P R L

TECHNICAL NOTE

TN-81-21

TN-81-021

NLPDE

A program for integrating nonlinear partial differential equations

By

Y.C.Saxena & A. Sen

1981

PHYSICAL RESEARCH LABORATORY
AHMEDABAD-9

ABSTRACT

We describe a computer program to solve nonlinear partial differential equations of the type $\frac{\partial F}{\partial \tau} = N(F) + D(F)$ where N is some nonlinear operator and D stands for a linear dispersive operator. The numerical algorithm uses the partially corrected Adams-Bashforth method for time integration and Fast Fourier Transform (FFT) routines for space integration. The KdV equation is solved as an illustration.

INTRODUCTION

The purpose of t is note is to provide a detailed description of a computer program that has been developed to numerically integrate nonlinear partial differential equations of the type

$$\frac{\partial F}{\partial \tau} = N(F) + D(F) \qquad (1)$$

where N is some nonlinear operator and D stands for a linear dispersive operator. Many wave propagation problems in plasma physics as well as in other systems yield such equations. Some familiar examples are the Korteveg de Vries (KdV) equation for ion acoustic or magnetohydrodynamic waves in plasmas, the Nonlinear Schrodinger (NLS) equation for high frequency Langmuir waves, the complex modified KdV equation for lower hybrid waves etc. Several numerical schemes are now available for the solution of these equations. Some of them are based on the finite-difference method (Zabusky & Kruskal 1965; Greig and Morris 1976) which reduces the partial differential equation to an appropriate set of algebraic equations. Alternatively Fourier expansion methods have also been employed (Schamel & Elsasser 1976; Watanabe et al 1977) which take advantage of the extremely efficient Fast Fourier Transform (FFT) techniques. Some hybrid methods have also been proposed (Gazdag 1973; Tappert 1974; Canosa and Gazdag 1977).

The present program is based on one such hybrid method where the time integration is done by using the partially corrected Adams-Bashforth scheme (Gazdag 1976) and the space operator is split up into a linear and nonlinear part (Tappert 1974). The space integration is mainly carried out in the Fourier space but the nonlinear terms are computed by getting back into real space representation. To prevent aliasing when computing the nonlinear term only the n/2 lowest Fourier modes are retained, where n is the number of grid points in the space coordinate. The accuracy of the integration is gauged by keeping a check on how well the constants of motion are conserved.

In the following sections we illustrate the algorithm by considering a specific equation, namely the Korteveg de Vries equation.. A listing of the program written in Fortran IV which is suitable for use on the present IBM 360/44 system at PRL is given at the end. The program can be adapted to solve other equations of type (1) by suitably changing the nonlinear terms and the coefficients of the linear operator; these parts are indicated by appropriate comment cards. The plotting subroutines are specific to the PRL machine and should be suitably replaced for running on other systems. A typical integration time for a single time step Δt of .001 and n = 512 for the KdV equation is less than 2 secs.

THE ALGORITHM

The KdV equation may in general be written as

$$\frac{\partial \varphi}{\partial \tau} + \beta \varphi \frac{\partial \varphi}{\partial \xi} + \delta^2 \frac{\partial^3 \varphi}{\partial \xi^3} = 0 \qquad (2)$$

where τ is the time variable and ξ a suitably constructed space variable. β and δ are constant coefficients (e.g. $\beta = 1$ and $\delta^2 = 1/2$ for ion acoustic waves in plasmas). Note that Eq. (2) is invariant under the scaling transformation

$$\xi \rightarrow \lambda \xi \;;\; \tau \rightarrow \lambda^3 \xi \quad \text{and} \quad \varphi \rightarrow \lambda^{-2} \varphi$$

This invariance is useful for example, to normalize the scale length λ in the ξ direction to unity. There is an alternate scaling given by Berezin & Karpman (1966) which can convert all initial conditions to a unit-amplitude unit-width type of initial condition. To see this, consider a class of initial conditions of the form

$$\varphi(\xi,0) = \varphi_0\, f(\xi/L) \qquad (3)$$

where φ_0 is the initial amplitude, L is the width and $f(\xi/L)$ is the functional form of the initial pulse (viz. a square pulse, half sine pulse, secant hyperbolic pulse etc.). The following new set of variables

$$\tilde{\xi} = \xi/L \;;\; \tilde{\tau} = \varphi_0 \tau/L \text{ and } \tilde{\varphi} = \varphi/\varphi_0 \qquad (4)$$

changes the KdV equation (2) to

$$\frac{\partial \tilde{\varphi}}{\partial \tilde{\tau}} + \beta \tilde{\varphi} \frac{\partial \tilde{\varphi}}{\partial \tilde{\xi}} + \frac{\delta^2}{\sigma^2} \frac{\partial^3 \tilde{\varphi}}{\partial \tilde{\xi}^3} = 0 \qquad (4)'$$

where

$$\sigma^2 = \varphi_0 L^2$$

Under the given transformation, all initial conditions are transformed to the same initial condition i.e. unit amplitude, unit-width functions. Initial conditions with same numerical values of σ evolve identically in the scaled co-ordinate system though differently in the laboratory system. This scaling applies for all time and not merely in the asymptotic region. With the use of appropriate scaling parameter the initial condition can be simply described by the functional form of unit width and the behavior in laboratory co-ordinates can be recovered by multiplying the space, time and amplitude in the scaled co-ordinates by the scaling parameters given below:

Scaling parameter for amplitude	=	φ_0
Scaling parameter for time	=	L/φ_0
Scaling parameter for space	=	L

In case, the calculations are to be carried out without this scaling, the scaling parameter σ may be set equal to unity and initial condition may be specified by its correct amplitude and width.

Fourier transforming Eq (2) gives us

$$\frac{\partial \varphi_k}{\partial \tau} - i k^3 \delta^2 \varphi_k + i k \langle \varphi^2 \beta/2 \rangle_k = 0 \tag{5}$$

where
$$\varphi_k = \int_{-\infty}^{+\infty} \varphi(\xi, \tau) \exp(-ik\xi) d\xi \tag{6}$$

is the kth. Fourier component of $\varphi(\xi, \tau)$ and

$$\langle \varphi^2 \beta/2 \rangle_k = \frac{\beta}{2} \int_{-\infty}^{+\infty} \varphi^2(\xi, \tau) \exp(-ik\xi) d\xi \tag{7}$$

is the k th component of the nonlinear term.

Equn. (5) consists of a linear part

$$\frac{\partial \varphi_k}{\partial \tau} - i k^3 \delta^2 \varphi_k$$

and a nonlinear part $-ik \langle \beta \varphi^2/2 \rangle_k$ which can be treated as a source term. Thus we can treat the linear part exactly by writing

$$\varphi_k(\tau) = \bar{\varphi}_k(\tau) \exp(i \Omega \tau) \tag{8}$$

where $\Omega = k^3 \delta^2$ From (8) we can get by Taylor expansion

$$\varphi_k(\tau + \Delta\tau) = \left[\bar{\varphi}_k(\tau) + \left. \frac{\partial \bar{\varphi}_k}{\partial \tau} \right|_\tau \Delta\tau + \frac{1}{2} \left. \frac{\partial^2 \bar{\varphi}_k}{\partial \tau^2} \right|_\tau \Delta\tau^2 + \cdots \right] \exp[i\Omega(\tau + \Delta\tau)] \tag{9}$$

The derivatives can be evaluated from the equation for $\bar{\varphi}_k(\tau)$ which can be obtained by substituting (8) in (5)

$$\frac{\partial \bar{\varphi}_k}{\partial \tau} = -ik S_k(\tau) \exp(-i\Omega\tau) \tag{10}$$

where $S_k(\tau) = \langle \varphi^2 \beta/2 \rangle_k$. If $\Delta\tau$ is chosen such that $S_k(\tau)$ does not vary significantly in this interval then (10) can be integrated to yield

$$\bar{\varphi}_k(\tau+\Delta\tau) = \bar{\varphi}_k(\tau) + \frac{k}{\Omega} S_k(\tau)\left[\exp(-i\Omega\Delta\tau) - 1\right]\exp(-i\Omega\tau) \tag{11}$$

Hence the second term of equation (9) may be written as

$$\left.\frac{\partial\bar{\varphi}_k}{\partial\tau}\right|_\tau \exp(i\Omega\Delta\tau).\Delta\tau = \left[\bar{\varphi}_k(\tau+\Delta\tau) - \bar{\varphi}_k(\tau)\right]\exp(i\Omega\Delta\tau)$$
$$= \frac{k}{\Omega}\left[1 - \exp(i\Omega\Delta\tau)\right] S_k(\tau)\exp(-i\Omega\tau) \tag{12}$$

Similarly the higher derivatives can be evaluated by using equation (11). Note that the nonlinear term $S_k(\tau)$ is a convolution term and hence it is convenient to first go back to real space, construct the nonlinear term and then take its Fourier transform. To prevent aliasing errors only the n/2 lowest Fourier modes are retained. The solution to equation (5) can then be obtained in terms of the Taylor series of equation (9) and truncating the series at an appropriate term. An inverse Fourier transform then yields the solution to the KdV equation. The method involves a single-step time differencing scheme.

It has however been shown (Gazdag 1973) that this method of using truncated Taylor series is unstable if the truncation is effected by neglecting time derivatives higher than first $(\partial\bar{\varphi}_k/\partial\tau)$ and second $(\partial^2\bar{\varphi}_k/\partial\tau^2)$; however, it is stable if the truncation is done by neglecting derivatives higher than $\partial^3\bar{\varphi}_k/\partial\tau^3$ and $\partial^4\bar{\varphi}_k/\partial\tau^4$ Thus for stable calculations it would be necessary to calculate second and third time derivatives at each step, which may become rather involved.

In order to avoid this difficulty, following Gazdag (1976) we have used a two-step time differencing scheme, which yields stable accurate solutions, with the evaluation of only first time derivatives.

The scheme consists of (1) a predictor step, followed by (2) a corrector step. For this purpose we write equation (12) as

$$\left.\frac{\partial \bar{\varphi}_k}{\partial \tau}\right|_{\tau} \exp(i\Omega\Delta\tau)\,.\,\Delta\tau = G_k(\tau) = \frac{k}{\Omega}\left[1-\exp(i\Omega\Delta\tau)\right]\frac{S_k(\tau)}{\exp(-i\Omega\tau)} \qquad (13)$$

Then if results obtained from predictor step are denoted by super script p then the predictor step according to partially corrected second-order Adams-Bashforth Scheme (Gazdag 1976) is given by

$$\bar{\varphi}_k^{\,p}(\tau+\Delta\tau) = \bar{\varphi}_k(\tau)\exp(i\Omega\Delta\tau) + (3G_k^{\,p}(\tau) - G_k^{\,p}(\tau-\Delta\tau))/2 \qquad (14)$$

and the corrector step

$$\bar{\varphi}_k(\tau+\Delta\tau) = \bar{\varphi}_k(\tau)\exp(i\Omega\Delta\tau) + (G_k^{\,p}(\tau) + G_k^{\,p}(\tau+\Delta\tau))/2. \qquad (15)$$

Equations (14) and (15) correspond to equations (8) and (9) of Gazdag (1976) respectively. In the first time step the predictor is calculated according to the equation

$$\bar{\varphi}_k^{\,p}(\tau+\Delta\tau) = \bar{\varphi}_k(\tau)\exp(i\Omega\Delta\tau) + G_k(\tau) \qquad (16)$$

and for subsequent time steps equations (14) and (15) are used to obtain predicted and corrected values for $\bar{\varphi}_k$. Inverse transform according to the equation

$$\varphi(\tau+\Delta\tau,\xi) = \frac{1}{\sqrt{2\pi}}\,\mathrm{Re}\int_{-\infty}^{+\infty}\bar{\varphi}_k(\tau+\Delta\tau)\exp\left[i(\Omega\tau+k\xi)\right]dk \qquad (17)$$

then yields the solution of KdV equation at a given time step $(\tau+\Delta\tau)$.

To check the accuracy of the integration, the conservation of the following invariant quantities are monitored.

$$I_1 = \int_{-\infty}^{+\infty}\varphi(\xi,\tau)\,d\xi \qquad (18)$$

$$I_2 = \int_{-\infty}^{+\infty}\varphi^2(\xi,\tau)\,d\xi \qquad (19)$$

$$I_3 = \int_{-\infty}^{+\infty}\Big[\,(\,\frac{\varphi^3(\xi,\tau)}{3} - \left(\frac{\partial\varphi}{\partial\xi}\right)^2\Big]\,d\xi \qquad (20)$$

These constants are conserved to better than 1 part in 10^6 and are sensitive to the size of the time step used. The time step should be chosen to satisfy $\pi^3\delta^{-2}k_{max}^3\Delta\tau \lesssim 1$. (Abe and Inoue, 1980).

DESCRIPTION OF THE PROGRAM: "NLPDE"

a) Main Program.

In the main program the constants and functions are n initialised the input parameters are read, the differential operator, the exact time development operator "UL" (corresponding to the dispersion relation of the differential equation) and the multiplicative factors of the nonlinear operator, "UNL" are calculated and stored. Fourier transform is initialised through subroutine "PREFFT" and the initial value problem is defined through a given functional form. After transforming the initial condition to k space, the time development calculations are carried out and results are printed/plotted at regular predetermined intervals. The program makes use of the following aubroutines: PREFFT, FFT, NLROP, RESOUT

b) Subroutine NLROP (PHI, DPHI)

This subroutine calculates the nonlinear part of the time derivative by first carrying out a Fourier transform to real space, then evaluating the nonlinear term of the differential equation and then carrying out a Fourier transform of this term to k space. This subroutine uses the subroutine FFT. The nonlinear part of the derivative of PHI is returned in DPHI. PHI is not destroyed.

c) Subroutine PREFFT (EXPF, EXPB).

This subroutine initialises the Fourier Transform by calculating the exponential functions EXPF and EXPB for the forward and backward Fourier Transform and stores these functions. It also calculates and stores the parameters MIXUP which unscramble the Fourier co-efficients. This subroutine must be called before a first call of the subroutine FFT.

d) Subroutine FFT (A, B, EXPT, INV)

This subroutine calculates the Fourier Transform of the function "A". The transform is returned in function "B" and function "A" is not destroyed during the transformation. The values of integer INV and function EXPT determine the nature of the transform. INV = 1 and EXPF yield a Fourier transform from real space to k-space and INV = -1 and EXPB give a transform from k-space to real space.

e) Subroutine RESOUT (IT)

This subroutine is called when it is desired to print out the results at any given time IT. It transforms the function from k-space to real space, evaluates its real part and prints/plots the value of the function at various space points. In addition to printing/plotting the function at time IT, it also calculates the invariants and prints out their values. The subroutine makes use of the subroutines PLOT and SCAL which are provided with system IBM 360/44 at PRL and suitable modifications would be necessary for its use with any other system.

DESCRIPTION OF FUNCTIONS & PARAMETERS USED.

F (X)	:	Functional form of the initial pulse
DT	:	Time Step
L ≡ 2**N	:	Defines the number of grid points.
XMN, XMX	:	Limits of space coordinates
AMP	:	Initial Amplitude of the function F
WIDTH	:	Initial width of the function F
X1	:	Initial position of maximum of F
TMX	:	Maximum Number of Time Steps.
IDIAG	:	Number of Time Steps that elapse between each print out.
SIGMA	:	Berezin - Karpman Scaling Factor.
CAY(I)	:	K - vector
DIF (I)	:	Differential Operator.
UL (I)	:	Linear Operator
UNL (I)	:	Multiplicative factor of Non linear Operator.
X (I)	:	Space coordinates.
F (I)	:	Function values at space points.
PSI (I), PSIP (I)	:	Fourier Coefficients; corrected and predicted values.
RINV 1, RINV 2 RINV 3.	:	Invariants

ACKNOWLEDGEMENT:

We would like to thank Mr. V.R. Choksi for his assistance in the computations.

REFERENCES.

Abe K and O. Inoue (1980) J. Comput. Phys. 34, 202-210

Berezin Yu. A and V. I. Karpman (1966) ; J. Expt. Theor. Phys. 51, 1557-68 (Soviet Physics, JETP 24, 1049 (1967).

Canosa J. and J. Gazdag (1977) J. Comput. Physics 23, 393-403.

Gazdag J. (1973) J. Comput. Physics 13, 100-113

Gazdag J. (1976) J. Comput. Physics 20, 196.

Greig I. S and J. L. Morris (1976); J. Comput. Physics 20, 64

Schamel H and K. Elsasser (1976); J. Comput. Phys. 22, 501

Tappert F. (1974); Lectures in Appl. Math. 15, 215

Watanabe S; M. Ohishi and H. Tanaka (1977); J. Phys. Soc. Japan. 42, 1382.

Zabuski N. J and M. D. Kruskal (1965); Phys. Rev. Lett. 15, 240.

ANNEXURE - I

```
C      PROGRAM - ' NLPDE' -
C
C   IT SOLVES THE FIRST ORDER IN TIME PARTIAL DIFFERENTIAL EQUATION
C   EXPRESSED BY D/DT(F)=NLEOP(F)+DISP(F) WHERE NLROP IS A NONLINEAR
C   OPERATOR AND DISP STANDS FOR A LINEAR DISPERSIVE OPERATOR
C   FOR K DV EQUATION THESE OPERATORS ARE AS FOLLOW
C     NLROP =F *(D/DX)(F)    *  DISP =(D3/DX3)(F) * DELSC
C   THE PROGRAM INTEGRATES IN TIME BY USING THE PARTIALLY CORRECTED 2ND
C   ORDER ADAMS-BASHFORTH SCHEME.-SEE GAZDAG J.C.P. 20,196.THE METHOD
C   OF SPLITTING OFF THE LINEAR PART IS DUE TO TAPPERT.LECT.APP.MATH.V.
C   25,215
C
      COMMON/IGSZZZ/XMN,XMX,DELSC,YMX,YMN
      COMMON/FOURIE /EXPF(512),EXPB(512),FRWD,BKWD,UNL(512),CI,
     1L,D,TMOD
      COMMON/TET/N,MIXUP(512)
      COMMON/PLOTS/X(512),PSI(512),DIF(512),DT,NPLOT
      COMMON/RINV/RINV1,RINV2,RINV3
      INTEGER FRWD,BKWD,T,TMOD,TMX
      COMPLEX EXPF,EXPB,PSI,F(512),PSIP(512),UNL,UL(512)
      COMPLEX GD(512),GN(512),GP(512),CI,DIF
      DIMENSION CAY(512),SOSH(512)
C
C   INITIALIZE CONSTANTS AND FUNCTIONS
C
      PI=3.141592653589
      CI=CMPLX(0.,1.)
      FRWD=1
      BKWD=-1
C   READ THE PARAMETERS
C     DT=0.001
C     XMX=25.6
C     XMIN=-25.6
C     N=9
C     AMP=1.0
C     WIDTH=SQRT(60.)
C     X1= 0.0
C
      READ(5,122)AMP,WIDTH,X1,IAMP
  122 FORMAT( 3F8.4)
      READ(5,124)TMX,IDIAG,XMN,XMX,N,DT
C
C   WRITE(6,NPARMS)
      PRINT123,N,TMX,IDIAG,AMP,WIDTH,X1,XMN,XMX,DT
C
C   SIGMA IS BEREZIN-KARPMAN SCALING FACTOR USED TO SCALE THE INITIAL
C   CONDITION TO UNITY AMPLITUDE UNIT WIDTH FUNCTION.
C   SIGMA=6.0 CORRESPONDS TO EXACT SOLITARY SOLUTION.
C   FUNCTION SCALES AS AMPLITUDE,THE SPACE AS WIDTH AND THE TIME
C   AS (WIDTH/AMPLITUDE).
C   INITIAL CONDITIONS WITH SAME VALUE OF SIGMA PROPAGATE SIMILARLY
C   IN SCALED SPACE-TIME BUT DIFFERENTLY IN LABORATORY.
C   SCALING MAY BE REMOVED BY DEFINING SIGMA=1.0
      IF(IAMP.LT.0)GOTO
      YMX=1.5
      YMN=-0.6
      GO TO 9
    8 YMX=0.5
      YMN=-1.0
    9 CONTINUE
      SCALT=WIDTH/AMP
      SIGMA=AMP*(WIDTH**2)
      WRITE(6,125) AMP,WIDTH,SCALT,SIGMA
      L=2**N
      D=XMX-XMN
      DX=D/L

 INITIALIZE FOURIER TRANSFORM

      CALL PREFFT(EXPF,EXPB)

 SET UP DIFFERENTIAL OPERATOR AND TRUNCTION PARAMETER

      DO3I=1,L
      SOSH(I)=1
      IF(I.GE.L/3.AND.I.LE.(2*L/3))SOSH(I)=0
    3 CAY(I)=2*PI*(I-1-(L/2))/D

 SET UP EXACT TIME DEVELOPMENT OPERATOR UL CORRESPONDING TO THE
 DISPERSION RELATION OF THE D.E. AND THE MULTIPLICATIVE FACTOR OF THE
 NON-LINEAR OPERATOR UNL.

      DELSQ=0.50/SIGMA
      UNL(1)=0
      DO7I=1,L
      J=MOD(I-1+L/2,L)+1
      DIF(J)=-(CAY(I)*CI)
      OMEGA=-CAY(I)*CAY(I)*CAY(I)*DELSQ
      UL(J)=CEXP(CI*OMEGA*DT)
      IF(J.EQ.1)GOTO7
      UNL(J)=-CAY(I)*SOSH(J)/OMEGA*(1-UL(J))
    7 CONTINUE

 DEFINE INITIAL VALUE PROBLEM
 IF SCALING IS NOT USED THE INITIAL VALUE FUNCTION SHOULD BE DEFINED
 WITH CORRECT AMPLITUDE AND WIDTH.

      DO1J=1,L
      X(J)=XMN+DX*(J-.5)
      XJ1=X(J)-X1
      IF(ABS(XJ1).LE.1.0) GO TO 5
      FJ=0.0
      GO TO 6
    5 FJ=(1.0+SIN(PI*(XJ1+0.5)))/2.0
      FJ=-FJ
    6 F(J)=CMPLX(FJ,0.0)
    1 CONTINUE
```

```
  TRANSFORM INITIAL CONDITION TO K-SPACE.

    CALL FFT(F,PSI,EXPF,FRWD)
    NPLOT=0
    T=0
    CALL RESOUT(T)

  FIND FIRST TIME STEP FOR PSI

    CALL NLROP(PSI,GO)
    WRITE(6,999)RINV1,RINV2,RINV3
999 FORMAT( 'I1=' ,F15.9, 'I2=' ,F15.9, 'I3=' ,F15.9)
    DO10I=1,L
 10 PSIP(I)=PSI(I)*UL(I)+GO(I)
  INCREAMENT TIME
    DO188T=1,TMX
    CALL NLROP(PSIP,GP)
    DO28I=1,L
  CAZDAG EQUATION 9
    PSI(I)=PSI(I)*UL(I)+0.5*(GO(I)+GP(I))
    GM(I)=GO(I)
    GO(I)=GP(I)
  CAZDAG EQUATION 8
    PSIP(I)=PSI(I)*UL(I)+0.5*(3*GO(I)-GM(I))
 28 CONTINUE
    TMOD=MOD(T,IDIAG)
    IF(TMOD.NE.0)GOTO188
    CALL RESOUT(T)
    WRITE(6,999)RINV1,RINV2,RINV3
188 CONTINUE
123 FORMAT(1X,3I5/(3F12.4))
124 FORMAT(2I6,2F8.4,I5,E10.4)
125 FORMAT(' SCALING FACTORS'/' AMPLITUDE'E12.4'  SPACE'E12.4,' TIME',
   1E12.4/' SIGMA'E12.4/)
    STOP
    END



    SUBROUTINE NLROP(PHI,DPHI)

  THIS SUBROUTINE FINDS THE NONLINEAR PART OF THE TIME DERIVATIVE

    COMMON/FOURIE /EXPF(512),EXPB(512),FRWD,BKWD,UNL(512),C1,
   1L,D,ITMOD
    COMPLEX PHI(1),DPHI(1),FPHI(512),SC(512)
    COMPLEX EXPF,EXPB,C1,UNL

  FIND THE CONVOLUTION TERM

    CALL FFT(PHI,FPHI,EXPB,BKWD)
    DO40I=1,L
    FPHI(I)=(FPHI(I)**2)/2.00
 40 CONTINUE
```

```
      CALL FFT(FPHI,SQ,EXPF,FFWD)
C
C   FIND THE NONLINEAR PART OF THE TIME DERIVATIVE
C
   45 DO50I=1,I
   50 DPHI(I)=UNL(I)*SQ(I)
      RETURN
      END


      SUBROUTINE PREFFT(EXPF,EXPB)
      COMMON/TFT/N,MIXUP(512)
      COMPLEX EXPF(1),EXPB(1)
      TWOPI=6.283185307179
      K=2**N
      K2=K/2
      DO10L=1,K
      LB=L-1
      LL=0
      DO11J=1,N
      JB=LB/2
      KIT=LB-2*JB
      LB=JB
   11 LL=2*LL+KIT
      MIXUP(L)=LL+1
   10 CONTINUE
      DO12KHETA=1,K2
      C=COS(TWOPI*(KHETA-1)/K)
      S=SIN(TWOPI*(KHETA-1)/K)
      EXPF(KHETA)=CMPLX(C,S)
      EXPB(KHETA)=CMPLX(C,-S)
   12 CONTINUE
      RETURN
      END


      SUBROUTINE FFT(A,B,EXPT,INV)
      COMMON/TFT/N,MIXUP(512)
      COMPLEX A(1),B(1),EXPT(1),B0,B1
      K=2**N
      K1=1
      K0=2
      DO1L=1,K
      LB=MIXUP(L)
    1 B(L)=A(LB)
      DO2MERGE=1,N
      DO3KP=1,K1
      KHETA=(KP-1)*K/K0+1
      DO4J=KP,K,K0
      JP=J+K1
      B0=B(J)
      B1=B(JP)*EXPT(KHETA)
      B(J)=B0+B1
    4 B(JP)=B0-B1
    3 CONTINUE
      K1=K0
    2 K0=2*K1
      IF(INV.NE.1)GOTO10
      S=K
      R=1/S
      DO6L=1,K
    6 B(L)=B(L)*R
   10 CONTINUE
      RETURN
      END


      SUBROUTINE RESULT(IT)
      COMMON/IGSZZZ/XMN,XMX,DELSQ,YMX,YMN
      COMMON/FOURIE /EXPF(512),EXPB(512),FRWD,BKWD,UNL(512),CI,
     1L,D,ITMOD
      COMMON/PLOTS/X(512),PSI(512),DIF(512),DT,NPLOT
      COMMON/RINV/RINV1,RINV2,RINV3
      COMPLEX EXPF,EXPB,UNL,CI,DIF
      COMPLEX PSI,FPSI(512),DPSI(512)
      DIMENSION Y(512),TITLE(4),CAY(512)
    EQUIVALENCE(FPSI(1),CAY(1))
      TIME=IT*DT
      DX3=D/L/3.0
      DELD=DELSQ/D
      RINV2=0.0

    TRANSFORM TO X SPACE AND CALCULATE INVARIANT 2

      CALL FFT(PSI,FPSI,EXPB,BKWD)
      DO5I=1,L
      FPHISQ=FPSI(I)*CONJG(FPSI(I))
      RINV2=RINV2+FPHISQ
    5 Y(I)=REAL(FPSI(I))
      RINV2=RINV2*D/L

    PRINT OUT VALUES OF FUNCTIONS

      WRITE(6,901)
      WRITE(6,903)TIME
      L4=L/4
      DO7I=1,L4
      J=I+L4
      K=J+L4
      M=K+L4
      WRITE(6,902)I,Y(I),J,Y(J),K,Y(K),M,Y(M)
    7 CONTINUE

    CALCULATE INVARIENTS 1 AND 3
```

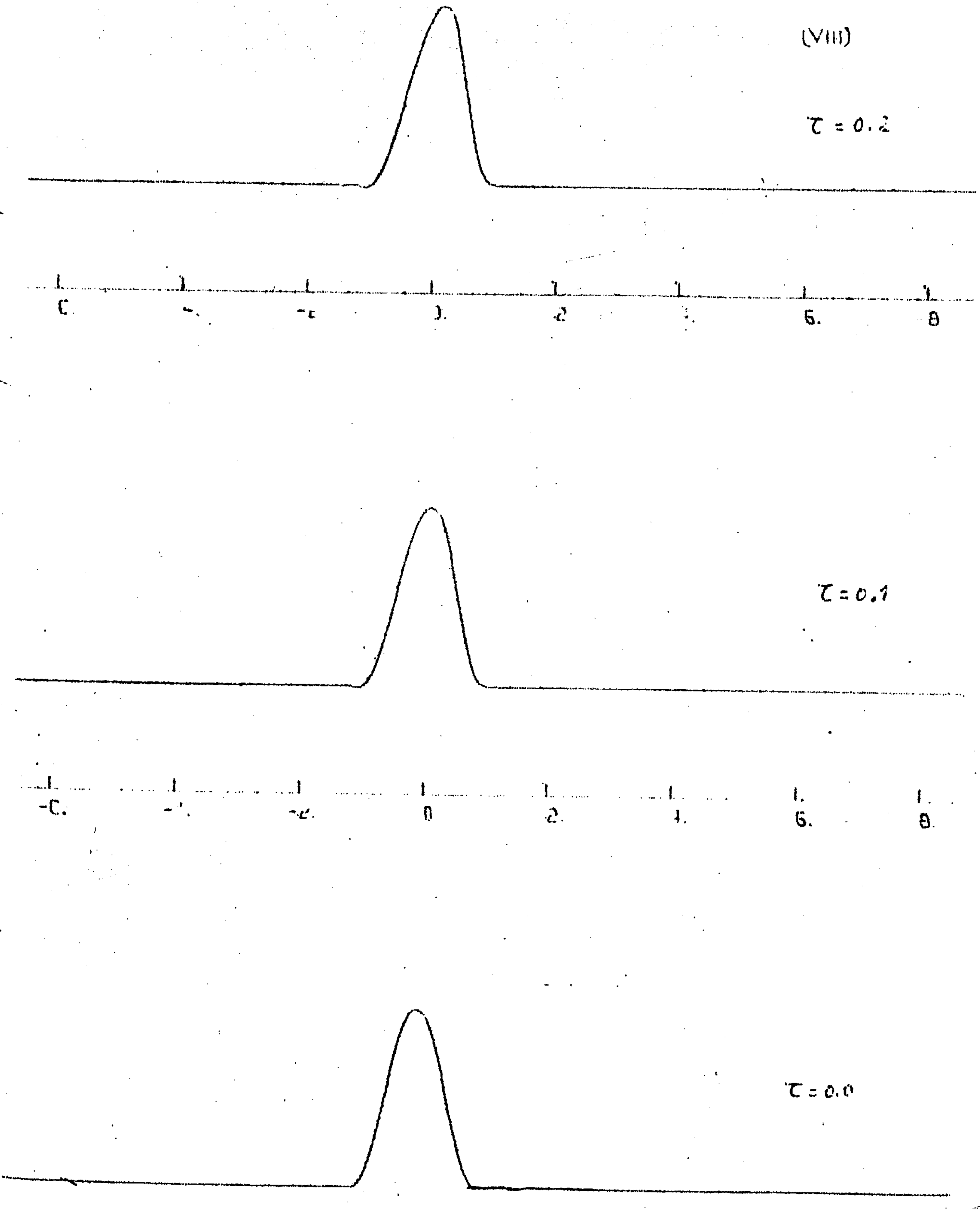
```
      RINV1=00.00
      RINV3=0.
      DO20I=1,L
      RINV1=RINV1+Y(I)*D/L
      DPSI(I)=-DIF(I)*PSI(I)
      W=DPSI(I)*CONJG(DPSI(I))
   20 RINV3=RINV3+(Y(I)**3)*DX3-W*DFLD
C     CALCULATE MAXIMUM OF FUNCTION TO BE PLOTTED
C
C     YMN=0
C     YMX=0
C     DO10J=1,L
C     IF(Y(J).LE.YMN)YMN=Y(J)
C     IF(Y(J).GE.YMX)YMX=Y(J)
C  10 CONTINUE
C     IF(YMX.GT.100)STOP
C     YMIN=AINT(YMN)
C     YMAX=AINT(YMX)
C     YMN =YMIN+AINT((YMN-YMIN)*10.0-1.0)/10.0
C     YMX=YMAX+AINT((YMX-YMAX)*10.0+1.0)/10.0
C     YL=2.00
C     YD=(YMX-YMN)/YL
      XL=10.00
      XD=(XMX-XMN)/XL
      YL =3.0
      YD=(YMX-YMN)/YL
      CALL PLOT(1,XMN,XMX,XL,XD,YMN,YMX,YL,YD)
      CALL SCAL(   XMN,XMX,XL,XD,YMN,YMX,YL,YD)
      DO510II=1,L
      CALL PLOT(90,X(II),Y(II))
  510 CONTINUE
      CALL PLOT(99)
      IF(NPLOT .EQ.2)GOTO60
      NPLOT=NPLOT+1
      YMAX=YMX+YD
      CALL PLOT(91,XMN,YMAX)
      RETURN
   60 NPLOT=0
      PAUSE'PLEASE ARRANGE THE PLOTTER-PEN ON THE NEXT PAGE'
C
  904 FORMAT( ' THE THIRD INVARIENT IS ' E15.9)
  905 FORMAT( '  THE INTEGRAL OF PHI SQUARED IS' E15.9)
  900 FORMAT('VEL1='F5.2,'   VEL2='F5.2,'   TIME='F5.2)
  901 FORMAT(1H1)
  902 FORMAT(1X,4(I3,4X,F10.4,15X))
  903  FORMAT( '   TIME IS'  F11.4)
      RETURN
      END
```

ANNEXURE II

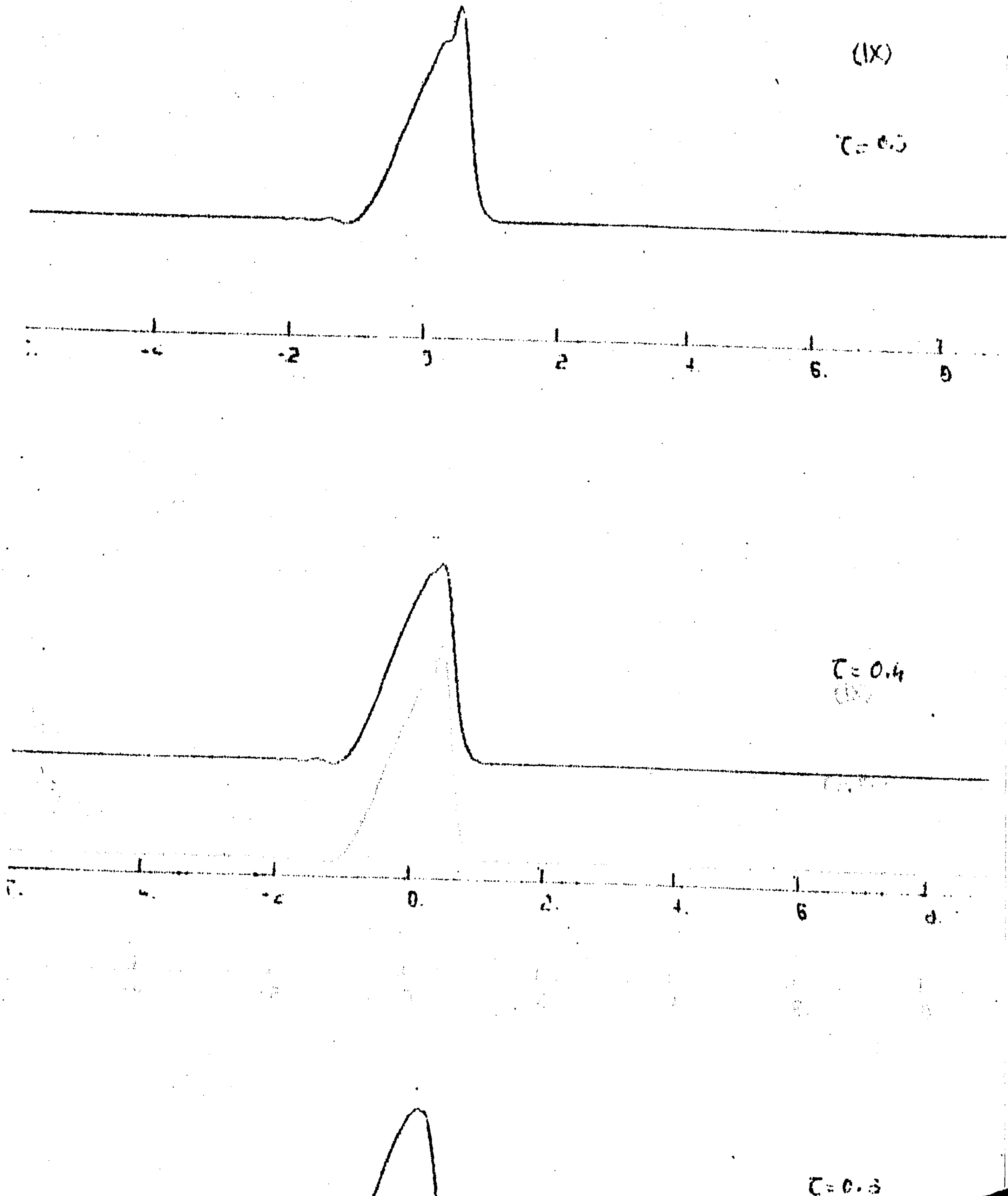
Caption to figures.

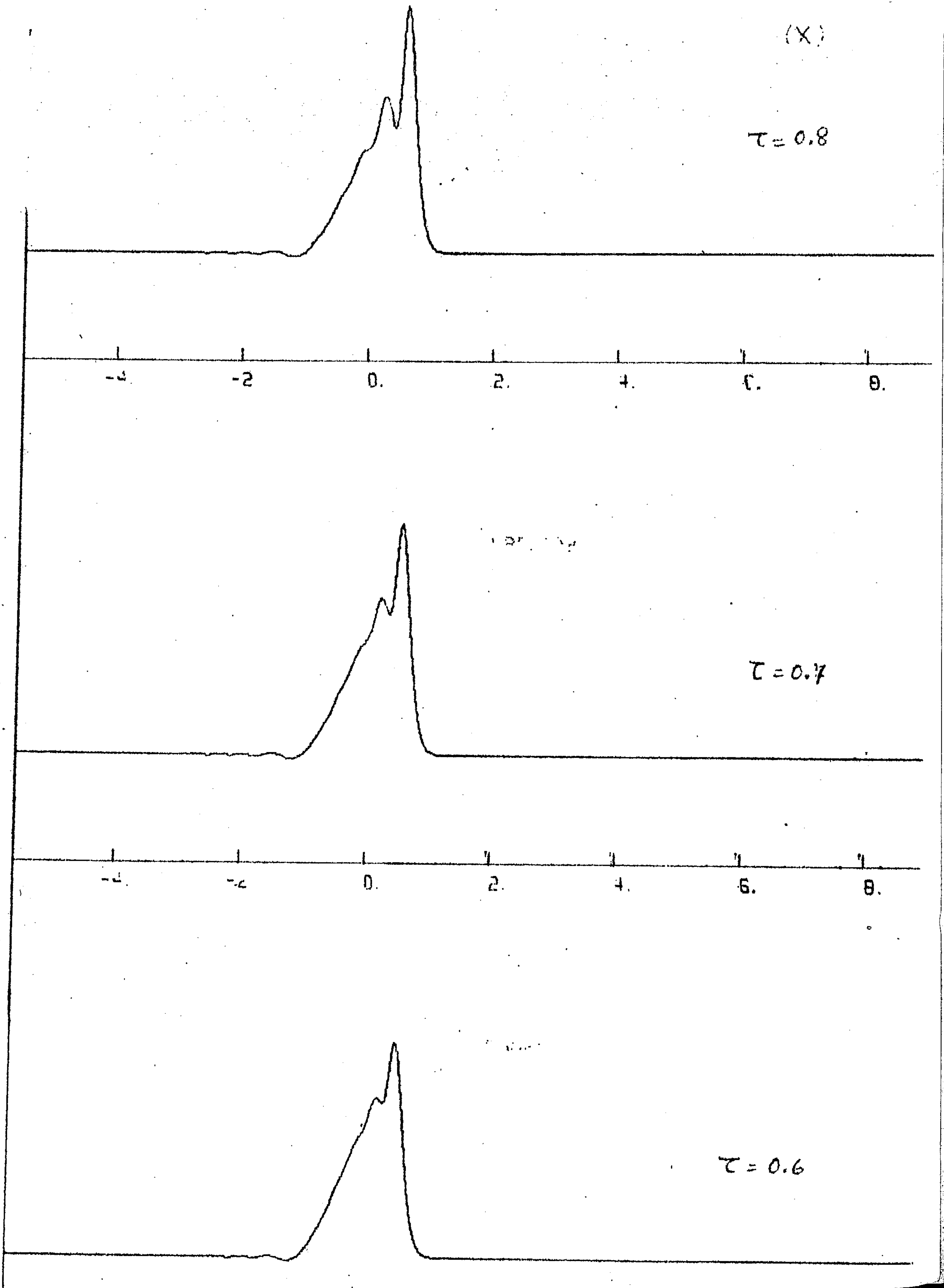
In subsequent figures we show the evolution of a single sinusoidal positive pulse as obtained by the numerical solution of the KdV equation. Amplitude, time and space are normalised according to Berezin-Karpman scaling and the scaling factor is $\sigma^2 = 250$. The process of steepening at the front edge and breaking of the initial perturvation into a number of solitons is clearly demonstrated.

(VIII)



(IX)




(X)
τ = 0.8
-4.
-2
0.
2.
4.
6.
8.
τ = 0.7
-4.
-2
0.
2.
4.
6.
8.
τ = 0.6