

Effects of Higher Order Dispersion Terms in the Nonlinear Schrodinger Equation

Robert Beech and Frederick Osman
 University of Western Sydney, Penrith-South, 1791, Australia

Abstract: This study presents a concise graphical analysis of solitonic solutions to a nonlinear Schrodinger equation (NLSE). A sequence of code using the standard NDSolve function has been developed in Mathematica to investigate the acceptable accuracy of the NLSE in relatively small ranges of the dispersive parameter space. An operator splitting approach was used in the numerical solutions to expand the boundaries and reduce the artifacts for a reliable solution. These numerical routines were implemented through the use with Mathematica and the results give a very clear view of this interesting and important practical phenomenon.

Key words: Solitons, Solitonic solutions, Nonlinear Schrodinger equation, Numerical artifacts

INTRODUCTION

The nonlinear Schrödinger equation (NLSE):

$$i \frac{\partial u}{\partial \xi} + \frac{1}{2} \frac{\partial^2 u}{\partial \tau^2} + \text{Abs}[u^2]u = -i\Gamma u + i\alpha \frac{\partial^3 u}{\partial \tau^3} \quad (1)$$

May be used to model the propagation of pulsed light in optical fibres^[1,2]. When $\Gamma = \alpha = 0$ this becomes the standard model of picosecond pulses^[3]. $u(\xi, \tau)$ is a complex-valued function modelling the envelope of the effective electric field as a function of distance ξ along the fiber and time τ in a frame moving at the group velocity. The left-hand side of Equation (1) models the effects of group velocity dispersion and self-phase modulation while the right-hand side accounts for the refractive nature of the fiber (with refractive index Γ) and higher order linear dispersion effects (with coefficient α). As an initial attempt to understand the contribution of the dispersion term, results for non-zero α (and $\Gamma = 0$) will be compared to the well-known soliton solution of the standard model. Defining the NLSE operator as:

$$\text{nlse}[\alpha, n][u_] := i\partial_{\xi}u + 1/2\partial_{(\tau,2)}u + \text{Abs}[u]^2u - i\alpha\partial_{(\tau,3)}u$$

the fundamental Zakharov-Shabat soliton solution of the standard model^[4] can be verified:

$$u(\xi, \tau) = \text{Sech}(\tau)e^{i\xi/2} \quad (2)$$

Simplify[nlse[0,0][Sech[τ]Exp[$i \xi /2$]]==0,{ ξ, τ } \in Reals]: The standard methods available with Mathematica's **NDSolve** function did not provide sufficient accuracy for our analysis. Rob Knapp has kindly provided a package

$$u[0, \tau] = \text{Sech}[\tau] \quad (3)$$

based on a split step method used in this study. The NLSE with n^{th} order dispersion is solved numerically with initial condition and asymmetric boundary condition:

$$u(\xi, -\tau_{\text{lim}}) = u(\xi, \tau_{\text{lim}}) \quad (4)$$

which models a train of interacting pulses.

Function definitions: The standard **NDSolve** method of **StiffnessSwitching** is implemented in the following function:

```
nsolStSw[ $\alpha$ , n,  $\xi_{\text{lim}}$ ,  $\tau_{\text{lim}}$ ] := nsolStSw[ $\alpha$ , n,  $\xi_{\text{lim}}$ ,  $\tau_{\text{lim}}$ ] =
NDSolve[{nlse[ $\alpha$ , n][u[ $\xi$ ,  $\tau$ ]] == 0, u[0,  $\tau$ ] == Sech[ $\tau$ ], u[ $\xi$ , - $\tau_{\text{lim}}$ ] == u[ $\xi$ ,  $\tau_{\text{lim}}$ ]},
u[ $\xi$ ,  $\tau$ ], { $\tau$ , - $\tau_{\text{lim}}$ ,  $\tau_{\text{lim}}$ }, { $\xi$ , 0,  $\xi_{\text{lim}}$ }, Method -> StiffnessSwitching,
MaxSteps -> Infinity, PrecisionGoal -> 2][[1]]
```

The non-standard **Split Step** method is implemented in the following function:

Corresponding Author: Robert Beech, University of Western Sydney, Penrith-South, 1791, Australia

```

nsolSpSt[α_, n_, ξlim_, τlim_] := nsolSpSt[α, n, ξlim, τlim] =
  NDSolve[{nlse[α, n][u[ξ, τ]] == 0, u[0, τ] == Sech[τ], u[ξ, -τlim] == u[ξ, τlim]},
    u[ξ, τ], {τ, -τlim, τlim}, {ξ, 0, ξlim}, Method -> {"MethodOfLines",
      "SpatialDiscretization" -> {"TensorProductGrid",
        "DifferenceOrder" -> "Pseudospectral", "MinPoints" -> mp, "MaxPoints" -> mp},
      Method -> {DoubleStep, Method -> {SplitStep, "DifferenceOrder" -> 4},
        StiffnessTest -> False}}, PrecisionGoal -> 4, AccuracyGoal -> 4][[1]]

mp = 2049;

Off[General::"spell1"]

```

The following functions produce amplitude, contour and τ and ξ cross-section plots for the NLSE solution:

```

ampliPlot1[α_, n_, ξlim_, τlim_, method_] :=
  Plot3D[Abs[u[ξ, τ] /. method[α, n, ξlim, τlim]], {ξ, 0, ξlim},
    {τ, -τlim, τlim}, PlotRange -> All, PlotPoints -> 100, Mesh -> False, Boxed -> False,
    ViewPoint -> {-2, -1.5, 1.5}, AxesLabel -> {ξ, τ, "|u(ξ, τ)|"},
    Ticks -> {{ξlim/2, ξlim}, {-2 τlim/3, 0, 2 τlim/3}, {0, 1}},
    PlotLabel -> "a. Forward amplitude plot"]

ampliPlot2[α_, n_, ξlim_, τlim_, method_] :=
  Plot3D[Abs[u[ξ, τ] /. method[α, n, ξlim, τlim]], {ξ, 0, ξlim},
    {τ, -τlim, τlim}, PlotRange -> All, PlotPoints -> 100, Mesh -> False, Boxed -> False,
    ViewPoint -> {2, 1.5, 1.5}, AxesLabel -> {ξ, τ, "|u(ξ, τ)|"},
    Ticks -> {{ξlim/2, ξlim}, {-2 τlim/3, 0, 2 τlim/3}, {0, 1}},
    PlotLabel -> "b. Reverse amplitude plot"]

contrPlot1[α_, n_, ξlim_, τlim_, method_] :=
  ContourPlot[Abs[u[ξ, τ] /. method[α, n, ξlim, τlim]], {ξ, 0, ξlim},
    {τ, -τlim, τlim}, PlotRange -> All, PlotPoints -> 100, Frame -> True,
    ContourShading -> False, FrameLabel -> {ξ, τ}, RotateLabel -> False, AxesLabel -> {ξ, τ},
    FrameTicks -> {{0, ξlim/2, ξlim}, {-2 τlim/3, 0, 2 τlim/3}, {}, {}},
    PlotLabel -> "c. Contour plot"]

crossPlot1[α_, n_, ξlim_, τlim_, method_] :=
  Plot[{Sech[τ], Abs[(u[ξ, τ] /. method[α, n, ξlim, τlim]) /. ξ -> ξlim]}, {τ, -τlim, τlim},
    PlotRange -> All, PlotPoints -> 100, PlotStyle -> Table[Hue[i/3], {i, 0, 1}],
    AxesLabel -> {τ, "|u(ξ, τ)|"}, AxesOrigin -> {-15, 0},
    Ticks -> {{-2 τlim/3, 0, 2 τlim/3}, {0, 0.5, 1}}, PlotLabel -> "d. τ cross-section plot"]

crossPlot2[α_, n_, ξlim_, τlim_, method_] :=
  Plot[{1, Abs[(u[ξ, τ] /. method[α, n, ξlim, τlim]) /. τ -> 0]},
    {ξ, 0, ξlim}, PlotRange -> {0, 1}, PlotPoints -> 100,
    PlotStyle -> Table[Hue[i/3], {i, 0, 1}], AxesLabel -> {ξ, "|u(ξ, τ)|"},
    Ticks -> {{0, ξlim/2, ξlim}, {0, 0.5, 1}}, PlotLabel -> "e. ξ cross-section plot"]

```

This function displays the plots:

```
<< Graphics`Graphics`
showPlots[α_, n_, ξlim_, τlim_] :=
{Print["Pulse evolution for NLSE with α = ", α, ", n = ", n,
" (Left column: SplitStep method, Right column: StiffnessSwitching method)"];
DisplayTogetherArray[
{{ampliPlot1[α, n, ξlim, τlim, nsolSpSt], ampliPlot1[α, n, ξlim, τlim, nsolStSw]},
{ampliPlot2[α, n, ξlim, τlim, nsolSpSt], ampliPlot2[α, n, ξlim, τlim, nsolStSw]},
{contrPlot1[α, n, ξlim, τlim, nsolSpSt], contrPlot1[α, n, ξlim, τlim, nsolStSw]},
{crossPlot1[α, n, ξlim, τlim, nsolSpSt], crossPlot1[α, n, ξlim, τlim, nsolStSw]},
{crossPlot2[α, n, ξlim, τlim, nsolSpSt],
crossPlot2[α, n, ξlim, τlim, nsolStSw]}}, ImageSize → 600}]
```

These functions produce a series of τ cross-section plots (using the **Split Step** method) for animation:

```
crossPlot3[α_, n_, ξlim_, τlim_, ξn_] :=
Plot[Abs[(u[ξ, τ] /. nsolSpSt[α, n, ξlim, τlim]) /. ξ → ξn], {τ, -τlim, τlim},
PlotRange → {0, 1}, PlotPoints → 50, PlotStyle → Hue[1/3], AxesLabel → {τ, "|u(ξ, τ)|"},
AxesOrigin → {-15, 0}, Ticks → {{-2 τlim/3, 0, 2 τlim/3}, {0, 0.5, 1}}, ImageSize → 300]
```

```
animatecrossPlot[α_, n_, ξlim_, τlim_] :=
Table[crossPlot3[α, n, ξlim, τlim, ξn], {ξn, 0, 2, 0.1}];
```

```
On[General::"spell1"]
```

Results of Split Step and StiffnessSwitching cases plotted side by side: The following plots explore the effect of order n linear dispersion. The dispersion coefficient α is chosen to produce representative examples behavior e observed behavior. On the left side of the graphic columns are the **Split Step** method graphs. The plots on the right show the results for the **StiffnessSwitching** method. In each case, plots a. and b. Illustrate the evolution of the soliton for $0 \leq \xi \leq \xi_{lim}$, plot (a) showing the forward amplitude and plot (b) the reverse. Plot (c) shows the associated contour plot. Plot (d) shows two cross-sectional profiles along the τ axis, the direction of motion of the wave, and compares the initial pulse at $\xi = 0$ (red) with the pulse obtained at $\xi = \xi_{lim}$ (green). Finally, plot (e) gives the corresponding cross-section along the ξ axis and indicates the deviation along the crest of the wave of the final pulse compared to the initial pulse.

For the **Split Step** method **PrecisionGoal** was set at 4 to achieve suitable accuracy for $n > 0$. The maximum achievable consistent value of **PrecisionGoal** was 2 for the **StiffnessSwitching** method.

* **No dispersion**

Zero order: Soliton solution remains unchanged. No apparent difference between methods.

* **Higher order dispersion**

First order: Soliton solution is retarded. **StiffnessSwitching** method apparently introduces numerical artifacts.

Second order: Soliton solution disperses symmetrically.

Third order, low α : Nonsymmetric radiation produced, possible numerical artifacts appearing as transverse waves originating on τ boundaries.

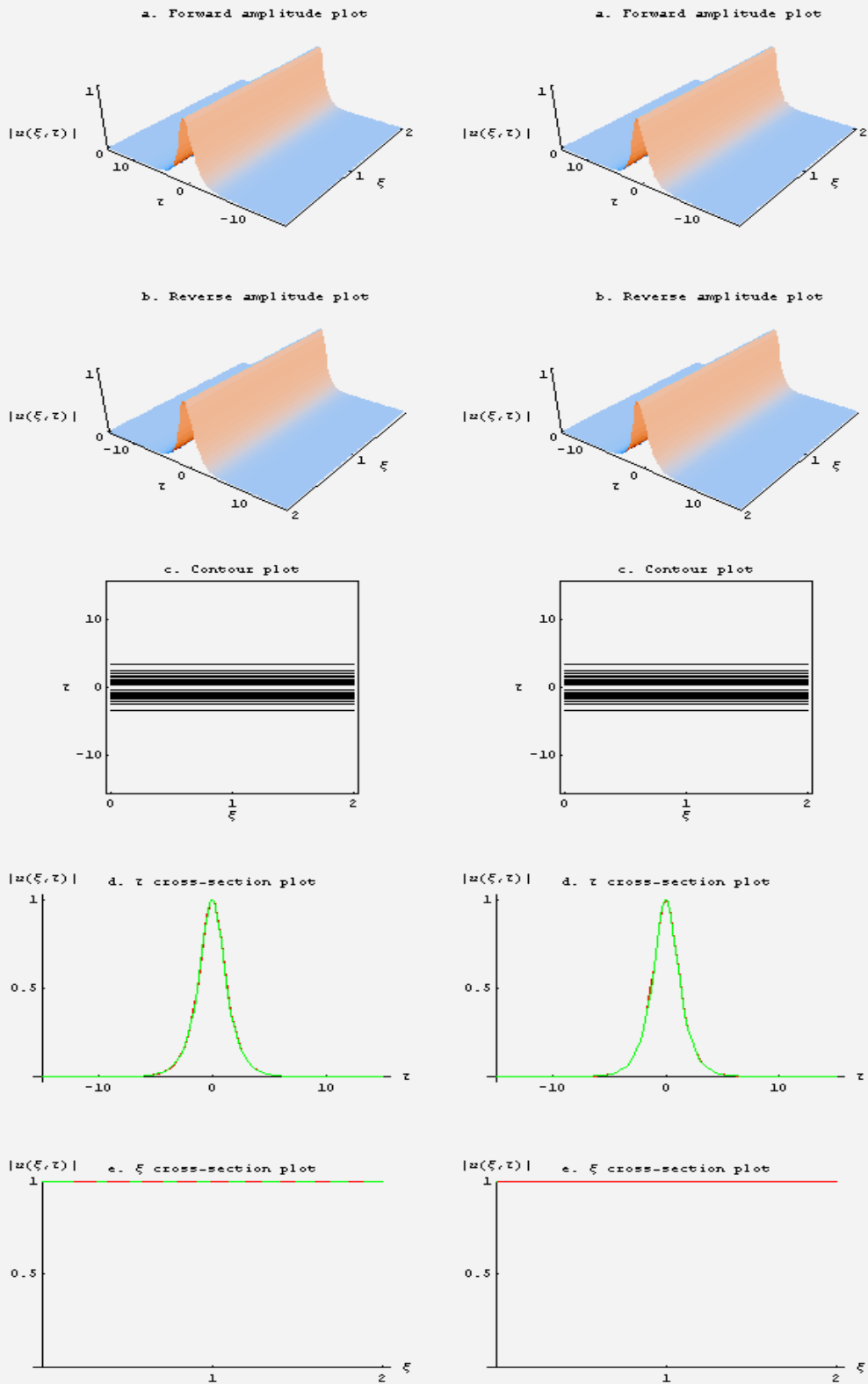
Third order, higher α : Nonsymmetric radiation increased and feeding into a pulse.

Fourth order: Symmetric dispersion, possible small radiation or numerical artifacts.

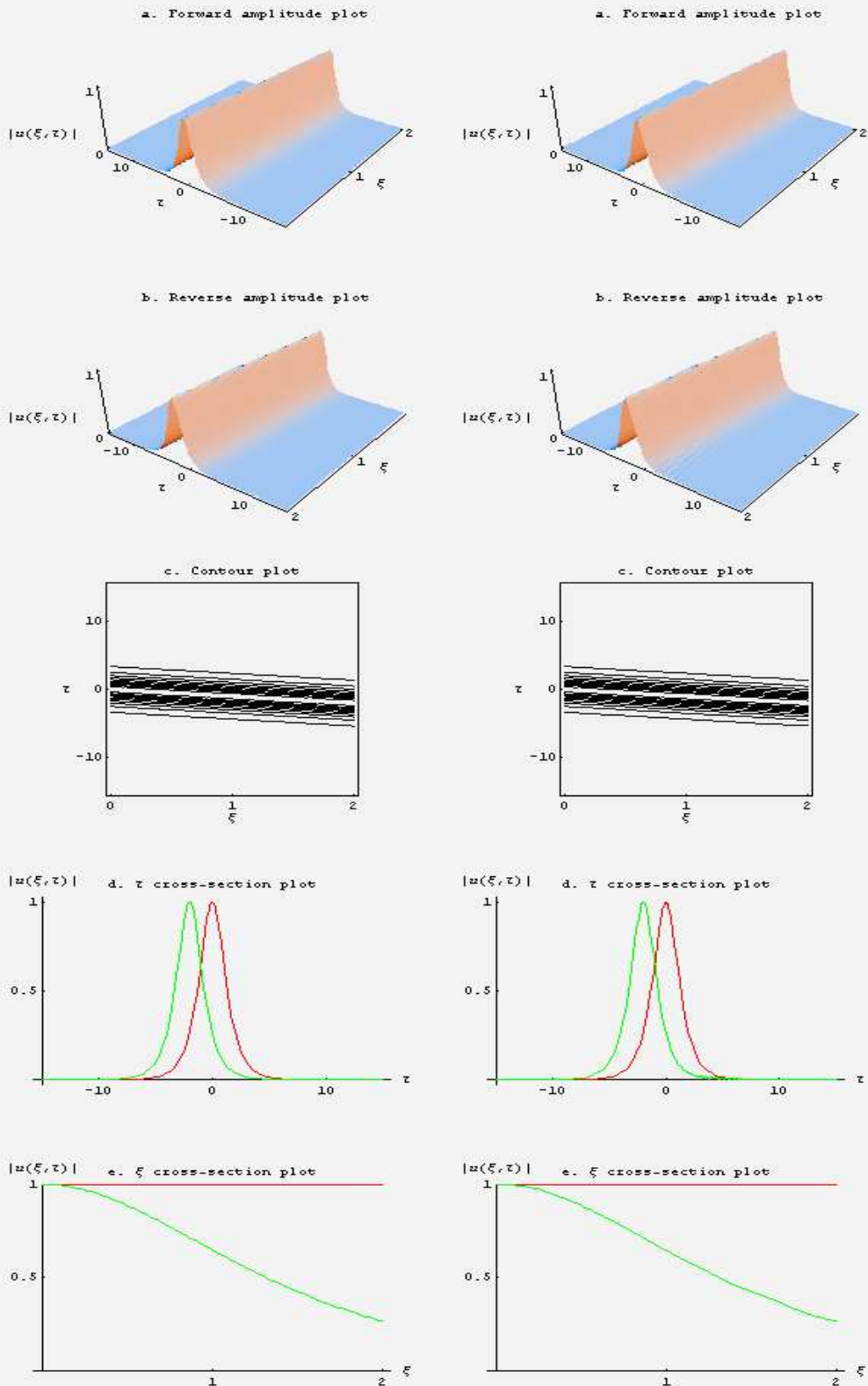
Fifth order: Increasing α from 0.005 to 0.006 initiates nonconvergence of **StiffnessSwitching** method before non-symmetric radiation becomes pronounced.

Sixth order: Increasing α past 0.009 initiates non-convergence of numerical method, numerical anomaly evident in the plot.

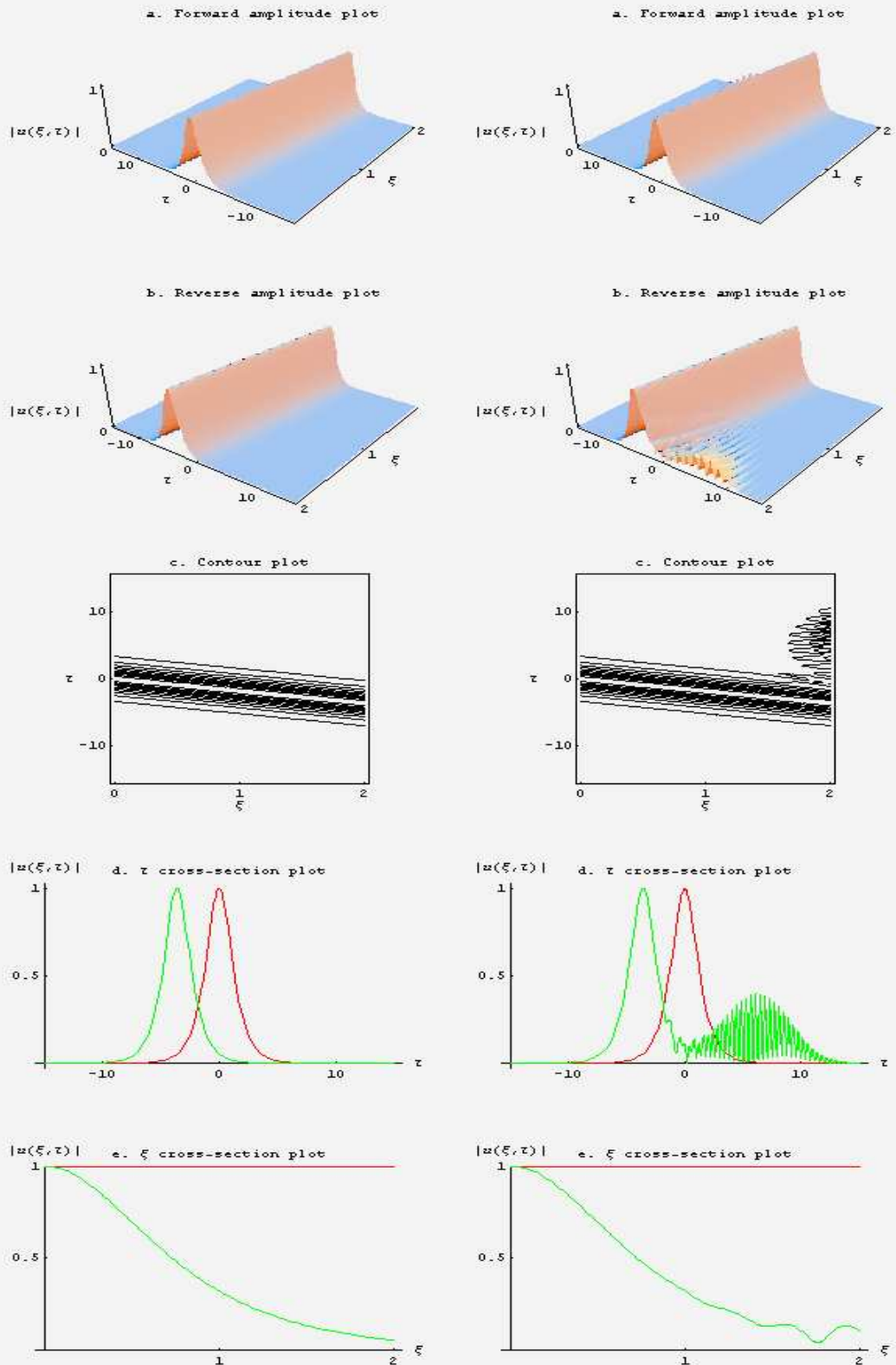
Pulse evolution for MLSE with $\alpha = 0$, $\kappa = 0$
 (Left column: SplitStep method, Right column: StiffnessSwitching method)



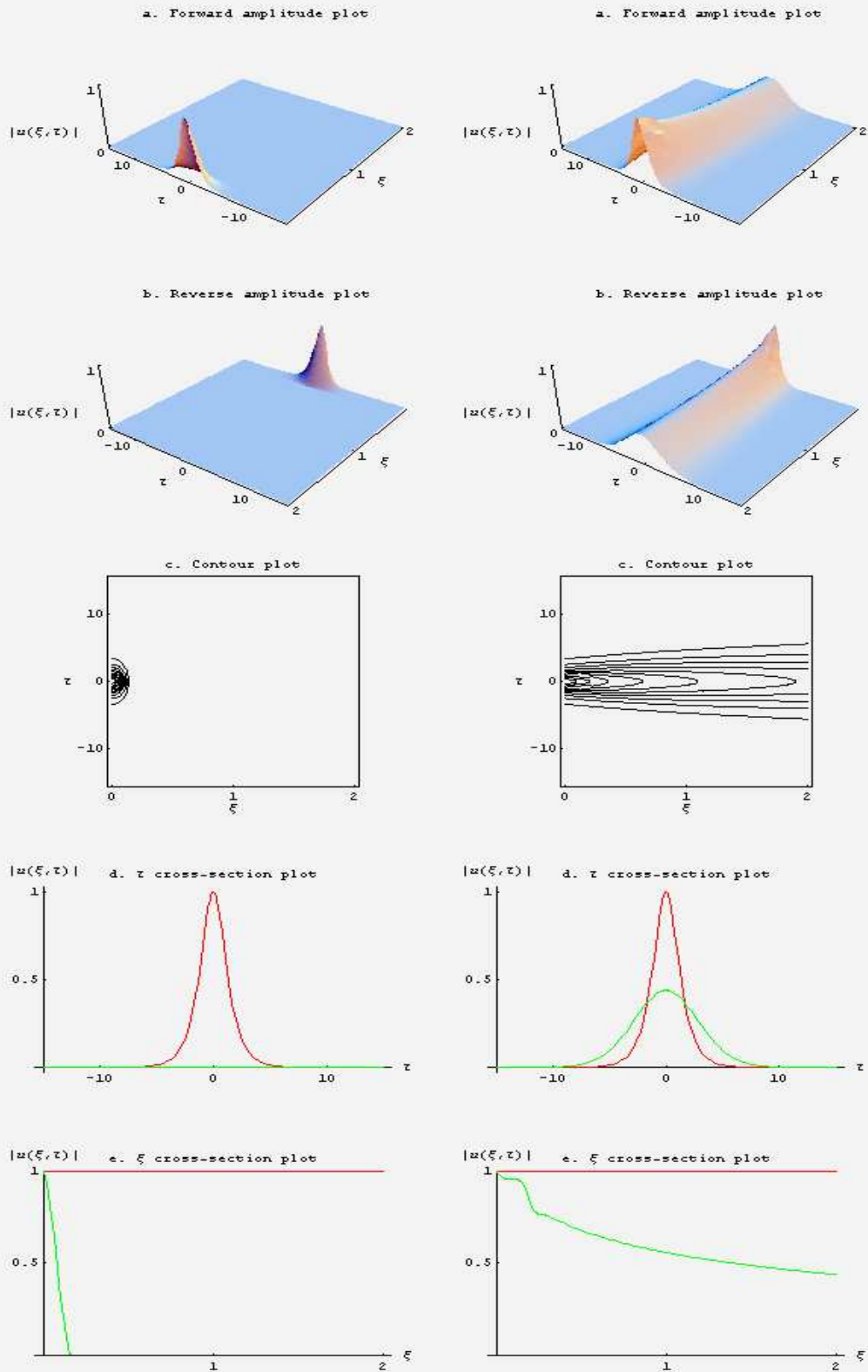
Pulse evolution for MLSE with $\alpha = 1$, $\kappa = 1$
 (Left column: SplitStep method, Right column: StiffnessSwitching method)



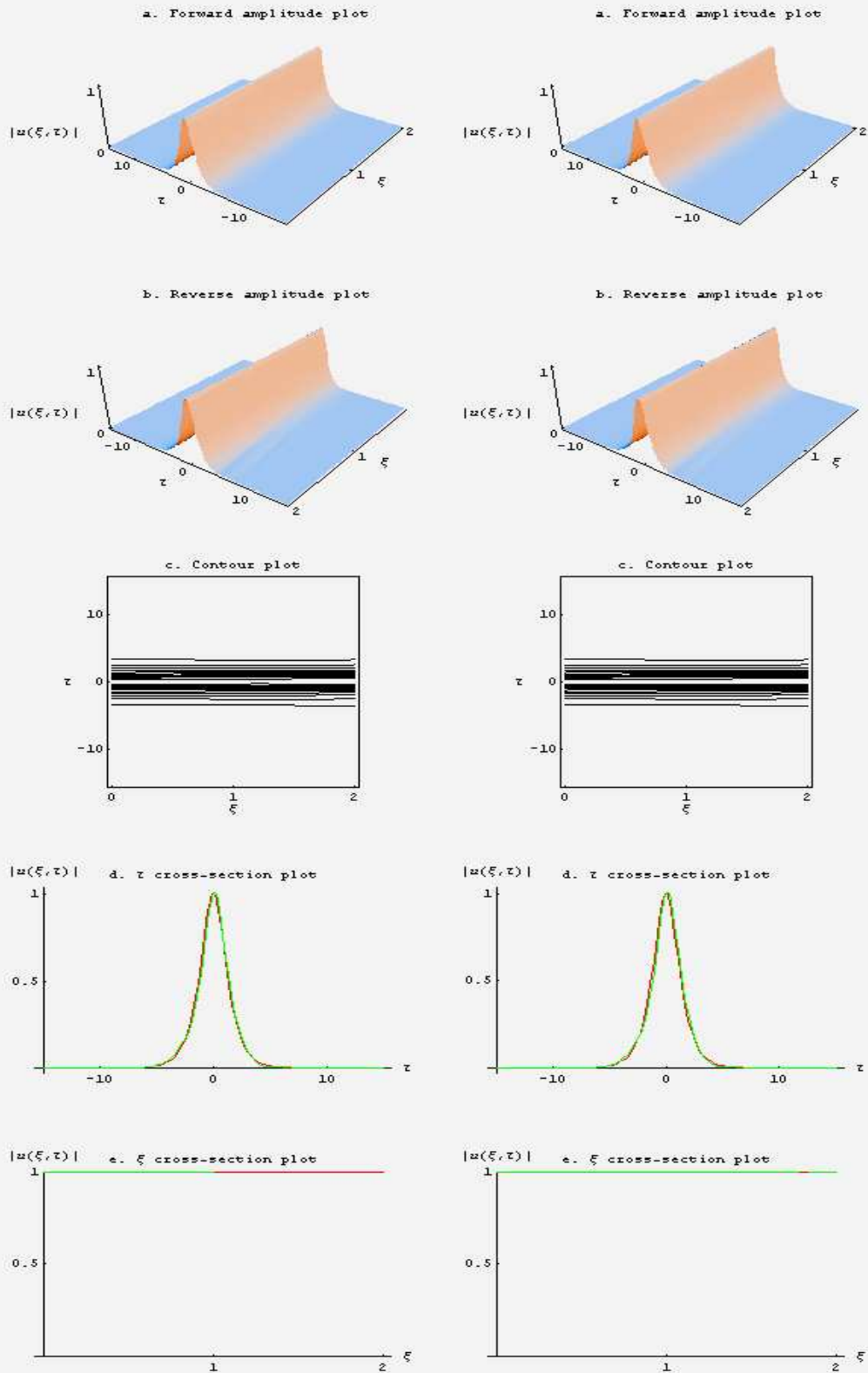
Pulse evolution for MLSE with $\alpha = 1.8$, $n = 1$
 (Left column: SplitStep method, Right column: StiffnessSwitching method)



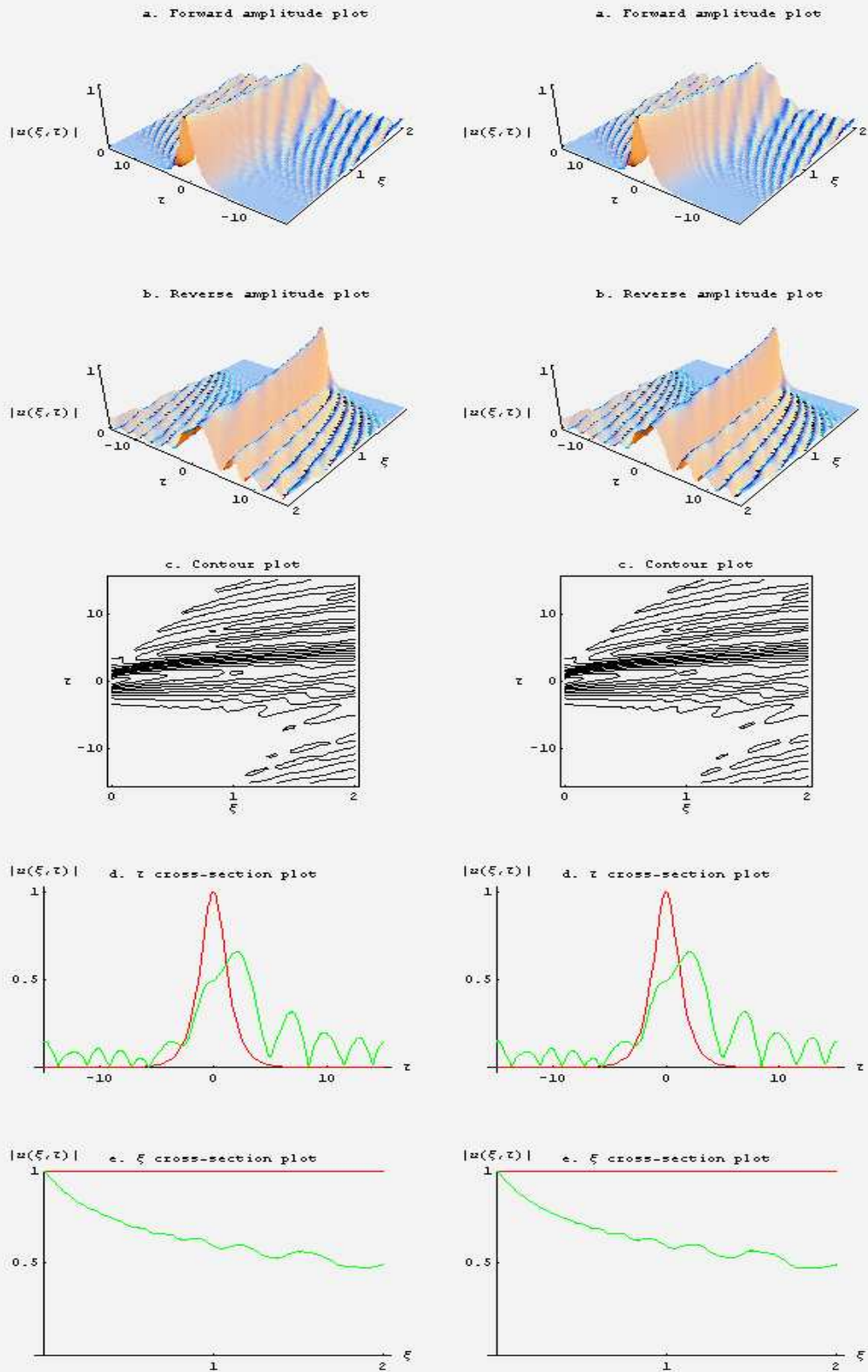
Pulse evolution for MLSE with $\alpha = 1.5$, $\kappa = 2$
 (Left column: SplitStep method, Right column: StiffnessSwitching method)



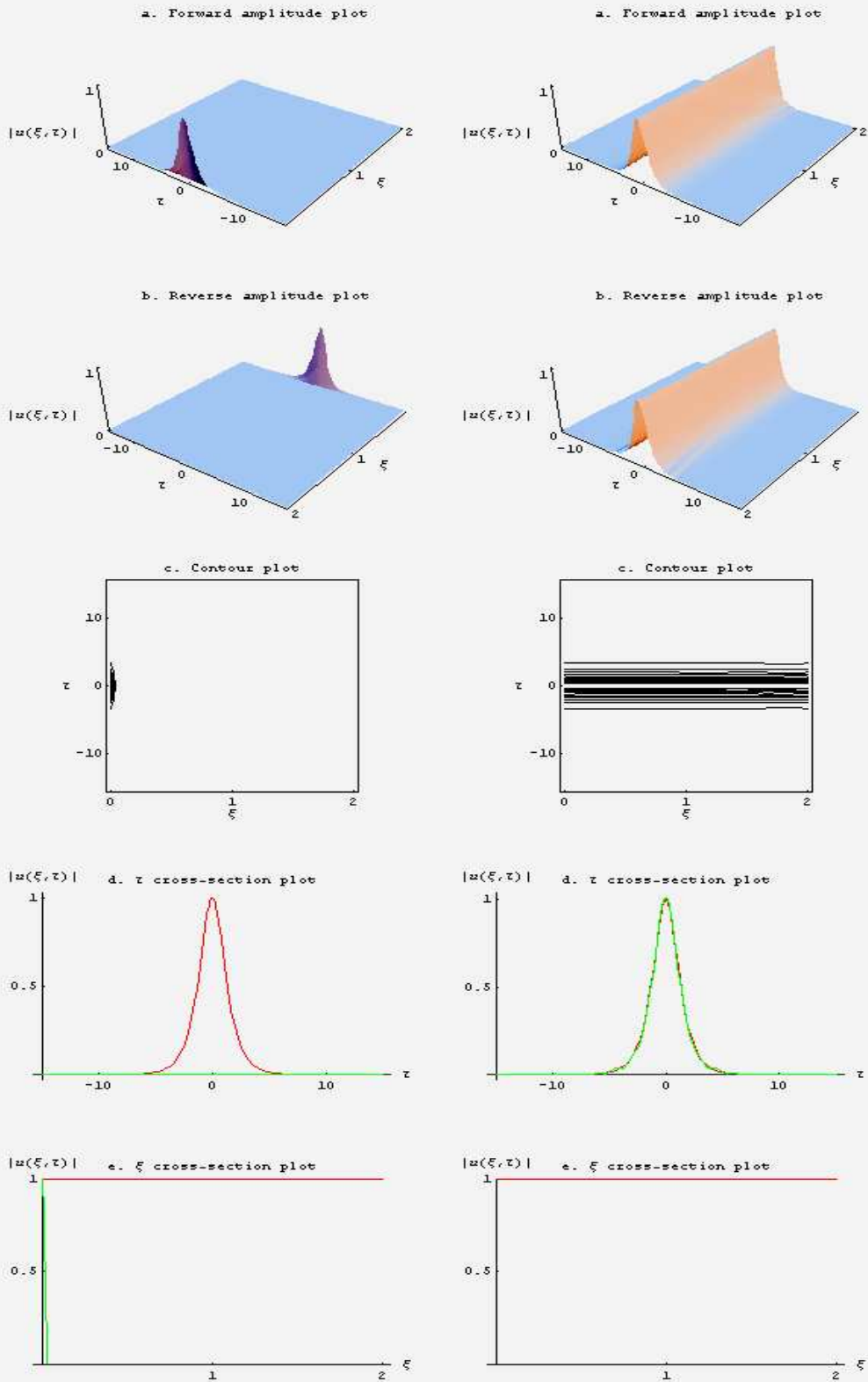
Pulse evolution for MLSE with $\alpha = 0.05$, $n = 2$
 (Left column: SplitStep method, Right column: StiffnessSwitching method)



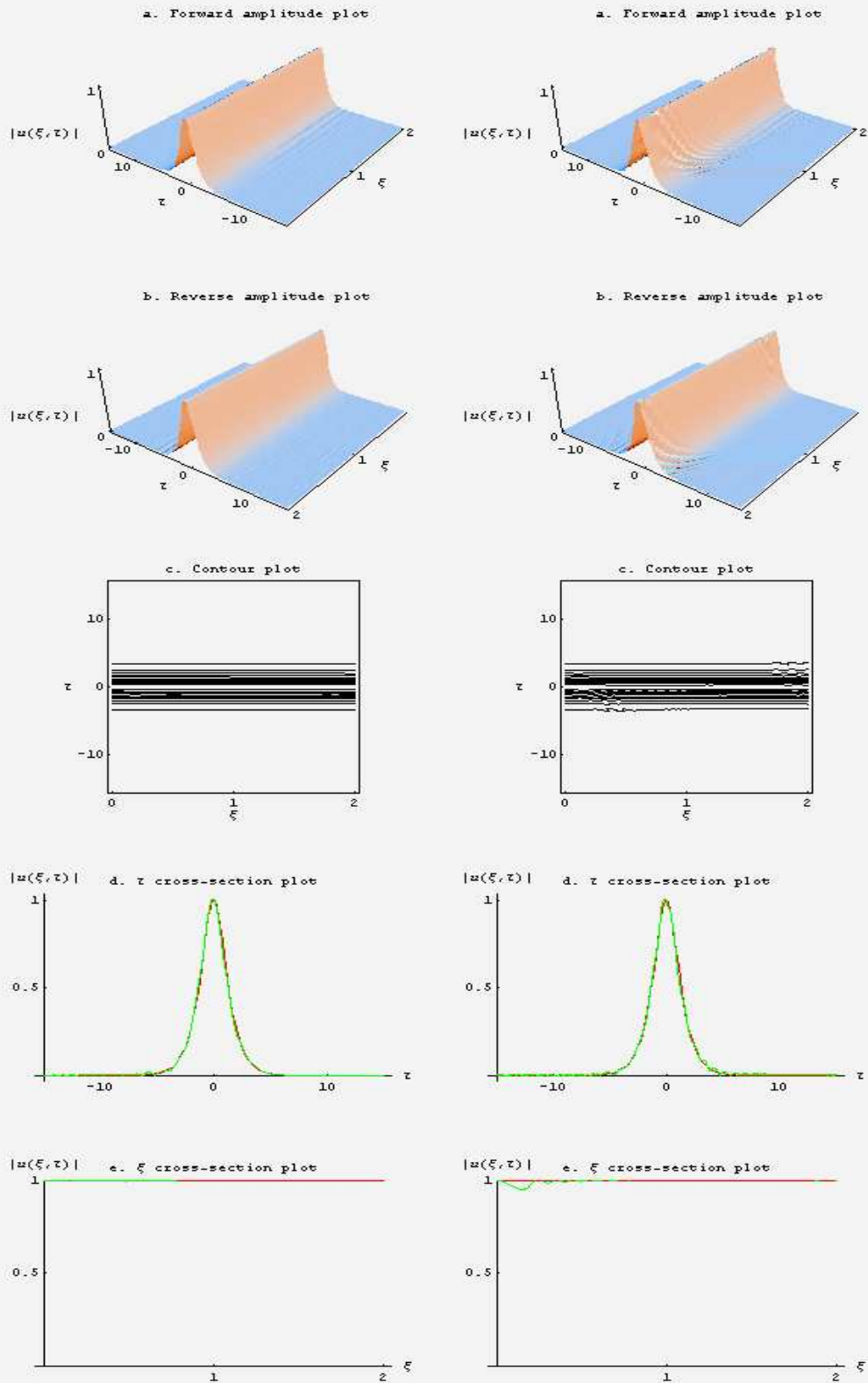
Pulse evolution for MLSE with $\alpha = 1.5$, $\kappa = 2$
 (Left column: SplitStep method, Right column: StiffnessSwitching method)



Pulse evolution for MLSE with $\alpha = 0.007$, $n = 4$
 (Left column: SplitStep method, Right column: StiffnessSwitching method)



Pulse evolution for MLSE with $\alpha = 0.005$, $n = 5$
 (Left column: SplitStep method, Right column: StiffnessSwitching method)

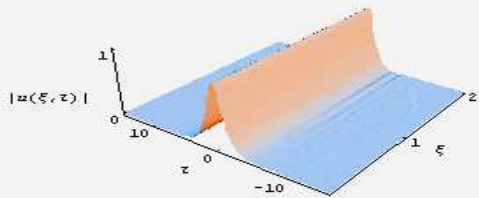


Pulse evolution for MLSE with $\alpha = 0.006$, $n = 5$
 (Left column: SplitStep method, Right column: StiffnessSwitching method)

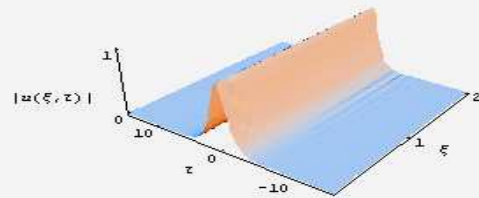
AMDSolve::err :

Warning: Scaled local spatial error estimate of $31.602453904661186^{\wedge} 2$ at $\xi = 2.$ in the direction of independent variable τ is much greater than prescribed error tolerance. Grid spacing with 109 points may be too large to achieve the desired accuracy or precision. A singularity may have formed or you may want to specify a smaller grid spacing using the MaxStepSize or MinPoints options. [More...](#)

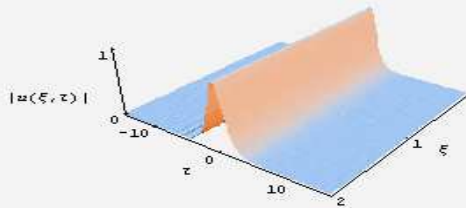
a. Forward amplitude plot



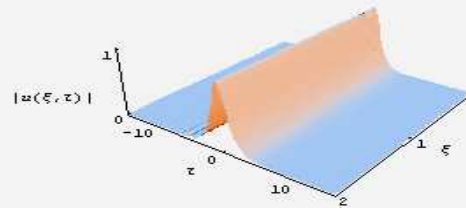
a. Forward amplitude plot



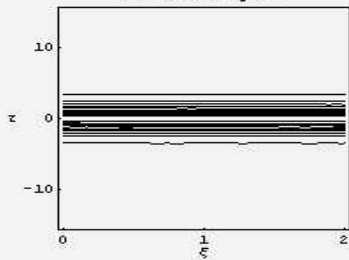
b. Reverse amplitude plot



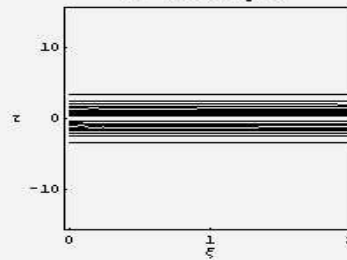
b. Reverse amplitude plot



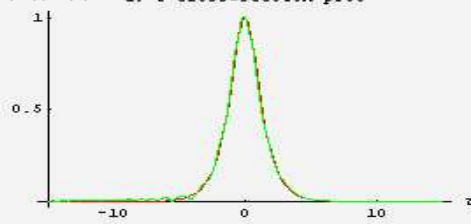
c. Contour plot



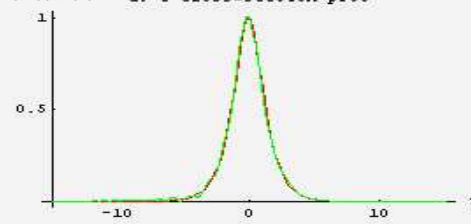
c. Contour plot



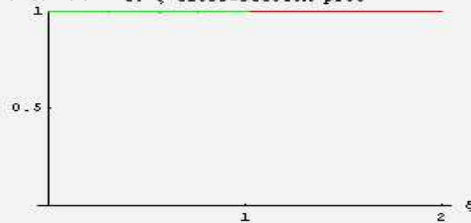
d. τ cross-section plot



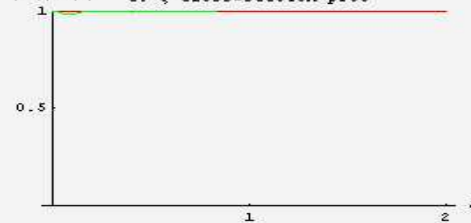
d. τ cross-section plot



e. ξ cross-section plot



e. ξ cross-section plot

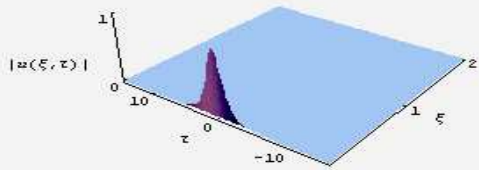


Pulse evolution for MLSE with $\alpha = 0.0009$, $n = 6$
 (Left column: SplitStep method, Right column: StiffnessSwitching method)

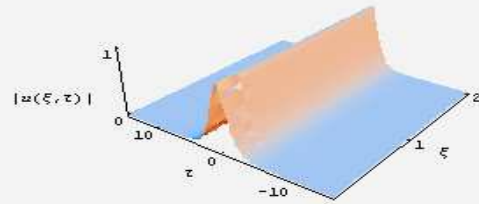
&General::ovfl : Overflow occurred in computation. [More...](#)

&General::stop : Further output of General::ovfl will be suppressed during this calculation. [More...](#)

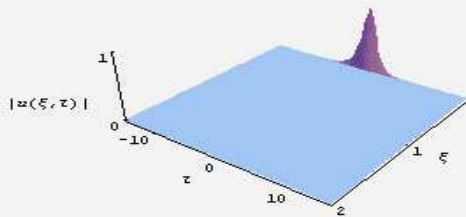
a. Forward amplitude plot



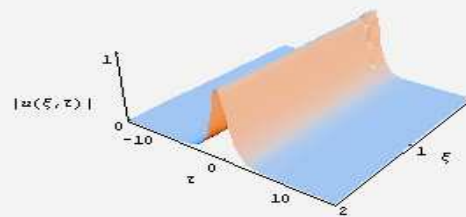
a. Forward amplitude plot



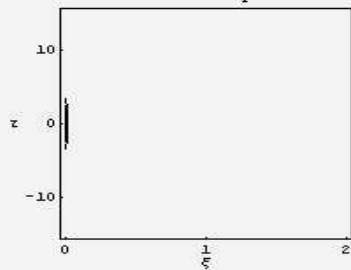
b. Reverse amplitude plot



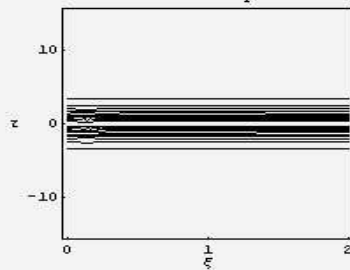
b. Reverse amplitude plot



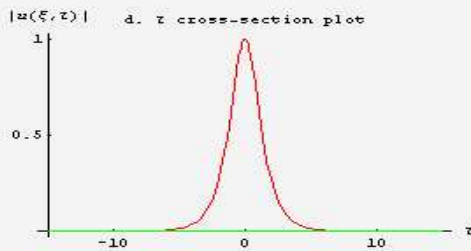
c. Contour plot



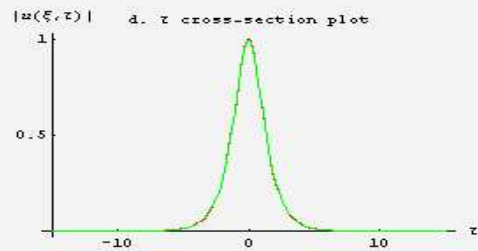
c. Contour plot



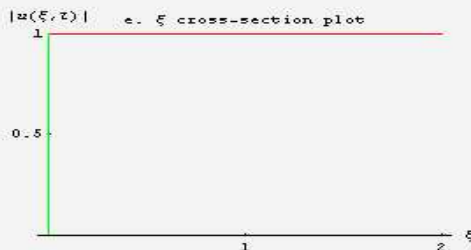
d. τ cross-section plot



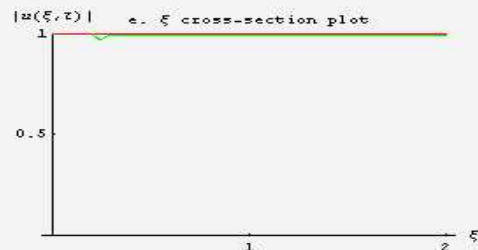
d. τ cross-section plot

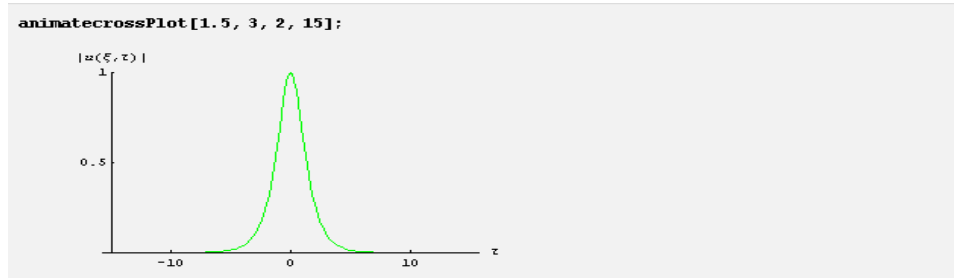


e. ξ cross-section plot



e. ξ cross-section plot





DISCUSSION

In the first set of plots; No dispersion, zero order, as was to be expected from the definition of a pure soliton wave, there was also no noticeable difference in the wave plots produced by the two methods under review here.

In the second case, part 1; Higher order dispersion, first order, coefficient (coef.) $\alpha = 1$, we get another soliton solution. This time the wave is retarded, as exhibited by its angle to the τ axis in the amplitude and contour plots and its separation in the first cross-section. The second cross-section shows the extent of the retardation. In part 2; coef. = 1.8, the previous observations are more marked, and here is observed an artifact in the **StiffnessSwitching** (St.Sw.) plots.

In the second order plots, coef. = 1.5 the soliton solution still disperses symmetrically, and as predicted for an even order, shows less evidence of other activity, especially in **SplitStep**. However the accuracy of both methods now comes into question. Most other resemblance to a soliton has gone. The Split Step case has been severely retarded, as has the St.Sw., which is again showing the presence of numerical artefacts.

In the graphs representing the third order, the lower coef. = 0.05 now shows the appearance of non-symmetric radiations, in both methods, though minor at this coef., where retardation seems less evident. Moving up to the coef. = 1.5 these effects become much more evident. Differences in the two methods are now evident, as is retarded. The radiation is seen to be feeding into the pulse and the question is now apparent as to whether this is due to nothing more than numerical artifacts.

Next is presented a short animation of the τ cross-section, which may be of some interest here. Coming now to the fourth order, the expectedly more peaceful background is again evident, though a small radiation, or artifact, may be seen. The resemblance to the classical definition of a soliton is, however markedly absent. In the **Split Step** the retardation is very severe and again raises serious doubts regarding the accuracy of this method.

For the fifth order non-convergence of the St.Sw. Method starts between coef. = 0.005 and 0.006. Up to this level radiation is minor and retardation has not set in. The evident differences in the form of this "radiation" in the two methods again reinforce the argument for the belief that it is an artifact.

For the sixth order, any attempt at increasing the coef. Past 0.0009 initiates non-convergence. Up to this point however the same inconsistencies are clearly emerging, again bringing into question the reliability of either method.

CONCLUSION

There is no evidence here in the results produced using either the traditional **StiffnessSwitching** Method, or the newer **Split Step** method which would reinforce the premise that secondary radiation in the soliton wave exists in the quadratic range and not in the cubicle. The results produced here would even give ground to the assertion that secondary radiations do not occur at all!

The only conclusion that we can reach from this is that the methods provided in Mathematica to assist in the use of the NDSolve command are full of flaws, especially in respect to the presence, or the creation of numerical artifacts. From what we have observed with the **Split Step** method we can safely say that, while there are minor improvements, it does not instill sufficient confidence in the user that he might conclude the absence of artifacts here either.

ACKNOWLEDGEMENT

The authors acknowledge the assistance given to them in the production of this work by Dr Rob Knapp, Wolfram Research, for the use of his Mathematica package "**Split Step. m**". The authors also acknowledge Dr Paul Abbott, UWA for their invaluable input, which improved the results of this study. Associate Professor Richard Ollerton for his invaluable assistance and advice, with particular reference to the Mathematica coding used in this study.

REFERENCES

1. Hasegawa, A. and F. Tappert, 1973. Appl. Phys. Lett., 23: 142.
2. Herman, R.L., 1992. American Scientist, Jul.-Aug.
3. Doktorov, E.V., 2002. The modified nonlinear Schrodinger Equation: Facts and artifacts. Eur. Phys. J., 20: 227-231.
4. Zakharov and Shabat, 1972. Exact theory of two-dimensional self-focusing and one-dimensional self-modulation of waves in nonlinear media. Soviet Physics JETP, 34: 62-69.