. The colon (:) will serve us as delimiter between the glueing and individual parameters. The latters will be put to the right of the colon sequentially, according to the positions of their non-zero components.

e. The simple parameters of the form $\langle a | 0, \cdots, 0, 1, 0, \cdots, 0 \rangle$ and $< \alpha \mid 1, \ldots, 1 >$ are included in the glueing and the individual lists as α and a where spectral components are omitted.

f. For brevity: $\bar{m}_i = -m_i$, $m_{1\bar{2}} = m_1 - m_2$, etc.

g. Empty set of parameters $*$ corresponds to formal Pochhammer symbol $(*, i) \equiv 1$.

5. Multiple series: general notation

$$ {}^N F[L;\mathbf{x}] \equiv {}^N F\left[\begin{array}{lcl} \langle\alpha^1|\mathbf{m}^1\rangle, & \ldots & ,\langle\alpha^A|\mathbf{m}^A\rangle;\ \mathbf{x} \\ \langle\beta^1|\,\mathbf{l}^1\rangle, & \ldots & ,\langle\beta^B|\,\mathbf{l}^B\rangle \end{array}\right] $$

$$ = \sum_{\mathbf{i}=\mathbf{0}}^{\infty} L(\mathbf{i})\frac{\mathbf{x}^{\mathbf{i}}}{\mathbf{i}\,!} = \sum_{i_1,\ldots,i_N} L(i_1,\ldots,i_N)\frac{x_1^{i_1}\cdots x_N^{i_N}}{i_1!\cdots i_N!}, $$

$$ L(\mathbf{i}) = \frac{(\alpha^1,\,\mathbf{m}^1\cdot\mathbf{i})\ldots(\alpha^A,\,\mathbf{m}^A\cdot\mathbf{i})}{(\beta^1,\,\mathbf{l}^1\cdot\mathbf{i})\ldots(\beta^B,\,\mathbf{l}^B\cdot\mathbf{i})}\,. $$

Any parameter can be transferred from numerator to denominator and *vice versa* with the help of identity $(\alpha, i) = (-1)^i(1-\alpha, i)^{-1}$.

In special cases it is of use to differ between glueing and individual parameters and numerator and denominator parameters.

6. Example of ${}^3F$ gives us an instructive way to understand what is implied by the above notation:

$$ {}^3F\left[\begin{array}{c} \langle a|1,\bar{2},0\rangle\langle b|2,1,\bar{1}\rangle, c : g\,;\langle h|2\rangle\,;*;\ x_1,x_2,x_3 \\ d,\langle e|0,1,1\rangle \qquad :*;\quad k \quad ;l \end{array}\right] $$

$$ = \sum_{i_1,i_2,i_3} \frac{(a, i_1-2i_2)(b, 2i_1+i_2-i_3)(c, i_1+i_2+i_3)}{(d, i_1+i_2+i_3)(e, i_2+i_3)} $$

$$ \times\ \frac{(g,i_1)(h,2i_2)}{(k,i_2)(l,i_3)}\,\frac{x_1^{i_1}}{i_1!}\,\frac{x_2^{i_2}}{i_2!}\,\frac{x_3^{i_3}}{i_3!}\,. $$

4. Operator factorization method

4.1. Ω-multiplication is a fundamental operation underlying the factorization method

Ω-product $u*v$ of functions $u(x_1,\ldots,x_N)$ and $v(x_1,\ldots,x_N)$ is defined as

$$ \langle u*v|x_1,\,\ldots\,,x_N\rangle = u\left(\frac{d}{ds_1},\ldots,\frac{d}{ds_N}\right) v(x_1s_1,\,\ldots\,,x_Ns_N)\,|_{\forall s_n=0}\,. $$

For the Ω-product some important general properties are fulfilled: for example, commutation property $u{*}v = v{*}u$; association property $w{*}(u{*}v) = (w{*}u){*}v$; coupling rule, etc. See subsection COMMANDS(II) in [**?**], Sec.2 .

4.2. Operator factorization principle: an illustrative example of general power series

Introduce the notation for arbitrary power series

$$F[A;\, x] = \sum_{i=0}^{\infty} A(i) \frac{x^i}{i\,!}\,, \quad F[A, B;\, x] = \sum_{i=0}^{\infty} A(i) B(i) \frac{x^i}{i\,!}\,,$$

etc. Then factorization formula holds:

$$F[A, B;\, x] = F\left[A; \frac{d}{ds}\right] \, F[B;\, xs]|_{s=0}\,.$$

The condition $s = 0$ should be introduced after fulfillment of term by term differentiation in the right hand side of the factorization formula.

4.3. Ω-representability of multiplication of power series coefficients

The above formula can be written as

$$F[A \times B;\, x] = \langle F[A] \, * \, F[B] \,|\, x\rangle.$$

This notation makes it obvious the **property of Ω-representability** of multiplication operation over coefficients of an *arbitrary power series.*

Suppose we multiply two series with coefficients $A(i)$ and $B(i)$, respectively. It is well known that the operation which should be applied to A and B to produce coefficients of the resultant series is the convolution operation. Now we multiply coefficients of the series. The operation which should be applied to the two initial series to produce the resultant series with coefficients $A(i) \times B(i)$ is just the Ω-multiplication operation.

4.4. Illustrative example of total factorization

Apart from the above simple exercise in calculus, what practical use can be reached with the help of the Ω-multiplication. Can we give a clear-cut illustrative example? Here it is

$$F\begin{bmatrix} a^1, \ldots, a^A\,;\, x \\ b^1, \ldots, b^B \end{bmatrix}$$
$$= F_0^1 \begin{bmatrix} a^1 \\ * \end{bmatrix} * F_0^1 \begin{bmatrix} a^2 \\ * \end{bmatrix} * \cdots * F_0^1 \begin{bmatrix} a^A \\ * \end{bmatrix} * F_1^0 \begin{bmatrix} * \\ b^1 \end{bmatrix} * F_1^0 \begin{bmatrix} * \\ b^2 \end{bmatrix} * \cdots * F_1^0 \begin{bmatrix} * \\ b^B \end{bmatrix}.$$

We see that any property of generalized hypergeometric series is a corollary of the properties of the binomial series F_0^1 and the Bessel-type series F_1^0. The technique necessary for derivations of this kind will be given later.

4.5. Factorization formulas

Factorization formulas for generalized hypergeometric series in one variable have the form:

$$F[\mathbf{d}_1\,;\;x_1\frac{d}{ds}]\;F[\mathbf{d}_2;x_2 s]\Big|_{s=0}=F[\mathbf{d}_1,\mathbf{d}_2\,;\;x_1\,x_2]\;,$$

$$F[\mathbf{d}_1\,;\;x_1\frac{d}{ds}]\;F[\mathbf{d}_2;x_2 s^m]\Big|_{s=0}=F[<\mathbf{d}_1\mid m>,\;\mathbf{d}_2\,;\;x_1^m\,x_2]\;,$$

where $\mathbf{d}_1$ and $\mathbf{d}_2$ are double sets of numerator and denominator parameters.

The above formulas allow complicated series to be expressed as Ω-products of simpler series; *vice versa*, an Ω-product can be expressed in an algebraic form of a more complicated series. Moreover the formulas permit us to introduce very useful general concepts (see subsection 4.6).

General factorization formula for multiple (not necessarily hypergeometric) series is

$$\begin{aligned}&{}^N F[L_1,L_2\,;\;x_1,\,\ldots\,,x_N]=\\&\qquad={}^N\,F\left[L_1\,;\;\frac{d}{ds_1},\,\ldots\,,\frac{d}{ds_N}\right]\;{}^N F[L_2\,;\;x_1s_1,\,\ldots\,,x_Ns_N]\,\big|_{\forall s_n=0}\;,\end{aligned}$$

where $L_k(i_1,\,\ldots\,,i_N)$, $k=1,2$ are **arbitrary** coefficients. It is a direct paraphrase of the factorization formula for $F[\mathbf{d}_1,\mathbf{d}_2;\,x]$.

In hypergeometric case the coefficients $(\mathbf{d},m_1i_1+\,\cdots\,+m_Ni_N)$ or $(\mathbf{d},i_1+\cdots+i_N)$ have a **special** dependence om summation variables. This permits us to use the **special factorization formulas**:

$$\begin{aligned}&{}^N F[<\mathbf{d}\mid m_1,\,\ldots\,,m_N>,\;L\,;\;x_1,\,\ldots\,,x_N]=\\&\qquad\qquad=F\left[\mathbf{d}\,;\;\frac{d}{ds}\right]\;{}^N F[L\,;\;x_1s^{m_1},\,\ldots\,,x_Ns^{m_N}]\Big|_{s=0}\;,\end{aligned}$$

$${}^N F[\mathbf{d},L;x_1,\,\ldots\,,x_N]={}^1F\left[\mathbf{d};\frac{d}{ds}\right]\;{}^N F[L;x_1s,\,\ldots\,,x_Ns]\Big|_{s=0}\;,$$

where, contrary to the general factorization formula, the first multipliers in the Ω-products are **simple** hypergeometric series.

4.6. General concepts, which constitute the structural basis of the method

1. Ω-identical expressions

By analogy with **arithmetically identical expressions** the algebraic expressions connected by finite number of arithmetic operations **and Ω-multiplication operations** will be called **Ω-identical expressions**.

2. Ω-equivalent operators

The operators F_1 and F_2 are called Ω-equivalent operators ($F_1 \rightleftharpoons F_2$) if the identity

$$F_1\left(\frac{d}{ds}, s\right)\Psi(xs)\Big|_{s=0} = F_2\left(\frac{d}{ds}, s\right)\Psi(xs)\Big|_{s=0}$$

holds for an arbitrary function Ψ. Note that the Ω-equivalent operators are not necessarily identical to one another.

3. Ω-equivalent relations

The functional relation $f * f_1 = f * f_2$ or

$$f\left(d/ds_1, \ldots, d/ds_N\right) f_1(x_1s_1, \ldots, x_Ns_N)\Big|_{\forall s_n=0} = f\left(d/ds_1, \ldots, d/ds_N\right) f_2(x_1s_1, \ldots, x_Ns_N)\Big|_{\forall s_n=0}$$

where f is an arbitrary function of N variables will be called Ω-equivalent to the relation

$$f_1(x_1, \ldots, x_N) = f_2(x_1, \ldots, x_N).$$

4. Ω-equivalent classes and proto-relations

In a class of Ω-equivalent relations a **simplest relation** can be chosen to serve as a **proto-relation** underlying the class and giving rise to all its members. Having proved the proto-relation we **prove all formulas** belonging to the class.

4.7. How the method works

(I) A standard four step scheme

1. Analysis. An initial series is decomposed into an Ω-product of simpler series.

2. Basic transformations. The known properties of the simpler series are used to transform the factorized expression to the desired form.

3. Auxiliary transformations. A finite number of auxiliary transformations is employed to convert the resultant expression to the form permitting the use of a factorization formula.

4. Synthesis. The appropriate factorization formula is applied to turn the operator expression into an algebraic form.

(II) Scheme based on the concept of Ω-equivalent relations

The usefulness of the relation $f * f_1 = f * f_2$ depends on f_1, f_2 and f. To have a very simple illustrative example we use the relation $F[a;x] = (1-x)^{-a}$ to convert $(1-x)^{-a} = (1-x)(1-x)^{-a-1}$ into

$$F_0^1[a;x] = F_0^1[a+1;x] - x\,F_0^1[a+1;x]\,.$$

Applying the operator $F_2^0[* \,//\, a, a+1; zd/dx]\big|_{x=0}$ to both sides of the binomial identity and using relations from the COMMANDS list (Ref.[?], Sec.2) we obtain the recurrence relation for the Bessel-type series

$$F_1^0\begin{bmatrix} * & ; z \\ b & \end{bmatrix} = F_1^0\begin{bmatrix} * & ; z \\ b-1 & \end{bmatrix} - \frac{z}{b(b-1)}\,F_1^0\begin{bmatrix} * & ; z \\ b+1 & \end{bmatrix}.$$

4.8. Different computational modes

1. Factorization method gives us a **limited set** of operations sufficient to obtain manually, with pen and paper, any property of an arbitrary series (**manual mode**).

2. If the operations were implemented in the form of computer commands we could obtain a superstructure over an existent computer algebra system capable to derive any formula in an interactive mode (**globally universal computer mode**). In the case in point we set problem to formalize the main operators of the method to make them consistent with the grammatical structure of the SANTRA language. The preliminary command set and its prototypical operations are presented in subsections COMMANDS (I) – COMMANDS(VI) of Sec.2, Ref.[?].

A representative example illustrating the main features of the both modes "emulating" one another will be given in Sec.5 where operations presented in Sec.2, Ref.[?] are only used.

3. So far we used another computational mode. Considering separate classes of formulas we used the factorization method to obtain a limited number of formulas playing the role of basis relations. Combining these relations (see MACRO – COMMANDS in Ref.[**?**], Sec.2) one can obtain all relations belonging to the class. Interactive and automated computer programs have been developed for 5 classes of formulas inaccessible in any other way (**bounded universal computer mode**). Some examples are given in Sec. 6.

5. Formula derivation based on the use of the operator factorization method

5.1. Passing to examples

Given hundreds and thousands of multiple hypergeometric series and innumerable scores of formulas fulfilling for these series it seems surprising that a **moderate set of operations** listed in COMMANDS and MACRO – COMMANDS sections **is sufficient** to derive any property of an arbitrarily complicated series.

Formal proof of derivation potential of the method **can hardly be given** at present. One can make certain of the merits of the method in a practical manner by deriving sufficiently large number of formulas. We further give derivation of **a formula illustrating four step analytical scheme** (see 4.7 (I)).

5.2. Transformation of F_4

1. Factorization allows the function F_4 to be expressed in terms of simpler functions U

Introduce the F_4 function

$$F_4 \equiv F\left[\begin{array}{lllll} a_1, a_2 & : * ; & * & ; \dfrac{z_1}{(1-v)(1-u)}, \dfrac{z_2}{(1-v)(1-u)} \\ \quad * & : b_1 ; & b_2 & \end{array}\right].$$

The most interesting factorization of the F_4 gives

$$F_4 = U\left[a_1, b_1;\ \frac{d(s_2)}{1-v},\ \frac{d(s_1)}{1-v}\right] U\left[a_2, b_2; \frac{z_1 s_1}{1-u}, \frac{z_2 s_2}{1-u}\right]\Bigg|_{s_1=s_2=0},$$

$$U[a, b; x_1, x_2] = F\left[\begin{array}{l} a: \;\; *; \;\; *; \;\; x_1, \;\; x_2 \\ *: \;\; *; \;\; b \end{array}\right].$$

2. Function U, in contrast to F_4, exhibits obvious simple properties

The function U

$$U[a, b; x_1, x_2] = F\left[\begin{array}{l} a: \;\; *; \;\; *; \;\; x_1, \;\; x_2 \\ *: \;\; *; \;\; b \end{array}\right]$$

has **binomial type** (1//0) with respect to x_1 and **Kummer type** (1//1) with respect to x_2. Specialization of **general** binomial and Kummer transformations gives

$$U[a, b;\, x_1 + u, x_2] = (1-u)^{-a}\, U_1\left[a, b; \frac{x_1}{1-u}, \frac{x_2}{1-u}\right],$$

$$U[a, b; x_1, x_2] = e^{x_2}\, F\left[\begin{array}{ccc} \langle b-a|1, \overline{1}\rangle & : *; & a,\, 1+a-b; \;\; -x_2, \;\; -x_1 \\ * & : b; & * \end{array}\right].$$

3. Using binomial properties of U-functions

Binomial transformation of both Ω-multipliers gives

$$F_4 = (1-v)^{a_1}(1-u)^{a_2} \\ \times U[a_1,\, b_1; d(s_2) + v, d(s_1)]\;\, U[a_2,\, b_2; z_1 s_1 + u, z_2 s_2]|\big|_{s_1 = s_2 = 0}.$$

Applying the **operator displacement formula**

$$F[d(s) + v] = \exp(-vs) F[d(s)] \exp(vs)$$

to the first multiplier and the **shift operator formula**

$$F[s + u] = \exp(u\, d/ds) F[s]$$

to the second multiplier ensures **elimination of interfering constant terms** u and v from the arguments.

4. Binomial and Kummer transformations give us a complicated Ω-product

After "uniformization of arguments" in U functions the result of binomial transformations in $F_4 = U * U$ is

$$F_4 = (1-v)^{a_1}(1-u)^{a_2} U[a_1, b_1;\, d(s_2),\, d(s_1)]$$
$$\times \exp[u z_1^{-1}\, d(s_1)] \exp(s_2 v)\, U\, [a_2,\, b_2;\, z_1 s_1,\, z_2 s_2]\big|_{s_1=s_2=0}\,.$$

Kummer transformation of the both U's gives us

$$F_4 = (1-v)^{a_1}(1-u)^{a_2}$$
$$\times F\left[\begin{matrix} \langle b_1 - a_1 | 1, \bar{1}\rangle & : & * & ; & a_1,\, 1+a_1-b_1; & -d(s_1), -d(s_2) \\ * & : & b_1 & ; & * & \end{matrix}\right]$$
$$\times\ \exp\left[((z_1+u)/(z_1))\, d(s_1)\right]\ \exp\left[(z_2+v)s_2\right]$$
$$\times F\left[\begin{matrix} \langle b_2 - a_2 | 1, \bar{1}\rangle & : & * & ; & a_2,\, 1+a_2-b_2; & -z_2 s_2, z_1 s_1) \\ * & : & b_2 & ; & * & \end{matrix}\right]\Bigg|_{s_1=s_2=0}.$$

5. Applying again a factorization formula we obtain the desired result

To simplify the Ω-product representation for F_4 we eliminate the exponential terms by putting $u = -z_1$, $v = -z_2$. Using factorization formula we finally obtain

$$F\left[\begin{matrix} a_1, a_2 : * \,;\, * \,; & \dfrac{z_1}{(1+z_1)(1+z_2)}, \dfrac{z_2}{(1+z_1)(1+z_2)} \\ * \quad : b_1 \,; b_2 & \end{matrix}\right] =$$
$$= (1+z_2)^{a_1}(1+z_1)^{a_2} F\left[\begin{matrix} \langle b_1 - a_1 | 1, \bar{1}\rangle, \langle b_2 - a_2 | \bar{1}, 1\rangle : \\ * \qquad\qquad : \end{matrix}\right.$$
$$\left.\begin{matrix} : a_2, 1+a_2-b_2 \,;\, a_1, 1+a_1-b_1 \,;\, z_1, z_2 \\ : \qquad b_1 \qquad ; \qquad b_2 \end{matrix}\right].$$

This formula transforms the complete series of the second order (of the type $[2//1, 2//1]$) into the complete series of the third order (of the type $[3//2, 3//2]$).

5.3. Analytical corollaries

Factorizing the r.h.s. of the above formula we can express the general F_4 as an Ω-product of a Kummer function F_1^1 and two Gauss functions F_1^2.

This implies that **many new formulas** for F_4 can be deduced from the known properties of F_1^1 and F_1^2.

Many known hard-hitting results follow immediately from the above formula. If $b_1 + b_2 = a_1 + a_2 + 1$ then $F_4 \sim F_1^2 F_1^2$ (**Watson formula**). If $b_2 = a_2$ then using the "indefinite" transformations and an appropriate linear transformation we obtain $F_4 \sim F_1$ (**Bailey formula**). If $a_2 = b_1 + b_2 - 1$ then using a couple of linear transformations we can see that $F_4 \sim F_2$ (**another Bailey formula**).

6. Computer generation of formulas using bounded universal programs

6.1. Programs

A complex of programs have been developed with the help of macro – commands realizing Kummer-type and Gauss-type **linear** transformations, **quadratic** transformations, analytic **continuation** formulas and **reduction** formulas turning multiple series into series depending on lesser number of variables. Provision was made for both **interactive** and **automated** modes of processing.

6.2. Example of computer generation: the case of the Appell function F_4

In the following we confine ourselves to the only example of special Appell function $F_4[a_1, a_2, a_1, b_2; x_1, x_2]$ having one constraint on parameters.

Using the special transformation of general F_4 and utilizing, for the case of restricted $F_4[a_1, a_2, a_1, b_2; x_1, x_2]$, either contraction reduction or the auxiliary "indefinite" transformation we represent the F_4 as non-Hornian functions:

$$K_{gb} = F \begin{bmatrix} \alpha : a_1, a_1' \,; a_2 \,; \; x_1, x_2 \\ \beta : \quad b_1 \quad ; * \end{bmatrix} ,$$

$$\Gamma_{bg} = F \begin{bmatrix} \langle \alpha_1 | 1, \bar{1} \rangle \langle \alpha_2 | \bar{1}, 1 \rangle : a_1 \,; a_2, a_2' \,; \; x_1, x_2 \\ : * \,; \quad b_2 \end{bmatrix} .$$

The processing of these functions consisted in using all possible linear commands **lin(G)** along with an auxiliary bilinear transformation (for the functions with parameter $\langle 0|1, \bar{1}\rangle$). The process performed in automatic mode gave us functions symbolically presented by double black circles on Fig. 1.

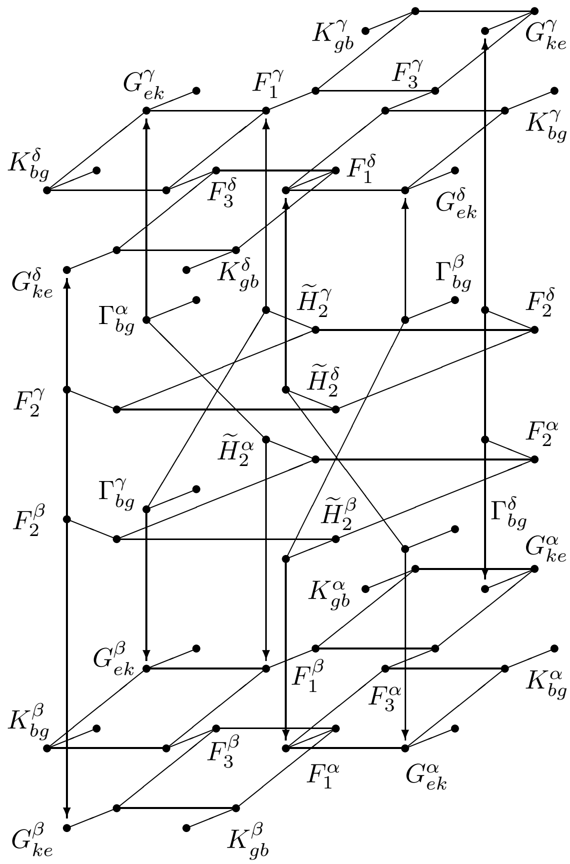

Fig. 1. The diagram representing 36 series generated by the function $F_4[a_1, a_2, a_1, b_2]$. The arrows correspond to additional auxiliary transformations connected with "indefinite" parameters (see Eqs. (37) and (38) in [**?**]).

Long lines symbolize general linear transformations (see Sec. 2.7.3 in [**?**]). Each series is represented by short inclined segments with two black nodes corresponding to the arguments of the series. The arrows indicate additional transformations connected with "indefinite" parameters.

Along with functions K_{gb} and Γ_{gb} we obtained, in an automatic mode, the following 5 functions:

$$G_{ek} = F \begin{bmatrix} \alpha_1, \alpha_2 : * \,; a_2 \,;\; x_1, x_2 \\ \beta \quad : * \,; b_2 \end{bmatrix}$$

$$F_1 = F \begin{bmatrix} \alpha : a_1 \,; a_2 \,;\; x_1, x_2 \\ \beta : \, * \;;\; * \end{bmatrix},$$

$$F_3 = F\begin{bmatrix} * : a_1, a_1' \,; a_2, a_2' \,;\; x_1, x_2 \\ \beta : \quad * \quad ; \quad * \end{bmatrix},$$

$$F_2 = F\begin{bmatrix} \alpha : a_1 \,; a_2 \,;\; x_1, x_2 \\ * : b_1 \,; b_2 \end{bmatrix},$$

$$\widetilde{H}_2 = F\begin{bmatrix} \langle\alpha|\bar{1}, 1\rangle : a_1, a_1' \,; a_2 \,;\; x_1, x_2 \\ : \quad * \quad ; b_2 \end{bmatrix}.$$

Transformation properties of the series G_{ke} and G_{ek} make it possible to obtain a toroidal construction made of three etageres jointed by their upper and lower facets with one another. Of course it would be difficult to establish such connection between 108 double series without using computer–aided symbolic approach based upon the new computational principle. Note that the use of new transformations permits our program to make start from an arbitrary point of the diagram.

7. Prospects of the globally-universal approach within the system of analytical transformations Santra 3 based on Refal language

We discuss in brief the problem of implementation of a globally universal approach (see Sec. 4.8.2) using the domestic algorithmic language Refal. Envision the 4-layer structure

Hypertrans graphic shell	(4)
Hypertrans functions	(3)
Santra 3	(2)
Refal 6	(1)

which is graphical representation of a conceived system of analytical transformations of hypergeometric series. The system will be called, for brevity, the Hypertrans system. A computer satisfying common general conditions plays the role of the "bottom" zero level. All peculiar features of system are inherent in the level 1 which plays the role of the "specialized basement" for all other upper levels. This leads to the machine – independent system. Any change in the level 1, for example, transfer to other Refal version may cause serious re-building of other levels. Having presently only the level 1 facilities with Refal 6 (http://www.refal.net/˜arklimov/refal6/) functioning in DOS window with tight graphic interface we aim, for the time being, at development of a model Hypertrans system having complete set of program facilities with but a limited set of graphic conveniences.

Level 2 is exhausted at present by tools of the system Santra 2 [?]-[?]. These tools are not convenient for implementation of Hypertrans functions. However Santra 2 provides a reliable basis for development of more convenient system Santra 3 because Santra 2 was conceived as a universal system of algebraic transformations using symbolic manipulations similar to those of Refal. The experience gained in course of implementing different algebraic transformations with the help of Refal tools [?]-[?] served us as a basis for development of Santra 2. The universality feature inhereted by Sanra 2 from Refal allows us to build up various problem-oriented superstructures. The aim of the improvements is development of more convenient tools without change of already written code. Really, the universality of Santra 2 implies that any algebraic transformations can be implemented, in principle, with the help of basic system tools. Practically, we need more convenient tools for a specific applied problem. It is important that any "applied subsystem" can be readily modified without any change in the basic system.

Specifically, it is Refal 2 [?]-[?] that plays the role of paradigm for Santra 2 language. One of the additional capabilities is that the left parts of statements can be formed algorithmically. This allows algorithmic recognition of objects having fixed structures, for example, the typed expressions. In turn, this allows to differentiate operations depending on the type of object on the same principle as in the case of abstract data types. Besides, the Santra 2 language admits dynamical program formation that allows the use of macrocommands and subroutine relocation depending on a specific data set. However the processing of the normal forms which are used in Santra 2 for representation of algebraic expressions is not sufficient for operations required by operator factorization method. For example, it is necessary to process functions whose arguments, exponents, coefficients and free terms play the role of parameters in contextual definition. In Santra 2, there are no convenient tools for extraction such functions from normal forms. One more deficiency is that Santra 2 is closely related to Refal 2 which is characterized by less convenient form of program record compare to Refal 5 and Refal 6 (http://www.refal.net/˜arklimov/refal6/) which gives an extension of Refal 5. Santra 3 is being developed on the basis of Refal 5 supplemented by ideas of Refal 6 and some additional options.

All extensions introduced in Santra 3 persue the goal to simplify programming by development of more simple and illustrative tools close to the typical manipulations of researcher engaged in derivation of a mathematical formula. The work on the project of Santra 3 is still in progress now.

The approach to construction of the level 3 functions will be illustrated by analysis of several typical transformations of the factorization method [?]. The level 3 as a whole is constituted by the set of commands corresponding to the complete set of operations used in the factorization method. It is just what we call the command set of the globally universal approach. The program implementation of the commands is illustrated by model constructions of Santra 3 [?].

Summary

He who has to deal with hypergeometric series systematically and even he who confronts with them occasionally should familiarize himself with the operator factorization method.

The reasons giving advantage to the method over innumerable traditional approaches to the theory of hypergeometric series and special functions are as follows:

- Factorization principle is a **key – stone** of the new theory
- A set of auxiliary identities together with the factorization formulas play the role of a **meta – language** giving us the shortest way to write the derivation story for any formula
- Hypergeometric series are expressed only through hypergeometric series (**closure property**). On the one hand the hypergeometric series are subjects of investigation, on the other hand they are investigatory tools. Such **an ideal correspondence** is the main reason of extraordinary efficiency of the method
- New theoretical concepts (Ω-equivalent relations, Ω-equivalent identities, Ω-identical transformations, etc.) have a significant heuristic value and give a well-structured form to the theory
- Bounded universal approach based on the use of **macro – commands** allowed extensive classes of formulas to be obtained with the help of computer-aided symbolic transformations
- Instead of **separate tools** inherent in other computer algebra approaches the factorization method gives us an **integrated system of commands**. Prospective development may give us a computer algebra superstructure allowing us to work with any types of hypergeometric series without need to tear ourselves from keyboard

References

[1] Niukkanen A.W. New theory of hypergeometric series and new prospects for computer algebra systems. Fundamental and applied mathematics (in Russian) v. 5, no 3, pp. 716-745, 1999.

[2] Niukkanen A.W. A transformation of F_4 suggestive of a new symbolic software. Computer Physics Communications v. 126, pp. 137–140, 2000.

[3] Niukkanen A.W., Paramonova O.S. Computer generation of complicated transformations and reduction formulas for multiple hypergeometric series. Computer Physics Communications v. 126, pp. 141 – 148, 2000.

[4] Niukkanen A.W. Extention of factorization principle on general hypergeometric series. Mat. Zametki (in Russian) v. 67, no 4, pp. 573-581, 2000.

[5] Niukkanen A.W. General linear transformations of hypergeometric functions. Mat.Zametki (in Russian) v. 70, no 5, pp. 769-779, 2001.

[6] Niukkanen A.W. Analytical continuation formulas for multiple hypergeometric series. Fundamental and applied mathematics (in Russian) v. 7, no 1, pp. 1-16, 2001.

[7] Niukkanen A.W. Quadratic transformations of multiple hypergeometric series. Fundamental and applied mathematics (in Russian) v. 8, no 2, pp. 517-531, 2002.

[8] Paramonova O.S., Niukkanen A.W. Computer–aided analysis of transformation formulas of the Appell and the Horn functions. Programmirovanie (in Russian) no 2, pp. 1-6, 2002.

[9] Niukkanen A.W. On the way to computerizable scientific knowledge (by the example of the operator factorization method). Nucl. Instr. and Methods A v. 502, pp. 639-642, 2003.

[10] I.B. Shchenkov. System of symbolic analytical transformations SANTRA-2. Description of formal part of command language. Keldysh

Inst. of Appl. Math. Academy of Science USSR, Preprint No 1, 1989 (32p.).

[11] I.B. Shchenkov. System of symbolic analytical transformations SANTRA-2. Description of dynamical functions. Keldysh Inst. of Appl. Math. Academy of Science USSR, Preprint No 7, 1989 (24p.).

[12] I.B. Shchenkov. System of symbolic analytical transformations SANTRA-2. Description of functions supporting non-algebraic operations. Keldysh Inst. of Appl. Math. Academy of Science USSR, Preprint No 21, 1989 (31p.).

[13] I.B. Shchenkov. System of symbolic analytical transformations SANTRA-2. Operations over expression belonging to the main classes. Keldysh Inst. of Appl. Math. Academy of Science USSR, Preprint No 14, 1991 (38p.).

[14] I.B. Shchenkov. System of symbolic analytical transformations SANTRA-2. Operations over matrices. Keldysh Inst. of Appl. Math. Academy of Science USSR, Preprint No 15, 1991 (29p.).

[15] I.B. Shchenkov. System of symbolic analytical transformations SANTRA-2. Program design and program checkout tools. Keldysh Inst. of Appl. Math. Academy of Science USSR, Preprint No 1, 1993 (28p.).

[16] A.P.Budnik, E.V.Gai, N.S. Rabotnov, V.F. Turchin, S.V. Popov, I.B. Shchenkov. Machine performance of analytical transformations in mathematical physics with the help of REFAL language. Theses. Symposium on processing of symbolic information. Computer Center GSSR, Tbilisi, 1970.

[17] I.B. Shchenkov. Program in mathematical physics with the help of REFAL language. Theses. Symposium on processing of symbolic information. Computer Center GSSR, Tbilisi, 1970.

[18] A.P.Budnik, E.V.Gai, N.S. Rabotnov, N.S.Klimov, V.F. Turchin, I.B. Shchenkov. Basis wave functions and operator matrices in the nuclei model. Yadernaya fizika (in Russian), V. 14, No 2, pp.304-314, 1971.

[19] V.N. Vinogradov, F.V. Gai, S.V. Popov, N.C. Rabotnov, I.B. Shchenkov. Construction of physical bases of O(5) and SV(3) groups with automated execution of symbolic transformations. Yadernaya fizika (in Russian), V. 16, No 6, pp.1178-1187, 1972.

[20] And. V. Klimov, S.A. Romanenko. Programming system Refal-2 for Unified Series of Electronoc Compating Machines. Description of command language. IAM, Academy of Science USSR, Moscow: 1987.

[21] And. V. Klimov, S.A. Romanenko. Programming system Refal-2 for Unified Series of Electronoc Compating Machines. Description of functions library. IAM, Academy of Science USSR, Preprint No 200, 1996.

[22] S.A. Romanenko. Implementation of Refal-2. IAM, Academy of Science USSR, Moscow: 1987.

[23] A.W. Niukkanen, I.B. Shchenkov, G.B. Efimov. A project of a globally universal interactive program of formula derivation based on the operator factorization technique. Keldysh Inst. of Appl. Math. RAS, Preprint No 82, 2003, (24 p.).

[24] D.W. Lozier. The DLMF Project: A new initiative in classical special functions, in C. Dunkl, M. Ismail and R. Wong, eds., Special Functions: Proceedings of the International Workshop, World Scientific (Singapore), pp. 207-220, 2000.