COMPUTER SIMULATION OF THE SINE-GORDON EXACT SOLITONS AND SOLITON COLLISIONS

©2007 Mykola PRYTULA [1], Yarema PRYKARPATSKY [2,3,4], Mykola STARCHAK [1]

[1] Ivan Franko Lviv National University,
1 Universytetska Str., Lviv 79000, Ukraine
[2] Instytut of Mathematics of NASU,
3 Tereshchenkivska Str., Kyiv 01601, Ukraine
[3] Ivan Franko State Pedagogical University,
24 Ivan Franko Str., Drogobych 82100, Ukraine
[4] Pedagogical Academy, Krakow, 30062 Poland

Received September 13, 2007

The article is devoted to devising a PC-program product, based on the MAPLE package, for analyzing dynamical properties of many-soliton and soliton-wise exact solutions to nonlinear dynamical systems on functional manifolds. In particular, as an example, there is treated the sine-Gordon nonlinear completely integrable Hamiltonian dynamical system, whose two-dimensional animation and three-dimensional graphic of solitonic collisions are presented.

1. INTRODUCTION

The localized excitations propagating in a nonlinear system with constant velocity and colliding with each other without change in their shapes are called [4, 6] solitons. During the collision of solitons the solution cannot be represented as a linear combination of two soliton solutions but after the collision solitons recover their shapes and the only result of collision is a phase shift.

The sine-Gordon equation

> **restart:**

MSC 2000: 34A30, 34B05, 34B15

> **diff(phi(x,t),t2)** $- diff(phi(x,t),x$**2)+sin(phi(x,t))=0;**

$$\left(\frac{\partial^2}{\partial t^2}\phi(x,t)\right) - \left(\frac{\partial^2}{\partial x^2}\phi(x,t)\right) + \sin(x,t) = 0$$

plays an important role in many branches of physics. It provides one of the simplest models of the unified field theory, can be found in the theory of dislocations in metals, in the theory of Josephson junctions [8, 10] and so on. It can be used also in interpreting certain biological processes like DNA dynamics. As one can show by means of the gradient–holonomic algorithm [2, 6], this equation is equivalent to a completely integrable nonlinear Hamiltonian system on a functional manifold, called SG-dynamical system. In particular, it possesses an infinite hierarchy of commuting to each other nontrivial conservation laws, which are often very useful at computer analysis of solutions. Moreover, this SG-dynamical system appeared to possess [3] two compatible Poissonian structures on this functional manifold, allowing to construct many nontrivial analytical relationships between its solutions, one of which is known in the literature [1, 6] as the Backlund transformation.

2. BACKLUND TRANSFORMATIONS AND NONLINEAR SUPERPOSITION PRINCIPLE

In the last century a Sweden mathematician Backlund, considering the geometry of surfaces with constant negative curvature, showed a way to obtain the hierarchy of sine-Gordon solutions when a new solution can be build on the bases of known solutions. The transformation, as applied to the sine-Gordon equation, has the form [1]:

```
> Backl_Trans:={diff((phi(xi,tau)+psi(xi,tau))/2,xi)
=a*sin((phi(xi,tau)-psi(xi,tau))/2),
diff((phi(xi,tau)-psi(xi,tau))/2,tau)
=(1/a)*sin((phi(xi,tau)+psi(xi,tau))/2)};
```

where $\psi = \frac{x-t}{2}, \tau = \frac{x+t}{2}$, a — transformation parameter, ϕ, ψ — solutions of the equation $\frac{\partial^2}{\partial\tau\partial\xi}\phi(\xi,\tau) = \sin(\phi(\xi,\tau))$.

Assuming that the diagram for the construction of the solutions is commutative, one can eliminate the partial derivatives and find the analytical expression of the Backlund transformations

```
> phi[n+1]:=phi[n-1]+4*arctan(((a[1]+a[2])
/(a[1]-a[2]))*(tan((phi1[n]-phi2[n])/4)));
```

$$\phi_{n+1} := \phi_{n-1} + 4\arctan\left(\frac{(a_1+a_2)\tan(\frac{1}{4}\phi 1_n - \frac{1}{4}\phi 2_n)}{a_1 - a_2}\right),$$

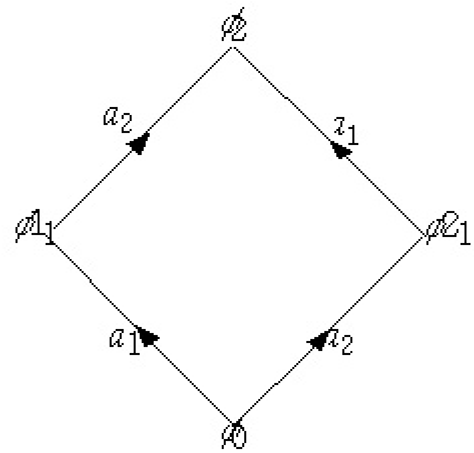

Fig. 1: Diagram 1

where a_1, a_2 — are the parameters of transformation, ϕ_n is the n-parametric solution. This formula gives a way to build the multi-soliton solutions.

Procedure „Backlund" realizes the algorithm of Backlund transformation in Maple.

```
> Backlund := proc (a1,a2,phi0,phi1,phi2)
RETURN(phi0+4*arctan(((a1+a2)*tan((phi1-phi2)/4))
/(a1-a2)))
end;
```

3. ONE-SOLITON SOLUTIONS

Obviously, the sine–Gordon equation has a trivial solution $\phi = 0$. Substituting the trivial solution into the system „Backl_Trans", one obtains the one-soliton solutions Kink and Antikink.

```
> Soliton := proc(x,t,a,x_0)
local xi,tau:
xi:=(x-x_0+t)/2;
tau:=(x-x_0-t)/2;
RETURN(4*arctan(exp(a* xi + tau /a))) end;
```

where x, t — space and time coordinates, a — parameter of transformation for Kink (Antikink), x_0 — initial position of Kink (Antikink).

AntiKink

```
> with(plottools):
> with(plots):
> v[K]:=0.1: # velocity of the Kink
> x_K:=-4*sqrt(1-v[K]^2): # initial state of the Kink
> a:=sqrt((1-v[K])/(1+v[K])): # parameter of transformation
> L:=40*sqrt(1-v[K]^2): T:=(L-2*x_K)/v[K]:
> K:=(x,t)->Soliton(x,t,a,x_K):
> animate(K(x,t),x=0..L,t=0..T,title="Kink",
color=red,view=[0..L,-1..2*Pi+1]);
```

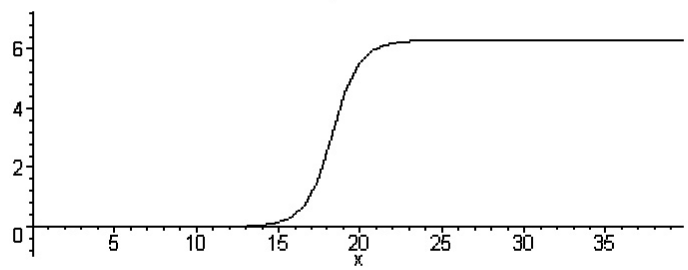

Fig. 2: Kink

Kink is the transition from one stationary solution, $\phi = 0$, to another, $\phi = 2\pi$.

AntiKink

AntiKink

```
> v[AK]:=0.1: # velocity of the AntiKink
> x_AK:=-4*sqrt(1-v[AK]^2): # initial state of the AntiKink
> a:=-sqrt((1-v[AK])/(1+v[AK])): # parameter
of transformation
> L:=40*sqrt(1-v[AK]^2): T:=(L-2*x_AK)/v[AK]:
> AK:=(x,t)->Soliton(x,t,a,x_AK):
> animate(AK(x,t),x=0..L,t=0..T,title="AntiKink",
color=red,view=[0..L,-1..2*Pi+1]);
```

Antikink, in contrast to the kink, is the transition from the solution $\phi = 2\pi$ to $\phi = 0$.

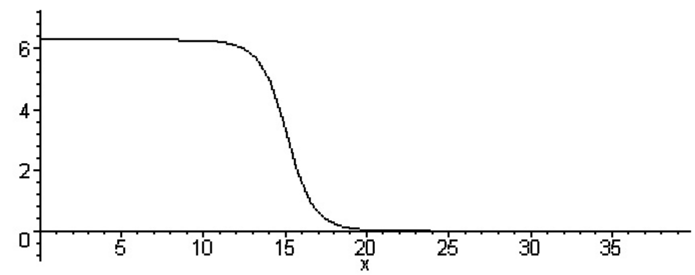

Fig. 3: AntiKink

4. TWO-SOLITONS SOLUTIONS

By applying the Backlund transformation to the trivial solution and one-soliton solution,the two-soliton solutions can be obtained.

```
> TwoSoliton := proc(x,t,a1,x_1,a2,x_2)
local phi0, phi1, phi2:
phi0[0]:=(x,t)->0;
phi1[1]:=(x,t)-> Soliton(x,t,a1,x_1);
phi2[1]:=(x,t)-> Soliton(x,t,a2,x_2);
phi1[2]:=(x,t)-> Backlund(a1,a2,phi0[0](x,t),phi1[1](x,t),
phi2[1](x,t));
RETURN(phi1[2](x,t)) end;
```

where x, t — space and time coordinates, $a1, a2$ — parameters of transformation, x_1, x_2 — initial positions.

Examples of two-soliton solutions and two-soliton collisions:

Kink-Kink Collision

```
> v[K_K]:=0.2: # velocity of the Kink-Kink
> x_K_K:=0: # initial state of the Kink-Kink
> a[1]:=-sqrt((1-v[K_K])/(1+v[K_K])): # parameters of
> a[2]:=sqrt((1+v[K_K])/(1-v[K_K])): # transformation
> L:=10/sqrt(1-v[K_K]^2): T:=L/v[K_K]:
> K_K:=(x,t)->TwoSoliton(x,t,a[1],x_K_K,a[2],x_K_K):
> animate(K_K(x,t),x=-3*L..3*L,t=-4*T..4*T,title
="Kink-Kink Collison",color=red,view
=[-3*L..3*L,-2*Pi-1..2*Pi+1]); >
```

Kink-Kink Collision

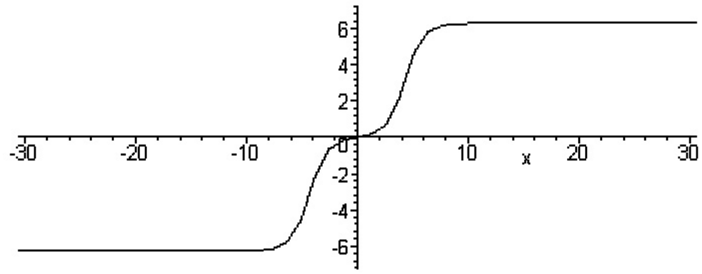

Fig. 4: Kink-Kink Collision

Before the collision, Kinks move toward each other with equal velocities. It seems that in the collision they repel each other, however, they actually pass through each other without change in their velocities.

Kink-AntiKink Collision

```
> v[K_AK]:=0.2: # velocity of the Kink-AntiKink
> x_K_AK:=0: # initial state of the Kink-AntiKink
> a[1]:=sqrt((1-v[K_AK])/(1+v[K_AK])): # parameters of
> a[2]:=sqrt((1+v[K_AK])/(1-v[K_AK])): # transformation
> L:=10/sqrt(1-v[K_AK]^2): T:=L/v[K_AK]:
> K_AK:=(x,t)->TwoSoliton(x,t,a[1],x_K_AK,
a[2],x_K_AK):
> animate(K_AK(x,t),x=-3*L..3*L,t=-2*T..2*T,title
="Kink-AntiKink Collision",
color=red,view=[-3*L..3*L,-2*Pi-1..2*Pi+1]); >
```

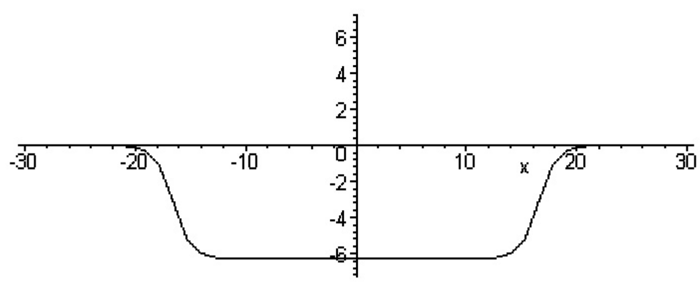

Fig. 5: Kink-AntiKink Collision

Before the collision, Kink and Antikink move toward each other with equal velocities. After the collision they move away with the same velocities but in the neighboring level.

5. BREATHER

Another two-soliton solution is the Breather. Breather is an oscillatory localized excitation, which is the Kink and Antikink bound together. Kink and Antikink have no enough energy to overcome their mutual attraction and that is why they make the oscillatory system.

In the following the examples of Breather solutions are given.

Standing Breather

```
> v[B]:=0: # velocity of the Breather
> omega:=0.154: # frequency of the Breather
> x_B:=0: # initial state
> R:=sqrt((1-v[B])/(1+v[B])):
> eta:=sqrt(1-omega^2):
> a[1]:=R*(eta+I*omega): # parameters of
> a[2]:=R*(eta-I*omega): # transformation
> T:=2*Pi/omega: L:=4*arctan(eta/omega):
> B:=(x,t)->TwoSoliton(x,t,a[1],x_B,a[2],x_B):
> animate(B(x,t),x=-2*L..2*L,t=-T..T,title=
"Standing Breather",color=red,view=[-2*L..2*L,-L-1..L+1]); >
```

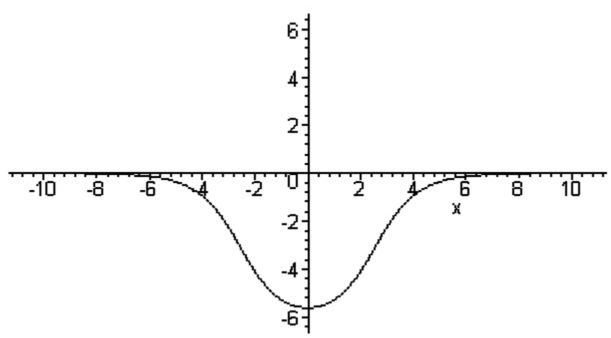

Fig. 6: Standing Breather

Large amplitude Breather

```
> v[B]:=0.2: # velocity of the Breather
> omega:=0.154: # frequency of the Breather
> x_B:=0: # initial state
```

```
> R:=sqrt((1-v[B])/(1+v[B])):
> eta:=sqrt(1-omega^2):
> a[1]:=R*(eta+I*omega): # parameters of
> a[2]:=R*(eta-I*omega): # transformation
> T:=2*2*Pi/omega: L:=T*abs(v[B])/(sqrt(1-v[B]^2)):
> B:=(x,t)->TwoSoliton(x,t,a[1],x_B,a[2],x_B):
> animate(B(x,t),x=0..L,t=0..T,title=
"Large amplitude Breather",color=red);
>
```

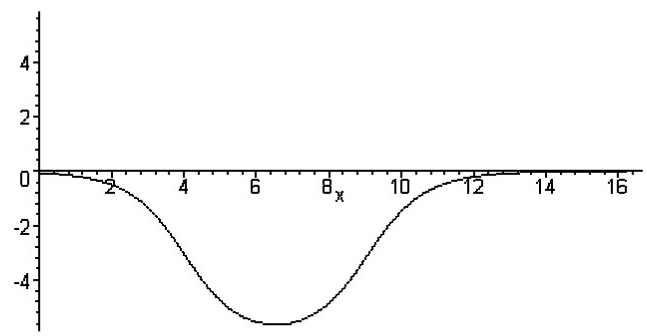

Fig. 7: Large amplitude Breather

Small amplitude Breather

```
> v[B]:=0.85: # velocity of the Breather
> omega:=0.99: # frequency of the Breather
> x_B:=0: # initial state
> R:=sqrt((1-v[B])/(1+v[B])):
> eta:=sqrt(1-omega^2):
> a[1]:=R*(eta+I*omega): # parameters of
> a[2]:=R*(eta-I*omega): # transformation
> T:=2*2*Pi/omega: L:=T*abs(v[B])/(sqrt(1-v[B]^2)):
> B:=(x,t)->Two-Soliton(x,t,a[1],x_B,a[2],x_B):
> animate(B(x,t),x=-2*L..2*L,t=-3.5*T..3.5*T,title
="Small amplitude Breather",
color=red,view=[-1.5*L..1.5*L,-0.9..0.9],numpoints=260); >
```

Small amplitude Breather looks exotically but its envelope has the form of large amplitude Breather.

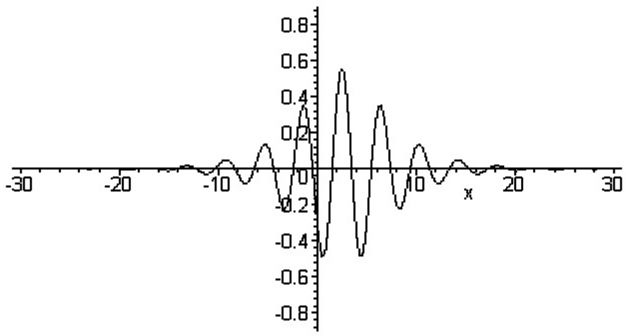

Fig. 8: Small amplitude Breather

6. THREE-SOLITONS SOLUTIONS

The three-soliton solutions can be obtained according to the diagram called „soliton ladder“,

which is realized in the following procedure

```
> ThreeSoliton := proc (x,t,a1,x_1,a2,x_2,a3,x_3)
local phi,phi1,phi2,phi3 :
phi2[1]:=(x,t)->Soliton(x,t,a2,x_2);
phi1[2]:=(x,t)->TwoSoliton(x,t,a1,x_1,a2,x_2);
phi2[2]:=(x,t)->TwoSoliton(x,t,a2,x_2,a3,x_3);
phi1[3]:=(x,t)->Backlund(a3,a1,phi2[1](x,t),phi2[2](x,t),
phi1[2](x,t));
RETURN(Re(phi1[3](x,t))) end;
```

where x, t — space and time coordinates, $a1, a2, a3$ — parameters of transformation, x_1, x_2, x_3 — initial positions.

This solution describes a wide spectrum of the three-soliton solutions: Kink-Antikink-Kink, Antikink–Kink–Antikink, Kink–Breather and so on.

Let us demonstrate the derivation and graphical representation of the Kink–Breather collision.

Standing Kink and Moving Breather collision

We input parameters and, with the use of the procedure, build the Kink-Breather solution with the Kink standing ($\nu_K = 0$) at $x = x\_K$ and Breather moving with the frequency ω and velocity ν_B.

```
> v[K]:=0: # velocity of the Kink
> v[B]:=0.5: # velocity of the Breather
> omega:=0.154: # frequency of the Breather
```

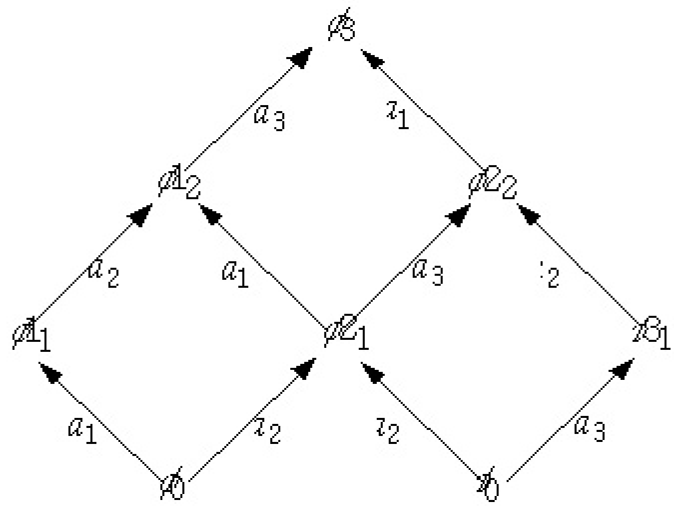

Fig. 9: Diagram 2

```
> R:=sqrt((1-v[B])/(1+v[B])):
> eta:=sqrt(1-omega^2):
> a[1]:=R*(eta+I*omega): # parameters of
> a[2]:=sqrt((1-v[K])/(1+v[K])): # transformation
> a[3]:=R*(eta-I*omega):
> T:=2*Pi/omega:
> L:=T*abs(v[B])/(sqrt(1-v[B]^2)):
> x_K:=L/2: # initial state of the Kink
> x_B:=0: # initial state of the Breather
> K_B:=(x,t)->ThreeSoliton(x,t,a[1],x_B,
a[2],x_K,a[3],x_B):
```

2D animation of the Kink-Breather interaction.

```
> ddx:=-2*arctanh(sqrt((1-omega^2)*(1-v[B]^2))):
> L0:=line([x_K-ddx,-4], [x_K-ddx,10],
color=blue, linestyle=3):
> L1:=line([x_K+ddx,-4], [x_K+ddx,10],
color=black, linestyle=3):
> an_K_B:=animate(K_B(x,t),x=-0.5*L..1.5*L,t=-T..2*T,
```

title="Standing Kink and Moving Breather collision", color=red,numpoints=100):
> **display([an_K_B,L0,L1]);**

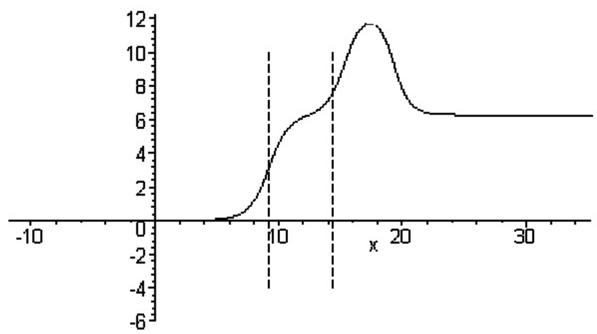

Fig. 10: Standing Kink and Moving Breather collision

The blue line shows the Kink before the collision and the black one after the collision.

Representation of the collision with the use of 3D graphics.

> **plot3d(K_B(x,t),x=-0.5*L..1.5*L,t =-T..2*T,title ="Standing Kink and Moving Breather collision", orientation=[75,35],grid=[40,40]);**

One can rotate the 3D image in order to observe the process from different points of view.

The presented graphics clearly show that the standing Kink after the collision with the moving Breather do not change its shape and velocity but only shifts to a new position with coordinate $\mathcal{X}\_K = x\_K + \Delta_K$. The shift can be found from the formula

$$\Delta_K = -4 \arctan \sqrt{(1-\omega^2)(1-\nu_{B^2})}.$$

Standing Breather and Moving Kink collision

In this section we present the interaction of standing $(\nu_B - 0)$ at $x = x\_b$ Breather with the Kink moving with the velocity ν_K.

> **v[K]:=0.5: # velocity of the Kink**
> **v[B]:=0: # velocity of the Breather**
> **omega:=0.154: # frequency of the Breather**
> **R:=sqrt((1-v[B])/(1+v[B])):**
> **eta:=sqrt(1-omega^2):**

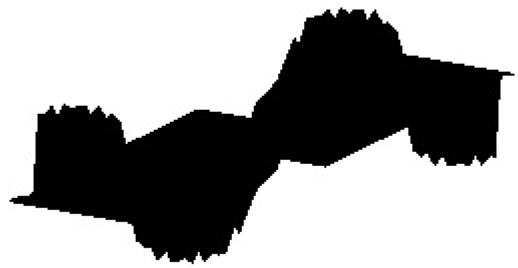

Fig. 11: Standing Kink and Moving Breather collisions

```
> a[1]:=R*(eta+I*omega): # parameters of
> a[2]:=sqrt((1-v[K])/(1+v[K])): # transformation
> a[3]:=R*(eta-I*omega):
> L:=10/sqrt(1-v[K]^2):
> x_K:=-4*sqrt(1-v[K]^2): # initial state of the Kink
> x_B:=L/2: # initial state of the Breather
> T:=(L-2*x_K)/v[K]:
> K_B:=(x,t)->ThreeSoliton(x,t,a[1],x_B,a[2],x_K,
a[3],x_B):
```

2D animation of the Kink–Breather interaction.

```
> ddx:=-arctanh(sqrt((1-omega^2)*(1-v[K]^2)))/
sqrt(1-omega^2):
> L0:=line([x_B-ddx,-6], [x_B-ddx,12],
color=blue, linestyle=3):
> L1:=line([x_B+ddx,-6],[x_B+ddx,12],color=black,
linestyle=3):
> an_B_K:=animate(K_B(x,t),x=-L..3*L,t
=-T..2*T,title=
"Standing Breather Moving Kink collision",color=red):
> display([an_B_K,L0,L1]); >
```

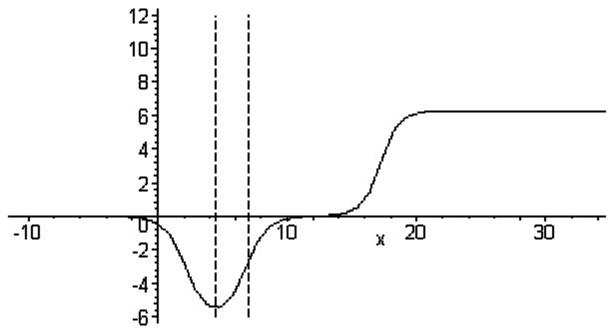

Fig. 12: Standing Breather Moving Kink collision

The blue line shows the breather before the collision and the black one after the collision.

3D representation of the collision.

> plot3d(K_B(x,t),x=-L..2*L,t=-T..2*T,title=
"Standing Breather Moving Kink collision",
orientation=[70,45],grid=[40,40]);

One can rotate the 3D image in order to observe the process from different points of view. The presented graphics clearly show that the velocity and the oscillation frequency of the standing Breather, after the collision with the moving Kink, are unchanged. The position of the Breather shifts after collision to $X\_B = x\_B + \Delta_B$. The shift can be found from the following formula:

$$\Delta_B = \frac{2 \arctan \sqrt{(1-\omega^2)(1-\nu_{K^2})}}{\sqrt{1-\omega^2}},$$

which is a result [4] of the inverse scattering transform analysis.

7. CONCLUSION

In the present worksheet, we demonstrated the power of MAPLE deriving the multi-soliton solutions to the sine-Gordon equation with the use of Backlund transformations. We considered both real and complex parameters of transformations. The solutions are given in the form of recursion procedures, which gives an efficient representation of the „soliton ladder“ idea. Animation and 3D graphics help to visualize the basic properties of soliton collisions.

It is also necessary to mention here that such MAPLE package assisted worksheets can be devised for studying dynamical properties and collisions

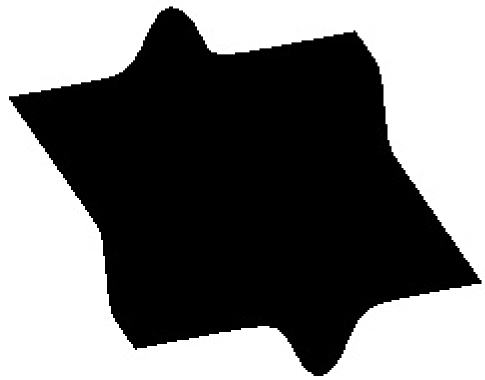

Fig. 13: Standing Breather and Moving Kink collision

of other types of solutions to the SG and many another nonlinear integrable dynamical systems. From this point of view a nontrivial interest presents nowadays the investigation of the dynamical behavior of special loop-wise exact solutions to the Ostrowski–Whitham type [5, 7] integrable dynamical systems on suitable functional manifolds. These aspects of the computer simulation problem within the MAPLE program package will be presented in a next work under preparation.

[1] *Dodd R.K., Eilbeck J.C., Gibbon J.D., Morries H.C.* Solitons and Nonlinear Wave Equations // Academic Press, London, 1982. – 694 p.

[2] *Hentosh O., Prytula M., Prykrapatsky A.* Differential-geometric backgrounds of integrable nonlinear dynamical systems on functional manifolds. – Lviv National University Publisher, The second edition, Lviv, 2006. – 408 p. (in Ukrainian).

[3] *Mitropolski Yu.A., Bogolubov N.N. (jr.), Prykarpatsky A.K., Samoylenko V.G.* Integrability of nonlinear dynamical systems: spectral and differential-geometric aspects. – Naukova Dumka, Kiev, 1987. – 296 p. (in Russian).

[4] *Novikov S.P.* Theory of solitons. The inverse scattering method (Editor). – Moscow: Nauka,1980. – 320 p. (English traslation: Plenum, New York, 1984).

[5] *Parkes J.* Explicit solutions of the reduced Ostrovsky equation // Chaos, Solitons and Fractals. – 2007, V. 31. – P. 602–610.

[6] *Prykarpatsky A., Mykytuyk I.* Algebraic integrability of nonlinear dynamical systems on manifolds: classical and quantum aspects. – Kluwer, 1998. – 553 p.

[7] *Prykarpatsky A., Prytula M.* The gradient-holonomic integrability analysis of a Whitham type nonlinear dynamical model with spacial memory // Nonlinearity. – 2006, 19. – P. 2115–2122.

[8] *Prykarpatsky A., Zagrodzinski J.* Lagrangian and Hamiltonian aspects of Josephson type media // Annales H. Poincare, sec. A. – 1999, v. 70, No. 5. – P. 497–524.

[9] *Prykarpatsky Ya.A.* New approach to studying the vortex structure of Josephson type media equations within the framework of the Chern–Simons–Higgs lagrangian model // Physica C. – 2002, V. 369. – P. 325–330.

[10] *Zagrodzinski J., Prykarpatsky A.* Dynamics of excitations in Josephson media // Physics C, 2002. – V. 332. – P. 313–319.

КОМП'ЮТЕРНИЙ АНАЛІЗ ТОЧНИХ РОЗВ'ЯЗКІВ РІВНЯННЯ СИНУС-ГОРДОНА ТА СОЛІТОННІ ЗІТКНЕННЯ

Микола ПРИТУЛА [1], *Ярема ПРИКАРПАТСЬКИЙ* [2,3,4], *Микола СТАРЧАК* [1]

[1] Львівський національний університет імені Івана Франка,
вул. Університетська, 1, Львів 79000, Україна
[2] Інститут математики НАН України,
вул. Терещенківська, 3, Київ 01601, Україна
[3] Дрогобицький державний педагогічний університет ім.І.Франка,
вул. І. Франка, 24, Дрогобич 82100, Україна
[4] Педагогічна академія, Краків 30062 Польща

Стаття присвячена демонстрації програмного PC-продукту на основі пакету MAPLE для аналізу динамічних властивостей багатосолітонних та солітоноподібних точних розв'язків нелінійних інтегровних гамільтонових систем на функціональних многовидах. На прикладі нелінійної цілком інтегровної гамільтонової системи синус-Гордона досліджено двовимірну анімацію та тривимірну графіку солітонних зіткнень.