S U S Y 2

by
Ziemowit Popowicz by

Institute of Theoretical Physics, University of Wrocław,
pl.M.Borna 9 50-205 Wrocław, Poland
e-mail ziemek@ift.uni.wroc.pl

version 1.2

March 20, 2004

Abstract

This package deal, with supersymmetric functions and with algebra of supersymmetric operators in the extended N=2 as well as in the nonextended N=1 supersymmery. It allows us to make realization of SuSy algebra of differential operators, compute the gradients of given SuSy Hamiltonians and to obtain SuSy version of soliton equations using SuSy Lax approach. There are also many additional procedures also encountered in SuSy soliton approach, as for example: conjugation of a given SuSy operator, computation of general form of SuSy Hamiltonians (up to SuSy-divergence equivalence), checking of the validity of the Jacobi identity for some SuSy Hamiltonian operators.

1 Introduction

The main idea of the supersymmetry (SuSy) is to treat boson and fermion operators equally [1,2]. This has been realised by introducing the so called supermultiplets constructed from the boson and fermion operators and additionally from the Mayorana spinors. Such supermultiplets posses the proper transfomations property under the transformation of the Lorentz group. At the moment we have no experimental confirmations that the supersymmetry appeare in the nature.

The idea of using supersymmetry (SuSy) for the generalization of the soliton equations [3-7] appeared almost in parallel to the usage of SuSy in the quantum field theory. The first results, concerning the construction of classical field theories with fermionic and bosonic fields depending on time and one space variable, can be found in [8-12]. In many cases, the addition of fermions fields does not guarantee that the final theory becomes SuSy invariant and therefore this method was named as the fermionic extension in order to distinguish it from the fully SuSy method.

In order to get a SuSy theory we have to add to a system of k bosonic equations kN fermion and k(N-1) boson fields (k=1,2,... N=1,2,..) in such a way that the final theory becomes SuSy invariant. From the soliton point of view we can distinguish two important classes of the supersymmetric equations: the non-extended ($N = 1$) and extended ($N > 1$) cases. Consideration of the extended case may imply new bosonic equations whose properties need further investigation. This may be viewed as a bonus, but this extended case is no more fundamental than the non-extended one. The problem of the supersymmetrization of the nonlinear partial differential equations has its own history, and at the moment we have no unique solution [13-40]. We can distinguish three different methods of supersymmetrization, as for example the algebraic, geometric and direct method.

In the first two cases we are looking for the symmetry group of the given equation and then we replace this group by the corresponding SuSy group. As a final product we are able to obtain SuSy generalization of the given equation. The classification into the algebraical or geometrical approach is connected with the kind of symmetry which appears in the classical case. For example, if our classical equation could be described in terms of the geometrical object then the simple exchange of the classical symmetry group of this object with its SuSy partner justifies the name geometric. In the case of algebraic we are looking for the symmetry group of the equation without any reference to its geometrical origin. This strategy could be applied to the so called hidden symmetry as for example in the case of the Toda lattice . These methods each have advantages and disadvantages. For example, sometimes we obtain the fermionic extensions. In the case of the extended supersymmetric Korteweg-de-Vries equation we have three different fully SuSy extensions; however only one of them fits to these two classifications.

In the direct approach we simply replace all objects which are appear in the evolution equation by all possible combinations of the supermultiplets and its superderivative in such a way that to conserve the conformal dimensions. This is non unique and we yields many different possibilities. However the arbitrariness is reduced if we additionally investigate super-bi-hamiltonian structure or try to find its supersymmetric Lax pair. In many cases this approach is successful.

The utilization of the above methods can be helped by symbolic computer algebraic and for this reason we prapared the package SuSy2 in the symbolic language REDUCE [41].

We have implemented and ordered the superfunctions in our program, extensively using the concept of " noncom operator " in order to implement the supersymmetric integro - differential operators. The program is meant to perform the symbolic calculations using either fully supersymmetric supermultiplets or the components version of our supersymmetry. We have constructed 25 different commands to allow us to compute almost all objects encountered in the supersymmetrization procedure of the soliton equation.

2 Supersymmetry

The basic object in the supersymmetric analysis is the superfield and the supersymmetric derivative. The superfields are the superfermions or the superbosons [1]. These fields, in the case of extended N=2 sypersymmetry, depends, in addition to x and t, upon two anticommuting variables, θ_1 and θ_2 $(\theta_2\theta_1 = -\theta_1\theta_2, \theta_1^2 = \theta_2^2 = 0)$. Their Taylor expansion with respect to the $\theta' s$ is

$$b(x,t,\theta_1,\theta_2) := w + \theta_1\zeta_1 + \theta_2\zeta_2 + \theta_2\theta_1 u, \tag{1}$$

in the case of superbosons, while for the superfermions reads

$$f(x,t,\theta_1,\theta_2) := \zeta_1 + \theta_1 w + \theta_2 u + \theta_2\theta_1\zeta_2, \tag{2}$$

where w and u are classical (commuting) functions depending on x and t , ζ_1 and ζ_2 are odd Grassmann valued functions depending on x and t.

In the set of these superfunctions we can defined the usual derivative and the superderivative. Usually, we encounter two different realizations of the superderivative : the first we call " traditional " and the second " chiral ".

The traditional realization can be defined by introducing two superderivatives D_1 and D_2

$$\begin{aligned} D_1 &= \partial_{\theta_1} + \theta_1\partial, & (3) \\ D_2 &= \partial_{\theta_2} + \theta_2\partial, & (4) \end{aligned}$$

with the properties:

$$\begin{aligned} D_1 * D_1 = D_2 * D_2 = \partial, && (5) \\ D_1 * D_2 + D_2 * D_1 = 0. && (6) \end{aligned}$$

The chiral denoted is by

$$\begin{aligned} D_1 &= \partial_{\theta_1} - \frac{1}{2}\theta_2\partial, & (7) \\ D_2 &= \partial_{\theta_2} - \frac{1}{2}\theta_1\partial, & (8) \end{aligned}$$

with the properties:

$$\begin{aligned} D_1 * D_1 = D_2 * D_2 = 0, && (9) \\ D_1 * D_2 + D_2 * D_1 = -\partial. && (10) \end{aligned}$$

Below we shall use the name " traditional" or " chiral " or " chiral1 "algebras to denote kind of the commutation realations on the superderivativeis assumed. The " chiral1 " algebras case possess, additioanly to the "chiral " algebra, the commutator of D_1 and D_2 denoted as

$$D_3 = D_1 * D_2 - D_2 * D_1. \tag{11}$$

In SuSy2 package we have will implemented the superfunctions and the algebra of superderivatives. Moreover, we have defined many additional procedures which are useful in the supersymetrizations of the classical nonlinear system of partial differential equation. Different applications of this package to the physical problems could be found in the papers [34-38].

3 Superfunctions

The superfunctions are represented in this package by:

$$\mathbf{bos}(f,0,0), \tag{12}$$

for superbosons, while by

$$\mathbf{fer}(g,0,0), \tag{13}$$

for superfermions.

The first index denotes the name of the given superobject, the second denotes the value of SuSy derivatives, and the last give the value of usual derivative. The *bos* and *fer* objects are declared as the operators and as noncom object in the Reduce language. The first index can take an arbitrary name but with the following restriction:

$$\mathbf{bos}(0,n,m) = 0, \tag{14}$$

$$\mathbf{fer}(0,n,m) = 0. \tag{15}$$

for any values of n,m.

The program has the capability to compute the coordinates of the arbitrary SuSy expression, using the expansions in the powers of θ. We have here four commands:

A) In order to have the given expression in the components use

$$\mathbf{fpart}(expression). \tag{16}$$

The output is in the form of the list, in which first element is the zero order term in θ, second is the first order term in θ_1, third is the first order term in θ_2 and the fourth is in $\theta_2 * \theta_1$. For example, the superfunction (11) has the representation

$$\begin{aligned}\mathbf{fpart}(\mathbf{bos}(f,0,0)) \quad => \quad & \{\mathbf{fun}(f_0,0), \mathbf{gras}(ff_1,0), \\ & \mathbf{gras}(ff_2,0), \mathbf{fun}(f_1,0)\},\end{aligned} \tag{17}$$

where *fun* denotes the classical function while the *gras* the Grassmann function. First index in the *fun* or in *gras* denotes the name of the given object, while the second denotes the usual derivative.

B) In order to have the bosonic sector only, in which all odd Grassmann functions disappear, use

$$\mathbf{bpart}(expression). \tag{18}$$

Example:

$$bpart(fer(g,0,0)) => \{0, fun(g_0,0), fun(g_1,0), 0\}. \tag{19}$$

C) In order to have the given coordinates, use

$$\mathbf{bf\_part}(expression, n), \tag{20}$$

where n=0,1,2,3.

Example:

$$bf\_part(bos(f,0,0),3) => fun(f_1,0). \tag{21}$$

D) In order to have the given coordinates in the bosonic sector, use

$$\mathbf{b\_part}(expression, n), \tag{22}$$

where n=0,1,2,3.

Example

$$b\_part(fer(g,0,0),1) => fun(g_0,0) \tag{23}$$

Notice that in the program, from the default we switch to on the factor $fer, bos, gras, fun$. If you remove this factor, then many commands give you wrong result (for example the command lyst, lyst1 and lyst2).

4 The inverse and exponentials of superfunctions.

In addition to our definitions of the superfunctions we can also define the inverse and the exponential of superboson.

The inverse of the given bos function (not to be confused with the “ inverse function ” encountered in the usual analysis) is defined as

$$\mathbf{bos}(f,n,m,-1), \tag{24}$$

for an arbitrary f,n,m with the property $bos(f,n,m,-1)*bos(f,n,m,1)=1$. The object $bos(f,n,m,k)$, in general denotes the k-th power of the $bos(f,n,m)$ superfunction. If we use the command “**let inverse**$''$ then three indices bos objects are transformed onto four indices objects.

The exponential of the superboson function is

$$\mathbf{axp}(bos(f,0,0)). \tag{25}$$

It is also possible to use $axp(f)$, but then we should specify what is f.

We have the following representation in the components for the inverse and axp superfunctions

$$\begin{aligned}
fpart(bos(f,0,0,-1)) &= \{fun(f_0,0,-1), -fun(f_0,0,-1)*gras(ff_1,0), \\
&\quad -fun(f_0,0,-1)*gras(ff_2,0), -fun(f_0,0,-2)*fun(f_1,0,1) \\
&\quad +2*fun(f_0,0,-3)*gras(ff_1,0)*gras(ff_2,0)\} \qquad (26) \\
fpart(axp(f)) &= \{\mathbf{axx}(bf\_part(f,0)), axx(bf\_part(f,0))*bf\_part(f,1), \\
&\quad axx(bf\_part(f,0))*bf\_part(f,2), axx(bf\_part(f,0)) \\
&\quad *(bf\_part(f,3)+2bf\_part(f,1)*bf\_part(f,2))\} \qquad (27)
\end{aligned}$$

where $axx(f)$ denotes teh exponentiation of the given classical function while $fun(f,m,n)$ the n th power of the function $fun(f,m)$.

5 Ordering.

Three different superfunctions fer, bos, axp are ordered among themselves as

$$fer(f,n,m)*bos(h,j,k)*axp(g), \tag{28}$$

$$fer(f,n,m)*bos(h,j,k,l)*axp(g), \tag{29}$$

indenpendently of the indices. Superfunctions *bos* and *axp* are commuting among themselves, while the superfunctions fer anticommutes among themselves. For these superfunctions we introduce the following ordering:.

A) The *bos* objects with three and four indices are ordered as: the first index antilexicographically, the second and the third index as decreasing order of natural numbers. The last, fourth index is not ordered because:

$$bos(f,n,m,k) * bos(f,n,m,l) => bos(f,n,m,k+l) \tag{30}$$

B) The anticommuting fer objects we ordered as follows: the first index antilexicographically, second and third index as decreasing order of natural numbers.

Example:

$$fer(f,n,m) * fer(g,k,l) => -fer(g,k,l) * fer(f,n,m) \tag{31}$$

for an arbitrary n,m,k,l

$$fer(f,n,m) * fer(f,n,m) => 0 \tag{32}$$

for an arbitrary f,n,m.

$$bos(f,2,3,7) * bos(aa,0,3) * bos(f,2,3,-7) => bos(aa,0,3), \tag{33}$$

$$bos(f,2,3,2) * bos(zz,0,3,2) * bos(f,2,3,-2) => bos(zz,0,3,2). \tag{34}$$

C) For all exponential functions we have

$$axp(f) * axp(g) => axp(f+g). \tag{35}$$

6 (Super)Differential operators.

We have implemented three different realizations of the supersymmetric derivatives. In order to select traditional realization declare **let trad** . In order to select chiral or chiral1 algebra declare **let chiral** or **let chiral1**. By default we have traditional algebra.

We have introduced three different types of SuSy operators which act on the superfunctions and are considered as operators and as noncomuting objects in the Reduce language.

For the usual differentiation we introduced two types of operators:

(i) rigth differentations,

$$\mathbf{d(1)} * bos(f,0,0) => bos(f,0,1) + bos(f,0,0) * d(1); \tag{36}$$

(ii) left differentations,

$$fer(f,0,0) * \mathbf{d(2)} => -fer(f,0,1) + d(2) * fer(f,0,0). \tag{37}$$

From this example follows that the third index in the bos, fer object can take an arbitrary integer value.

Susy derivatives we denote as *der* and *del*. *Der* and *del* represent the right and left operatopns, respectively, and are one component argument operations. The action of these objects on the superfunctions depends on the choice of the supersymmetric algebra.

Explicitely we have for the traditional algebra:

a) Right SuSy derivative

$$\mathbf{der}(\mathbf{1}) * bos(f,0,0) \quad => \quad fer(f,1,0) + bos(f,0,0) * der(1), \tag{38}$$

$$\mathbf{der}(\mathbf{2}) * fer(g,0,0) \quad => \quad bos(g,2,0) - fer(g,0,0) * der(2), \tag{39}$$

$$der(1) * fer(f,2,0) \quad => \quad bos(f,3,0) - fer(f,2,0) * der(1), \tag{40}$$

$$der(2) * bos(f,3,0) \quad => \quad -fer(f,1,1) + bos(f,3,0) * der(2), \tag{41}$$

$$\begin{aligned} der(1) * bos(f,0,0,-1) \quad => \quad & -fer(f,1,0) * bos(f,0,0,-2) + \\ & bos(f,0,0,-1) * der(1), \end{aligned} \tag{42}$$

$$\begin{aligned} der(2) * axp(bos(f,0,0)) \quad => \quad & fer(f,2,0) * axp(bos(f,0,0)) + \\ & axp(bos(f,0,0)) * der(2). \end{aligned} \tag{43}$$

b) Left SuSy derivative

$$bos(f,0,0) * \mathbf{del}(\mathbf{1}) \quad => \quad -fer(f,1,0) + del(1) * bos(f,0,0), \tag{44}$$

$$fer(g,0,0) * \mathbf{del}(\mathbf{2}) \quad => \quad bos(g,2,0) - del(2) * fer(g,0,0), \tag{45}$$

$$fer(f,2,0) * del(2) \quad => \quad bos(f,3,0) - del(1) * fer(f,2,0), \tag{46}$$

$$bos(f,3,0) * del(2) \quad => \quad fer(f,1,1) + del(2) * bos(f,3,0), \tag{47}$$

$$\begin{aligned} bos(f,0,0,-1) * del(1) \quad => \quad & fer(f,1,0) * bos(f,0,0,-2) + \\ & del(1) * bos(f,0,0,-1), \end{aligned} \tag{48}$$

$$\begin{aligned} axp(bos(f,0,0)) * del(2) \quad => \quad & -fer(f,2,0) * axp(bos(f,0,0)) + \\ & del(2) * axp(bos(f,0,0)). \end{aligned} \tag{49}$$

From these examples follows that the second index in the fer, bos objects can take 0, 1, 2, 3 values only with the following meaning: 0 - no SuSy derivatives, 1 - first SuSy derivative, 2 - second SuSy derivative, 3 - first and second SuSy derivative.

Using the notations we obtain

$$\begin{aligned} der(1) * der(2) * bos(f,0,0) \quad => \quad & bos(f,3,0) + \\ & bos(f,0,0) * der(1) * der(2) + \\ & fer(f,1,0) * der(2) \\ & -fer(f,2,0) * der(1). \end{aligned} \tag{50}$$

For the “chiral ” representation, the meaning of the second argument in the *bos* or *fer* object is same as in the “traditional ” case while the actions of susy operators on the superfunctions are different. For example we have

$$der(1) * fer(f,1,0) => -fer(f,1,0) * der(1), \tag{51}$$

$$der(1) * fer(f,2,0) => bos(g,3,0) - fer(f,2,0) * der(1), \tag{52}$$

$$der(2) * bos(g,3,0) => -fer(g,2,1) + bos(g,3,0) * der(2) \tag{53}$$

$$bos(g,2,0) * del(2) => del(2) * bos(g,2,)). \tag{54}$$

For the "chiral1" representation we have different meanig of the second argument in the *bos* and *fer* object. Explicitely the values 0,1,2 in this second arguments denotes the values of the susy derivatives while 3 denotes the value of the commutator. Explicitey we have

$$\begin{aligned} der(3)*bos(f,0,0) \quad &=> \quad bos(f,3,0)+2*fer(f,1,0,0)*der(2) \\ &\quad\quad -2*fer(f,2,0)*der(1)+bos(f,0,0)*der(3) \qquad (55) \\ der(1)*fer(f,2,0) \quad &=> \quad (bos(f,3,0)-bos(f,0,1))/2-fer(f,2,0)*der(1). \qquad (56) \end{aligned}$$

The supersymmetric operators are always ordered in the case of "traditional" algebra as

$$\begin{aligned} der(2)*der(1) \quad &=> \quad -der(1)*der(2), \qquad (57) \\ del(2)*del(1) \quad &=> \quad -del(1)*del(2), \qquad (58) \\ der(1)*del(1) \quad &=> \quad d(1), \qquad (59) \\ der(1)*del(2) \quad &=> \quad -del(2)*der(1), \qquad (60) \end{aligned}$$

and similarly for others.

For the "chiral" algebra we postulate

$$\begin{aligned} der(2)*der(1) \quad &=> \quad -d(1)-der(1)*der(2), \qquad (61) \\ del(2)*del(1) \quad &=> \quad -d(1)-del(1)*del(2), \qquad (62) \\ der(1)*del(1) \quad &=> \quad 0, \qquad (63) \\ der(1)*del(2) \quad &=> \quad -d(1)-del(2)*der(1), \qquad (64) \end{aligned}$$

while for "chiral1" additionaly we have

$$\begin{aligned} der(3)*der(1) &=> -der(1)*d(1) \qquad (65) \\ der(1)*der(3) &=> der(1)*d(1) \qquad (66) \\ der(3)*der(2) &=> der(2)*d(1) \qquad (67) \\ der(2)*der(3) &=> -der(2)*d(1). \qquad (68) \end{aligned}$$

Please notice that if we would like to have the commponents of some $bos(f,3,0,-1)$ superfunction in the "chiral" representation then new object appear. Indeed,

$$b\_part(bos(f,3,0,-1),1) => \mathbf{fun(f1,0,f0,1,-1)}, \qquad (69)$$

We should consider this five indices object *fun* as

$$fun(f,n,g,m,-k) => (fun(f,n)-fun(g,m)/2)^{-k}. \qquad (70)$$

Similar interpretation is valid for other commands containing objects like $bos(f,3,n,-k)$

7 Action of the operators.

In order to have the value of the action of the given operator on some superfunction we introduce two operations pr and pg.

A)

$$pr(n, expression) \tag{71}$$

where n:=0,1,2,3. This command denotes the value itself of action of the SuSy derivatives on the given expression.For n=0 there is no SuSy derivative, n=1 corresponds to $der(1)$, n=2 to $der(2)$, while n=3 to $der(1) * der(2)$.

Example:

$$pr(1, bos(f, 0, 0)) => fer(f, 1, 0), \tag{72}$$

$$pr(3, fer(g, 0, 0)) => fer(f, 3, 0). \tag{73}$$

B) For the usual derivative we reserve command

$$pg(n, expression) \tag{74}$$

where n=0,1,2,...., denotes the value of the usual derivative on the expression

Example

$$pg(2, bos(f, 0, 0)) => bos(f, 0, 2) \tag{75}$$

8 Supersymmetric integration

There is one command **s_int**($number$, $expression$, $list$) only. This allows us to compute the value of supersymmetric integration of arbitrary polynomial expression constructed from fer and bos objects. It is valid in the traditional representation of the supersymmetry. The $numbers$ takes the following values: 0 → corresponds for usual "x" integration, 1 or 2 for the first or second supersymmetric index while 3 to the integration both over first and second indexes. The $list$ is the list of the names of the superfunctions over which we would like to integrate. The output of this command is in the form of the integrated part and non-integrated part. The non-integrated part is denoted by $del(-number)$ if $number = 1, 2, 3$ and by $d(-3)$ for 0.

Example

$$\mathbf{s\_int}(0, 2bos(f, 0, 1) * bos(f, 0, 1), \{f\}) = bos(f, 0, 0)^2, \tag{76}$$

$$s\_int(1, 2 * fer(f, 1, 0) * bos(f, 0, 0), \{f\}) = bos(f, 0, 0)^2, \tag{77}$$

$$s\_int(3, bos(f, 3, 0) * bos(g, 0, 0) + bos(f, 0, 0) * bos(g, 3, 0), \{f, g\}) = \tag{78}$$

$$bos(f, 0, 0) * bos(g, 0, 0) - \tag{79}$$

$$del(-3)\Big(fer(f, 1, 0) * fer(g, 2, 0) - fer(f, 2, 0) * bos(g, 1, 0)\Big). \tag{80}$$

9 Integration operators.

We introduced four different types of integration operators: right and left denoted as $d(-1)$ and $d(-2)$ respectively and moreover two different types of neutral integration operators $d(-3)$ and $d(-4)$. In first two cases they act acorrding to the formula

$$\mathbf{d(-1)} * bos(f, 0, 0) = \sum_{i=1}^{\infty} (-1)^i * bos(f, 0, i-1) * d(-1)^i, \tag{81}$$

for the right integration, while

$$bos(f,0,0) * \mathbf{d(-2)} = \sum_{i=1}^{\infty} d(-2)^i * bos(f,0,i-1), \tag{82}$$

for the left integration.

Before using these operators the precision of the integration must be specified by the declaration **ww := number**. If required this precision can be changed by clearing the old value of ww and introducing the new one. Both operators are defined by their action and by the properties

$$\begin{aligned} d(1) * d(-1) &= d(-1) * d(1) = d(2) * d(-1) = d(2) * d(-1) = 1, && (83)\\ & der(1) * d(-1) = d(-1) * der(1), && (84)\\ & d(-1) * del(1) = del(1) * d(-1), && (85)\end{aligned}$$

and analogously for $d(-2)$ and $der(2), del(2)$.

The neutral operator does not show up any action on some expression but has several properties. More precisly

$$\begin{aligned} d(1) * \mathbf{d(-3)} &= d(-3) * d(1) = d(2) * d(-3) = d(-3) * d(2) = 1, && (86)\\ & der(k) * d(-3) = d(-3) * der(k), && (87)\\ & d(-3) * del(k) = del(k) * d(-3), && (88)\end{aligned}$$

while for $d(-4)$

$$\begin{aligned} d(1) * \mathbf{d(-4)} &= d(-4) * d(1) = d(2) * d(-4) = d(-4) * d(2) = 1, && (89)\\ & der(k) * d(-4) = d(-4) * der(k), && (90)\end{aligned}$$

where k=1, 2.

From the last two formulas we see that $d(-3)$ operator is transparent under del operators while $d(-4)$ operators stops del action.

Similarly to $d(-3)$ or $d(-4)$ it is also possible to use the neutral differentation operator denote $\mathbf{d(3)}$. It has the properties

$$\begin{aligned} d(3) * d(-4) &= d(-4) * d(3) = d(3) * d(-3) = d(-3) * d(3) = 1, && (91)\\ & der(k) * d(3) = d(3) * der(k), && (92)\\ & d(3) * del(k) = del(k) * d(3), && (93)\end{aligned}$$

where k=1, 2.

We can have also " accelerated " integration operators denoted by $dr(-n)$ where n is a natural number. The action of these operators is exactly the same as $d(-1) * *n$ but instead of using n - times the integration formulas in the case $d(-1) * *n$, $dr(-n)$ uses only once the following formula

$$\mathbf{dr}(-\mathbf{n}) * bos(f,0,0) = \sum_{s=0}^{ww} (-1)^s \binom{n+s-1}{n-1} bos(f,0,s) dr(-n-s). \tag{94}$$

We have to, similarly to the $d(-1)$ case, declare also the "precision" of integration if we would like to use the "accelerated" integration operators. The switch **let cutoff** and command **cut** := **number** allows us to annihilate the higher order terms in the dr integrations procedure. Moreover, the switch **let drr** automatically changes usual integrations $d(-1)$ into "accelerated" integrations dr. The switch **let nodrr** changes dr integrations onto $d(-1)$.

10 Useful Commands.

A) Combinations.

We encounter, in many practical applications, problem of construction of different possible combinations of superfunction and super-pseudo-differential elements with the given conformal dimensions. We declare three different procedures in order to realize this requirement:

$$\mathbf{w\_comb}(list, n, m, x), \tag{95}$$
$$\mathbf{fcomb}(list, n, m, x), \tag{96}$$
$$\mathbf{pse\_ele}(n, list, m). \tag{97}$$

All these commands are based on the gradations trick (to associate with superfuction and superderivative the scaling parametr - conformal dimension). We consider here k/2 and k (k natural number and $k > 0$) gradation only.

Command *w_comb* gives the most general form of superfunctions combinations of given gradation. It is four argument procedure in which:

(i) first argument is a list in which each element is three elements list in which: first element is the name of the superfuction from which we would like to construct our combinations, second denotes its gradation while the last can take two values f - in the case where superfunction is superfermionic or b - for superbosonic.

(ii) second argument is a number - the desired gradation.

(iii) third argument is an arbitrary not numerical value which enumerates the free parameters in our combinations.

(iv) fourth argument takes two values

f - in the case when whole combinations should be fermionic or

b - for the bosonic nature of combination.

Examples:

$$\begin{aligned} w\_comb(\{\{f,1,b\},\{g,1,b\}\},2,z,b) \quad &=> \quad z1 * bos(f,3,0) + z2 * bos(f,0,1) + \\ & \qquad z3 * bos(f,0,0)^2; \qquad (98) \\ w\_comb(\{\{f,1,b\}\},3/2,g,f) \quad &=> \quad g1 * fer(f,1,0) + g2 * fer(f,2,0); \qquad (99) \end{aligned}$$

Command *fcomb*, simillarly to *w_comb*, gives us general form of an arbitrary combination of superfunctions modulo divergence terms. It is four argument command with the same meaning of arguments as in *w_comb* case. This command first calls *w_comb*, then eliminates in the canonical way SuSy - derivatives, by integrations by parts of *w_comb*. By canonical we understand that (SuSy) derivatives are removed first from the superfunction which is first in the list of superfuctions in fcomb command, next from second etc.

In order to illustrate cannonical manner of elimination of (SuSy) derivatives let us consider some expression which is constructed from f, g and h superfunctions and their (SuSy) derivatives. This expression is first splited onto three subexpression called $f-expression$, $g-expression$ and $h-expression$. $F-expression$ contains only combinations of f with f or g or (and) h, while $g-expression$ contains only combinations of g with g or h and last $h-expresion$ contains only combinations of h with h. Command $fcomb$ removes first (SuSy) derivatives from f in f-exprssion, next from g in g-expression, and finally from h in h-expression. Let us present such situation on the following example

$$fer(f,1,0)*fer(g,2,0)+bos(g,0,0)*bos(g,3,0). \qquad (100)$$

Let us now assume that we have f,g order then $f-expression$ is $fer(f,1,0)*fer(g,2,0)$, while $g-expression$ is $bos(g,0,1)*bos(g,3,0)$. Now canonical elimination gives us

$$-bos(f,0,0)*bos(g,3,0)+2*bos(g,0,0)*bos(g,3,1), \qquad (101)$$

while assuming g,f order we obtain

$$-bos(f,3,0)*bos(g,0,0)+2*bos(g,0,0)*bos(g,3,1) \qquad (102)$$

Example

$$\begin{aligned} fcomb(\{\{u,1\}\},4,h) \quad => \quad & h(1)*fer(u,2,0)*fer(u,1,0)*bos(u,0,0)+ \\ & h(2)*bos(u,3,0)*bos(u,0,0)^2+ \\ & h(3)*bos(u,0,2)*bos(u,0,0)+ \\ & h(4)*bos(u,0,0)^4; \end{aligned} \qquad (103)$$

Finally, comand pse_ele gives us the general form of element which belongs to algebra of pseudo-SuSy derivative algebra [3]. Such element can be symbolically written down as

$$(bos+fer*der(1)+fer*der(2)+bos*der(1)*der(2))*d(1)^n, \qquad (104)$$

for the traditional and "chiral" representation while for "chiral1" as

$$(bos+fer*der(1)+fer*der(2)+bos*der(3))*d(1)^n, \qquad (105)$$

where at the moment, bos and fer denotes some an arbitrary superfunctions. The mentioned command allows us to obtain such element of the given gradation which is constructed from some set of superfunctions of given gradation. This command is three arguments.

$$\mathbf{pse\_ele}(wx,wy,wz), \qquad (106)$$

First index denotes the gradation of SuSy-pseudo-element. Second the names and gradations of the superfunctions from which we would like to construct our element. This second index wy is in the form of list exactly the same as in the w_comb command. Last index denotes the names which enumerates the free parameters in our combination.

B) Parts of the pseudo-SuSy-differential elements.

In order to obtain the components of the (pseudo)-SuSy element we have three different commands:

$$\mathbf{s\_part}(expression, n), \tag{107}$$

$$\mathbf{d\_part}(expression, m), \tag{108}$$

$$\mathbf{sd\_part}(expression, n, m), \tag{109}$$

where n,m=0,1,2,3,....

The *s_part* gives us coefficient standing in n-th SuSy derivative. However notice, that for n=3 we should consider the coefficients standing in the $der(1) * der(2)$ operator for the traditional or chiral representations while for the chiral1 representation the terms standing in the $der(3)$ operator. The *d_part* command give us the coefficients standing in same power of d(1), while *sd_part* the term standing in n-th SuSy derivative and m-th power of usual derivative.

Example:

$$\begin{aligned} ala: \quad &= \quad bos(g,0,0) + fer(f,3,0) * der(1) + (fer(h,2,0) * der(2) + \\ & \quad bos(r,0,0) * der(1) * der(2)) * d(1); && (110) \\ s\_part(ala,3) \quad &=> \quad fer(f,3,0); && (111) \\ d\_part(ala,1) \quad &=> \quad fer(h,2,0) * der(2) + \\ & \quad bos(r,0,0) * der(1) * der(2); && (112) \\ sd\_part(ala,0,0) \quad &=> \quad bos(g,0,0); && (113) \end{aligned}$$

C) Adjoint.

The adjoint of some SuSy operator is defined in standard form as

$$<< \alpha, PP * \beta >>=<< \beta, PP^{*}\alpha >> \tag{114}$$

where α and β are the test superboson functions, PP is the opertor under consideration and $<< \alpha, \beta >>$ is a scalar product defined as

$$<< \alpha, \beta >>= \int \alpha * \beta * d\theta_1 * d\theta_2 \tag{115}$$

where we use the Berezin integral definition [1]

$$\int \theta_i * d\theta_j = \delta_{i,j}, \tag{116}$$

$$\int d\theta_i = 0. \tag{117}$$

For this operation we have command

$$\mathbf{cp}(expression); \tag{118}$$

Examples:

$$\begin{aligned} cp(der(1)) \quad &=> \quad -der(1), && (119) \\ cp(del(1) * fer(r,1,0) * der(1)) \quad &=> \quad fer(r,1,1) + fer(r,1,0) * d(1) - \end{aligned}$$

$$del(1) * bos(r,0,1), \tag{120}$$

From the last example there follows that it is possible to define $cp(del(1)*fer(r,1,0)*der(1))$ in the different but equivalent manner namely as $fer(r,1,0)*d(1)-bos(r,0,1)*der(1)$.

From the practical point of view, we do not define the conjugation for the $d(-1)$ and $d(-2)$ operators, because then we should define the precision of the action of the operators $d(-1)$ or $d(-2)$ and even then, we would obtain very complicated formulas. However, if somebody decides to use this conjugation to the $d(-1)$ or to the $d(-2)$, it is recommended, first to change by hand, these operators on $d(-3)$, next to compute cp and change once more $d(-3)$ into $d(-1)$ or $d(-2)$ together with the declaration of the precision.

D) Projection.

In many cases, especially in SuSy approach to soliton theory we have to obtain projection onto the invariant subspace (with respect to commutator) of algebra of pseudo-SuSy-differential algebra. There are three different subspaces [4] and hence we have two argument command

$$\mathbf{rzut}(expression, n) \tag{121}$$

in which n=0, 1, 2.

Example

$$\begin{aligned} ewa: \quad &= \quad (bos(f,0,0)+fer(f1,1,0)*der(1)+fer(f2,2,0)*der(2)+ \\ & \quad bos(f3,0,0)*der(1)*der(2))+(bos(g,0,0)+ \\ & \quad fer(g1,1,0)*der(1)+fer(g2,2,0)*der(2)+ \\ & \quad bos(g3,0,0)*der(1)*der(2))*d(1), \qquad (122) \\ rzut(ewa,0) \quad &=> \quad ewa, \qquad (123) \\ rzut(ewa,1) \quad &=> \quad ewa-bos(f,0,0); \qquad (124) \\ rzut(ewa,2) \quad &=> \quad bos(f3,0,0)*der(1)*der(2)+(fer(g1,1,0)*der(1) \\ & \quad +fer(g2,2,0)*der(2)+ \\ & \quad bos(g3,0,0)*der(1)*der(2))*d(1), \qquad (125) \end{aligned}$$

E) Analogon of coeff.

Motivated by practical applications, we constructed for our supersymmetric functions three commands, which allow us to obtain the list of the same combinations of some superfunctions and (SuSy) derivatives from some given operator-valued expression.

The first command is one argument

$$\mathbf{lyst}(expression) \tag{126}$$

with the output in the form of list.

Example

$$magda := fer(f,1,0)*fer(f,2,0)*a1+der(1), \tag{127}$$

$$lyst(magda) => \{fer(f,1,0)*fer(f,2,0)*a1, der(1)\}, \tag{128}$$

The second command is also one argument

$$\textbf{lyst1}(expression) \tag{129}$$

with the output in the form of list in which each element is constructed from coefficients and (SuSy) operators of corresponding element in *lyst* list. For example

$$lyst1(magda) => \{a1, der(1)\}, \tag{130}$$

The third command is also one argument

$$\textbf{lyst2}(expression) \tag{131}$$

with the output in the form of list in which each element is constructed from coefficients standing in the given expression. For exampla

$$lyst2(magda) => \{a1, 1\} \tag{132}$$

F) Simplifications.

If we encounter during the process of computations such expression

$$fer(f,1,0) * d(-3) * fer(f,2,0) * d(1) \tag{133}$$

it is not reduced further. On the other side we can replace $d(1)$ onto $d(2)$ and back $d(2)$ onto $d(1)$. In order to do such replacement we have the command

$$\textbf{chan}(expression) \tag{134}$$

Example

$$\begin{aligned} &chan(fer(f,1,0) * d(-3) * fer(f,2,0) * d(1)) => \\ &-fer(f,2,0) * fer(f,1,0) - fer(f,1,0) * d(-3) * fer(f,2,1). \end{aligned} \tag{135}$$

Notice that as the result we kill the d(1) operation.

G) O(2) invariance.

In many cases in the supersymmetric theories we deal with the O(2) invariance of SuSy indices. This invariance follows from the physical assumption on the nonprivileging the "fermionic" coordinates in the superspace. In order to check whether our formula posseses such invariance we can use

$$\textbf{odwa}(expression) \tag{136}$$

This procedure replaces in the given expresion $der(1)$ onto $-der(2)$ and $der(2)$ onto $der(1)$. Next, it changes, in the same manner, the values of the action of these operators on the superfunctions.

F) Macierz

Similarly to the representation of the superfunctions in the components We can define the supercomponent form for the *pse_ele* objects similarly to the representation of the supersfunctions. Usually we can consider such object as the matrix which acts on the components of the superfunctions.It is realized in our program using the command :

$$\textbf{macierz}(expression, x, y), \tag{137}$$

where expression is the formula under consideration while x can take two values f or b depending wheather we would like to conside bosonic (b) part or fermionic (f) part of the expression. Last index in this command denotes the option in which we acts on the bosonic or fermionic superfunction. It takes two values f- for fermionic test superfunction or b - for bosonic case. More explicitely we obtain

$$macierz(der(1) * der(2), b, f) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & d(1) & 0 \\ 0 & -d(1) & 0 & 0 \\ -d(1) * *2 & 0 & 0 & 0 \end{pmatrix} \tag{138}$$

$$macierz(der(1) * der(2), f, b) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & d(1) \\ -d(1) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \tag{139}$$

11 Functional gradients.

In SuSy soliton approach we very frequently encounter problem of computing the gradient of the given functional. The usual definition of the gradient [2] is adopted, in the supersymmetry also.

$$H^{'}[v] =< gradH, v >, \tag{140}$$

$$H^{'}[v] = \frac{\partial}{\partial \epsilon} H(u + \epsilon v) \mid_{\epsilon=0}, \tag{141}$$

where H denotes some functional which depends on u. v denotes vector under which we compute the gradient and $<, >$ the relevant scalar product.

We implemented all that in our package for the “tradicional ” algebra only. In order to compute the gradient with respect to some superfuction use

$$\textbf{gra}(expression, f), \tag{142}$$

where ”expression” is the given density of the functional, while f denotes the first index in the superfunction (name of the superfunction).

Example

$$gra(bos(f, 3, 0) * fer(f, 1, 0), f) => -2 * fer(f, 2, 1) \tag{143}$$

For practical use we perform two additional commands:

$$\textbf{dyw}(expression, f) \tag{144}$$

$$\textbf{war}(expression, f). \tag{145}$$

The first computes the variation of expression with respect to superfunction f, next removes (via integrations by parts) SuSy- derivatives from varied functions and finally produces list of factorized fer and bos superfunctions. When the given expression is full (SuSy)-derivative, the result of the dyw command is 0 and hence this command is very usefull in verifications of (SuSy)-divergences of expressions.

When result of applications of dyw command is not zero then we would like to have the system of equations on the coefficients standing in the same factorized fer and bos superfunction. We can quickly obtain such list using command $war(expression, f)$ with the same meaning of arguments as in the dyw command.

Examples

$$xxx := fer(f,1,0) * fer(f,2,0) + x * bos(f,3,0)^2; \tag{146}$$

$$\begin{aligned} dyw(xxx,f) \quad => \quad & \{-2 * bos(f,3,0) * bos(f,0,0), \\ & -2 * x * bos(f,0,2) * bos(f,0,0)\} \end{aligned} \tag{147}$$

$$war(xxx,f) => \{-2, -2 * x\}. \tag{148}$$

12 Conservation Laws.

In many cases we would like to know whether the given expression is a conservation law for some Hamiltonian equation. We can quikly check it using

$$\textbf{dot\_ham}(equation, expression) \tag{149}$$

where "equation" is a set of two elements list in which first element denotes the function while the second its flow. The second argument should be understand as the density of some conserved current. For example, for SuSy version of the Nonlinear Schrodinger Equation [7] we obtain

$$\begin{aligned} ew: \quad = \quad & \{\{q, -bos(q,0,2) + bos(q,0,0)^3 * bos(r,0,0)^2 - \\ & 2 * bos(q,0,0) * pr(3, bos(q,0,0) * bos(r,0,0))\}, \\ & \{r, bos(r,0,2) - bos(q,0,0)^2 * bos(r,0,0)^3 + \\ & 2 * bos(r,0,0) * pr(3, bos(q,0,0) * bos(r,0,0))\}\}, & (150) \\ ham: \quad = \quad & bos(q,0,1) * bos(r,0,0) + x * bos(q,0,0)^2 * bos(r,0,0)^2, & (151) \\ yyy: \quad = \quad & dot\_ham(ew, ham). & (152) \end{aligned}$$

As the result of previous computations we have a complicated expression which is not zero. We woulld like to interpreted it as a full (SuSy)-divergence and we can quickly verify it, if we use command war. We can solve, obtained list of equations, using known techniques. For example, in our previous case we obtain

$$war(yyy,q) => \{-4 * x, -8 * x, -4 * x\}; \tag{153}$$

$$war(yyy,r) => \{4 * x, 8 * x, 4 * x\}; \tag{154}$$

and we conclude that our ham is a constant of motion if x=0.

It is also possible to use command *dot_ham* to the pseudo-SuSy-differential element what is very useful in SuSy approach to Lax operator in which we would like to check validity of the formula

$$\partial_t * L := [L, A]. \tag{155}$$

where A is a some (SuSy) operator.

13 Jacobi Identity.

The Jacobi identity for some SuSy - hamiltonian operators is verified using the relation

$$<< \alpha, P^{'}_{(P\beta)} * \gamma >> + cyclic\ permutation(\alpha, \beta, \gamma), \tag{156}$$

where $P^{'}$ denotes the directional derivative along the $P(\beta)$ vector and $<<, >>$ scalar product. Directional derivative is defined in the standard manner as [44]

$$F^{'}(u)[v] = \frac{\partial}{\partial \epsilon} F(u + \epsilon v) \mid_{\epsilon=0}, \tag{157}$$

where F is some functional depending on u. V is a directional vector.

In this package we have several commands which allow us to verify the Jacobi identity. We have the possibility to compute, indenpendently of veryfing Jacobi identity, directional derivative for the given Hamiltonian operator along the given vector using

$$\mathbf{n\_gat}(pp, wim) \tag{158}$$

where pp is scalar or matrix Hamiltonian operator. Wim denotes components of a vector along which we compute derivative and has the form of list in which each element has following representation

$$bos(f) => < expression > . \tag{159}$$

The $bos(f)$, in the last formula, denotes the shift of $bos(f, 0, 0)$ superfunction according to definition of directional derivative.

In order to compute Jacobi identity use command

$$\mathbf{fjacob}(pp, wim), \tag{160}$$

with the same meaning of *pp* and *wim* as in *n_gat* command.

Notice that ordering of components in *wim* list is important and is connected with interpretation of components of Hamiltonian operator *pp* as a set of Poisson brackets constructed just from elements of *wim* list. For example, in our scheme, first component of wim is always connected with element, from which we create Poisson bracket and which corresponds to first element on the diagonal of pp, second element of *wim* with second element on diagonal of *pp* and etc.

As the result of applications of $fjacob$ command to some Hamiltonian operator we obtain a complicated formula, not necesarily equal to zero but which would be expressed as (SuSy) divergence. However, we can quickly verify it using the same method as in *dot_ham* command which has been described in previous section.

Usually, after the application of the $fjacob$ command to some matrix Hamiltonian operator we obtain the hudge expression which is too complicated to analyze even when we would like to check its (SuSy)divergence. In this case we could extract from $fjacob$ expression terms containing given components of vector test functions fixed by us. We can use in this order command

$$\mathbf{jacob}(pp, wim, mm) \tag{161}$$

where pp and wim has the same meaning as in $fjacob$ case while mm is a three elements list denoting the components of α, β, γ.

This command is not prepered to compute in full the Jacobi identity, which contains the integrations operators. We do not implement here the symbolic integrations of superfunctions in order to simplify the final results.

14 The list of Objects, Commands and Switches

Objects:

bos(f,n,m) **bos**(f,n,m,k) **fer**(f,n,m) **axp**(f) **fun**(f,n)
fun(f,n,m) **gras**(f,n) **axx**(f) **d**(1) **d**(2)
d(3) **d**(-1) **d**(-2) **d**(-3) **d**(-4)
dr(-n) **der**(1) **der**(2) **del**(1) **del**(2)

Commands

fpart(expression) **bpart**(expression) **bf_part**(expression,n)
b_part(expression,n) **pr**(n,expression) **pg**(n,expression)
w_comb ({ { f,n,x },...} ,m,z,y) **fcomb** ({ { f,n,x },...},m,z,y) **pse_ele** (n,{ { f,n },... },z)
s_part(expression,n) **d_part**(expression,n) **sd_**(expression,n,m)
cp(expression) **rzut**(expression,n) **lyst**(expression)
lyst1(expression) **lyst2**(expression) **chan**(expression)
odwa(expression) **gra**(expression,f) **dyw**(expression,f)
war(expression,f) **dot_ham**(equations,expression) **n_gat**(operator,list)
fjacob(operator,list) **jacob**(operator,list,{ α, β, γ }) **macierz**(expression,x,y)
s_ int(numbers, expession,list)

Switches

trad **chiral** **chiral1** **inverse** **drr** **nodrr**

15 Acknowledgement

The author would like to thank to dr. W.Neun for valuable remarks.

References

[1] J.Wess and J.Bagger, "Supersymmetry and Supergravity" (Princeton, NJ 1982);

[2] S.Ferrara and J.G.Taylor "Introduction to Supergravity" (Moscow 1985).

[3] L.Faddeev and L.Takhtajan "Hamiltonian Methods in the Theory of Solitons " (Springer-VerlaG 1987); A.Das "Integrable Models" (World Sci.1989); M.Ablowitz and H.Segur "Solitons and the Inverse Scattering Transform" (SIAM Philadelphia 1981).

[4] A.Polyakov in "Fields, Strings and critical Phenomena" ed.E.Brezin and J.Zinn-Justin, (North Holland 1989).

[5] S.Manakov, S.Novikov, L.Pitaevski and V.Zakharov "Soliton Theory: The Inverse Problem" (Nauka, Moscow (1980).

[6] L.A.Dickey "Soliton Equations and Hamiltonian Systems", (World Scientific, Singapore 1991).

[7] E.Date, M.Jimbo, M.Kashiwara and T.Miwa, in “Nonlinear Integrable Systems - Classical and Quantum Theory”, ed. by M.Jimbo and T.Miwa , (World Scientific, Singapore 1983) p. 39.

[8] B.Kupershmidt, “Elements of Superintegrable Systems”(Kluwer 1987).

[9] M.Chaichian, P.Kulish Phys.Lett.18B (1978) 413.

[10] R.D’Auria and S.Sciuto, Nucl. Phys. B.171 (1980) 189.

[11] M.Gurses and O.Oguz, Phys.Lett.108A (1985) 437.

[12] Y.Manin and R.Radul, Commun.Math.Phys. 98 (1985) 65 .

[13] C.Morosi and L.Pizzochero, Commun.Math.Phys. 158 (1993) 267 .

[14] C.Morosi and L.Pizzochero, J.Math.Phys.35 (1994) 2397.

[15] C.Morosi and L.Pizzochero ”A Fully Supersymmetric AKNS Theory ” preprint of Dipartimento di Matematica,Politecnico di Milano. April 1994, to appeare in Commun.math.Phys.(1995).

[16] P.Mathieu, J.Math.Phys.29 (1988) 2499.

[17] C.A.Laberge and P.Mathieu, Phys.Lett.215B (1988) 718 .

[18] P.Labelle and P.Mathieu, J.Math.Phys.32 (1991) 923 .

[19] M.Chaichian and J.Lukierski, Phys.Lett. 183B (1987) 169 .

[20] T.Inami and H.Kanno, Commun.Math.Phys. 136 (1991) 519 .

[21] S.K.Nam, Intern.J.Mod.Phs.4 (1989) 4083.

[22] K.Hiutu and D.Nemeschansky, Mod.Phys.Lett.A. 6 (1991) 3179.

[23] C.M.Yung, Phys.Lett. 309B (1993) 75.

[24] E.Ivanov and S.Krivons, Phys.Lett.291B (1992) 63.

[25] P.Kulish Lett.Math.Phys.10 (1985) 87.

[26] G.H.M.Roelofs and P.H.M. Kersten Journ. Math.Phys.33 (1992) 2185 .

[27] J.C.Brunelli and A.Das, Journ.Math.Phys.36 (1995) 268.

[28] F.Toppan, Int.Journ.Mod.Phys.A10 (1995) 895.

[29] S.Krivonos and A.Sorin “The minimal N=2 superextension of the NLS equation” hep-th/9504084 to appeare in Phys.lett.B.

[30] S.Krivonos, A.Sorin and F.Toppan “ On the Super-NLS Equation and its Relation with N=2 Super-KdV within Coset Approach” hep-th/9504138.

[31] W.Oevel and Z.Popowicz, Commun.Math.Phys. 139 (1991) 441.

[32] Z.Popowicz J.Phys.A:Math.Gen. 19 (1986) 1495.

[33] Z.Popowicz J.Phys.A:Math.Gen. 23 (1990) 1127.

[34] Z.Popowicz Phys.Lett.319B (1993) 478.

[35] Z.Popowicz Phys.Lett. 174A (1993) 411.

[36] Z.Popowicz Int.Jour.Mod.Phys. 9 (1994) 2001.

[37] Z.Popowicz Phys.Lett. 194A (1994) 375.

[38] Z.Popowicz J.Phys.A.Math.Gen. 29 (1996) 1281.

[39] M.Olshanetsky, Comm.Math.Phys. 88 (1983) 63.

[40] J.Evans and T.Hollowood, Nucl.Phys. B352 (1991) 723.

[41] A.C.Hearn "REDUCE user's manual 3.6" (Rand.Publ.1995).

[42] P.K.H.Gragert and P.H.M.Kersten "Liesuper" ftp.math.utwente.nl/pub/rweb /appl/lisuper.web

[43] S.Krivonos and K.Thielmans "A Mathematica Package for Computing N=2 Superfield Opeerator Product Expansion", Preprint Imperial College London TP 95-96/13 or hep-th/9512029.

[44] B.Fuchssteiner and A.S.Fokas, Physica 4D (1981) 718.