Computation of solitary wave profiles described by a Hamiltonian model for surface waves

Technische
Universiteit
Eindhoven

Opleiding
Wiskunde voor de Industrie
Eindhoven

STUDENT REPORT 91-02

COMPUTATION OF SOLITARY WAVE PROFILES
DESCRIBED BY A HAMILTONIAN MODEL FOR
SURFACE WAVES

APPENDICES

DRS. T.J.G. ZWARTKRUIS

MARCH 1991

ECMI

Den Dolech 2
Postbus 513
5600 MB Eindhoven

Eindverslag van de ontwerpersopleiding
WISKUNDE VOOR DE INDUSTRIE

Final report of the postgraduate programme
MATHEMATICS FOR INDUSTRY

COMPUTATION OF SOLITARY WAVE PROFILES DESCRIBED BY A HAMILTONIAN MODEL FOR SURFACE WAVES

APPENDICES

Drs T.J.G. Zwartkruis

University supervisor: Dr J. Molenaar,
Technische Universiteit, Eindhoven

Industrial supervisor: Drs A.C. Radder,
Rijkswaterstaat, Tidal Waters Devision, The Hague (NL)

March 1991

APPENDIX I

Ad footnote (1):

Let $V(p) \equiv \hat{\phi}_p(p)$. Then

$$I(p) \equiv \frac{d}{dp}\int_{-\infty}^{p}\int_{p}^{\infty}\frac{V(q)V(r)}{\sinh(2(r-q))}drdq = \frac{d}{dp}\int_{-\infty}^{0}\int_{0}^{\infty}\frac{V(q'+p)V(r'+p)}{\sinh(2(r'-q'))}dr'dq' =$$

$$\int_{-\infty}^{0}\left\{\left[\frac{d}{dp}V(q'+p)\right]\int_{0}^{\infty}\frac{V(r'+p)}{\sinh(2(r'-q'))}dr'\right\}dq' +$$

$$\int_{0}^{\infty}\left\{\left[\frac{d}{dp}V(r'+p)\right]\int_{-\infty}^{0}\frac{V(q'+p)}{\sinh(2(r'-q'))}dq'\right\}dr' .$$

Integration by parts yields:

$$\int_{0}^{\infty}\frac{V(r'+p)}{\sinh(2(r'-q'))}dr' = -\tfrac{1}{2}V(p)ln(tanh\{-q'\}) - \tfrac{1}{2}\int_{0}^{\infty}\frac{dV}{dr'}(r'+p)ln(tanh\{r'-q'\})dr'$$

and

$$\int_{-\infty}^{0}\frac{V(q'+p)}{\sinh(2(r'-q'))}dq' = -\tfrac{1}{2}V(p)ln(tanh\{r'\}) + \tfrac{1}{2}\int_{-\infty}^{0}\frac{dV}{dq'}(q'+p)ln(tanh\{r'-q'\})dq'$$

so

$$I(p) = -\tfrac{1}{2}V(p)\int_{-\infty}^{\infty}\frac{dV}{dq'}(q'+p)ln(tanh\{|q'|\})dq' .$$

With the help of *(3.11)* and $q'=q-p$ equation *(3.15)* is found.

Ad footnote (3):

Let $V(x)\equiv\phi_x$, $J(x,t)\equiv\int_{-\infty}^{x}\int_{x}^{\infty}\frac{V(x')V(x'')}{\sinh\left(\frac{\pi}{2}\left|\int_{x'}^{x''}\frac{1}{\eta}dr\right|\right)}dx''dx'$. Then

$$J(x+\delta,t) - J(x-\delta,t) = \int_{x-\delta}^{x+\delta}\int_{x}^{\infty}\frac{V(x')V(x'')}{\sinh\left(\frac{\pi}{2}\left|\int_{x'}^{x''}\frac{1}{\eta}dr\right|\right)}dx''dx' -$$

$$\int_{-\infty}^{x}\int_{x-\delta}^{x+\delta}\frac{V(x')V(x'')}{\sinh\left(\frac{\pi}{2}\left|\int_{x'}^{x''}\frac{1}{\eta}dr\right|\right)}dx''dx' + O(\delta^2) .$$

Changing the order of integration in the second integral yields:

$$J(x+\delta,t) - J(x-\delta,t) = \int_{x-\delta}^{x+\delta} \left\{ \int_{x}^{\infty} - \int_{-\infty}^{x} \right\} \frac{V(x')V(x'')}{\sinh\left(\frac{\pi}{2}\left|\int_{x'}^{x''} \frac{1}{\eta} dr\right|\right)} dx''dx' + O(\delta^2) =$$

$$\int_{x-\delta}^{x+\delta} V(x') \left\{ \int_{-\infty}^{\infty} \frac{V(x'')U(x''-x)}{\sinh\left(\frac{\pi}{2}\left|\int_{x'}^{x''} \frac{1}{\eta} dr\right|\right)} dx'' - \int_{-\infty}^{\infty} \frac{V(x'')U(x-x'')}{\sinh\left(\frac{\pi}{2}\left|\int_{x'}^{x''} \frac{1}{\eta} dr\right|\right)} dx'' \right\} dx' + O(\delta^2),$$

where U denotes the Heavyside-function, i.e., $\begin{cases} U(x) = 1 , & x \geq 0 \\ U(x) = 0 , & x < 0. \end{cases}$

With the help of

$$\frac{\partial}{\partial x} \int_{-\infty}^{\infty} V(x') \ln\left(\tanh\left\{\frac{\pi}{4}\left|\int_{x}^{x'} \frac{1}{\eta} dr\right|\right\}\right) dx' = \frac{\pi}{2\eta(x)} \int_{-\infty}^{\infty} V(x') \frac{(U(x-x')-U(x'-x))}{\sinh\left(\frac{\pi}{2}\left|\int_{x}^{x'} \frac{1}{\eta} dr\right|\right)} dx'$$

and *(4.4)* one derives that

$$\lim_{\delta \downarrow 0} \frac{J(x+\delta,t) - J(x-\delta,t)}{2\delta} = V(x) \left\{ \int_{-\infty}^{\infty} V(x'') \frac{(U(x''-x)-U(x-x''))}{\sinh\left(\frac{\pi}{2}\left|\int_{x}^{x''} \frac{1}{\eta} dr\right|\right)} dx'' \right\} \stackrel{!}{=}$$

$- V(x) \frac{2}{\pi}\eta(x)\pi\zeta_t = - 2\eta V(x)\zeta_t$, and thus *(4.6)* is found.

Ad footnote (4):

Substitution of *(4.10)* into *(4.1)* gives

$$\frac{\partial \mathcal{H}}{\partial p_k} = \frac{\partial}{\partial p_k}\left(- \frac{1}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left\{\sum_j p_j S_j\right\}_x \left\{\sum_m p_m S_m\right\}_{x'} \ln\left(\tanh\left\{\frac{\pi}{4}\left|\int_{x}^{x'} \frac{1}{\eta} dr\right|\right\}\right) dx'dx \right) =$$

$$- \frac{1}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left\{ (S_k)_x \phi_{x'} + \phi_x (S_k)_{x'} \right\} \ln\left(\tanh\left\{\frac{\pi}{4}\left|\int_{x}^{x'} \frac{1}{\eta} dr\right|\right\}\right) dx'dx$$

Using symmetry leads to *(4.12)*.

Substitution of *(4.10)* into *(4.1)* also gives

$$- \frac{\partial \mathcal{H}}{\partial q_k} = - \frac{\partial}{\partial q_k}\left(\frac{1}{2} \int_{-\infty}^{\infty} \left\{\sum_j q_j S_j\right\}^2 dx - \right.$$

$$\left. \frac{1}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \phi_x \phi_{x'} \ln\left(\tanh\left\{\frac{\pi}{4}\left|\int_{x}^{x'} \frac{1}{h+\sum_j q_j S_j} dr\right|\right\}\right) dx'dx \right) =$$

$$-\int_{-\infty}^{\infty}\zeta S_k\,dx + \frac{1}{2\pi}\int_{-\infty}^{\infty}\phi_x\left\{\int_{-\infty}^{x}\phi_{x'}\frac{-\frac{\pi}{2}\int_{x'}^{x}\frac{S_k}{\eta^2}dr}{\sinh\left(\frac{\pi}{2}\left|\int_{x'}^{x}\frac{1}{\eta}dr\right|\right)}dx' + \right.$$

$$\left.\int_{x}^{\infty}\phi_{x'}\frac{-\frac{\pi}{2}\int_{x}^{x'}\frac{S_k}{\eta^2}dr}{\sinh\left(\frac{\pi}{2}\left|\int_{x'}^{x}\frac{1}{\eta}dr\right|\right)}dx'\right\}dx =$$

$$-q_k - \frac{1}{2}\int_{-\infty}^{\infty}\phi_x\int_{-\infty}^{\infty}\phi_{x'}U(x-x')\frac{\int_{-\infty}^{\infty}\frac{S_k}{\eta^2}(r)U(x-r)U(r-x')dr}{2\sinh\left(\frac{\pi}{2}\left|\int_{x'}^{x}\frac{1}{\eta}dr\right|\right)}dx'\,dx -$$

$$\frac{1}{2}\int_{-\infty}^{\infty}\phi_x\int_{-\infty}^{\infty}\phi_{x'}U(x'-x)\frac{\int_{-\infty}^{\infty}\frac{S_k}{\eta^2}(r)U(x'-r)U(r-x)dr}{2\sinh\left(\frac{\pi}{2}\left|\int_{x'}^{x}\frac{1}{\eta}dr\right|\right)}dx'\,dx =$$

$$-q_k - \frac{1}{2}\int_{-\infty}^{\infty}\frac{S_k}{\eta^2}(r)\left\{\int_{-\infty}^{\infty}\phi_x U(x-r)\int_{-\infty}^{\infty}\phi_{x'}\frac{U(x-x')U(r-x')}{2\sinh\left(\frac{\pi}{2}\left|\int_{x'}^{x}\frac{1}{\eta}dr\right|\right)}dx'\,dx + \right.$$

$$\left.\int_{-\infty}^{\infty}\phi_x U(r-x)\int_{-\infty}^{\infty}\phi_{x'}\frac{U(x'-x)U(x'-r)}{2\sinh\left(\frac{\pi}{2}\left|\int_{x'}^{x}\frac{1}{\eta}dr\right|\right)}dx'\,dx\right\}dr =$$

$$-q_k - \frac{1}{2}\int_{-\infty}^{\infty}\frac{S_k}{\eta^2}(r)\left\{\int_{r}^{\infty}\phi_x\int_{-\infty}^{r}\phi_{x'}\frac{1}{2\sinh\left(\frac{\pi}{2}\left|\int_{x'}^{x}\frac{1}{\eta}dr\right|\right)}dx'\,dx + \right.$$

$$\left.\int_{-\infty}^{r}\phi_x\int_{r}^{\infty}\phi_{x'}\frac{1}{2\sinh\left(\frac{\pi}{2}\left|\int_{x'}^{x}\frac{1}{\eta}dr\right|\right)}dx'\,dx\right\}dr =$$

$$- q_k - \frac{1}{2}\int_{-\infty}^{\infty} \frac{S_k}{\eta^2}(r) \int_{-\infty}^{r} \phi_x \Big| \int_{r}^{\infty} \phi_{x'} \frac{1}{\sinh\left(\frac{\pi}{2}\left|\int_{x'}^{x} \frac{1}{\eta} dr\right|\right)} dx' dx dr .$$

So, *(4.13)* is found.

Ad footnote (5):

Write $\left(v(x',t) - \frac{\eta(x,t)}{\eta(x',t)} v(x,t)\right) + \frac{\eta(x,t)}{\eta(x',t)} v(x,t)\Big)$ instead of $v(x',t)$ in the right hand side of *(4.16)*. With the help of *(3.19)* one can derive:

$$\int_{-\infty}^{\infty} \frac{1}{\eta(x',t)} \ln\left(\tanh\left\{\frac{\pi}{4}\left|\int_{x}^{x'} \frac{1}{\eta} dr\right|\right\}\right) dx' = \frac{4}{\pi}\int_{-\infty}^{\infty} \ln(\tanh\{|p-q|\}) dq = -\pi ,$$

and this leads to *(4.16)*.

Footnote (6):

The kinetic energy is given by $\mathcal{T} = -\frac{1}{2\pi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} v(x,t)v(x',t)dx'dx$. Write

$$v(x,t)v(x',t) = \frac{1}{2}\left(\left\{v(x,t) - \frac{\eta(x',t)}{\eta(x,t)} v(x',t)\right\}\left\{v(x',t) - \frac{\eta(x,t)}{\eta(x',t)} v(x,t)\right\} + \right.$$

$$\left. \frac{v^2(x',t)\eta(x',t)}{\eta(x,t)} + \frac{v^2(x,t)\eta(x,t)}{\eta(x',t)} \right) .$$

The same tricks as above and the fact that v vanishes outside the interval *[0,L]* now lead to *(4.24)*.

APPENDIX 2

This appendix contains the source texts of the used FORTRAN programs. The comments are written in Dutch.

Ad footnote (2):

```
      PROGRAM EENLGOLF
C
C     *****************************************************************
C     VERSIE DD 11-1-91
C
C     Dit programma berekent een eenlinggolfprofiel door een niet-
C     lineaire integraalvergelijking m.b.v. Picard-iteratie bena-
C     derend op te lossen.
C     Tevens wordt de bijbehorende snelheid aan het golfoppervlakte
C     berekend. De resultaten worden naar de dimensieloze plaatsvaria-
C     bele X teruggeschaald.
C
C     De gebruiker dient als invoer te geven:
C     (1) de bovengrens UPB van het te gebruiken interval.
C         Dit betekent dat de integraal afgekapt wordt tot het interval
C         [-UPB,UPB].
C     (2) Het aantal roosterpunten N op de rechterhelft van het inter-
C         val. Dit betekent dat er roosterpunten met de indices
C         -N t/m N bepaald worden.
C     (3) het bijbehorende Froude-getal C
C     (4) het maximale aantal iteratieslagen ITMAX
C     (5) de index M van het roosterpunt dat als tussentijds vast punt
C         dienst doet en waarmee geschaald wordt.
C     (6) de gewenste relatieve fout tussen twee opeenvolgende
C         iteranden (gemeten in Euclidische norm). Deze wordt gebruikt
C         bij het stopcriterium.
C     De invoer moet in het file "eenl.in" staan.
C
C     Het programma stopt als er ITMAX slagen uitgevoerd zijn of als
C     (na symmetrisch maken van de oplossing) de relatieve fout tussen
C     twee opeenvolgende iteranden gehaald is.
C
C     De uitvoer wordt weggeschreven naar de file "eenl.uit" en bevat
C     de invoer, het aantal iteratieslagen en de gevonden oplossing.
C
C     *****************************************************************
C
C
C
C
C
C
C
```

```
C      **************************************************************
C      declaraties
C      **************************************************************
      IMPLICIT   NONE
      INTEGER    NMIN,NMAX,DNMAX
      REAL       PI
      PARAMETER (NMIN=-250,NMAX=250,DNMAX=500,PI=3.141592653590)
      INTEGER    N,TELLER,ITMAX,I,M,KLAAR,ST
      REAL       UPB,STAPG,C,ERRREL
      REAL       ROOS(NMIN:NMAX),ZETA(NMIN:NMAX),DFI(NMIN:NMAX)
      REAL       HLNTH(DNMAX)
C
      INTRINSIC MOD
C
C      **************************************************************
C      hoofdprogramma
C      **************************************************************
      CALL LEESIN(UPB,N,C,ITMAX,M,ERRREL)
      CALL MEQDR(ROOS,UPB,N,STAPG)
      CALL MAAKTAB(HLNTH,2*N,STAPG)
      CALL INIT(ZETA,ROOS,N,C)
      KLAAR=0
C      maak het profiel tussentijds symmetrisch
      DO 10 TELLER=1,ITMAX
         IF (MOD(TELLER,10).EQ.0) THEN
         DO 20 I=1,N
            ZETA(I)=0.5*(ZETA(I)+ZETA(-I))
            ZETA(-I)=ZETA(I)
   20    CONTINUE
         ENDIF
         CALL BERSNEL(ZETA,DFI,N,C)
         CALL BERPROF(ZETA,DFI,HLNTH,N,STAPG,M,ERRREL,KLAAR)
         IF (KLAAR.EQ.1) THEN
         DO 30 I=1,N
            ZETA(I)=0.5*(ZETA(I)+ZETA(-I))
            ZETA(-I)=ZETA(I)
   30    CONTINUE
         KLAAR=0
         CALL BERSNEL(ZETA,DFI,N,C)
         CALL BERPROF(ZETA,DFI,HLNTH,N,STAPG,M,ERRREL,KLAAR)
         IF (KLAAR.EQ.1) THEN
         GOTO 25
         ENDIF
         ENDIF
   10 CONTINUE
   25 ST=TELLER
      CALL TERUG(ROOS,ZETA,DFI,N,STAPG,C)
      CALL SCHRIJF(UPB,N,C,ITMAX,M,ERRREL,ST,ROOS,ZETA,DFI)
      END
C      ------------------------------------------------------------
C
C
C
C
C
C
C
C
C
```

```
C     ************************************************************
C     subroutines en functies
C     ************************************************************
      SUBROUTINE LEESIN(UPB,N,C,ITMAX,M,ERRREL)
C
C     deze procedure leest de invoer. De invoer wordt gevormd door
C     UPB,N,C,ITMAX,M en ERRREL. De invoer dient in de file
C     "eenl.in" te staan.
C
      IMPLICIT  NONE
      INTEGER   N,ITMAX,M
      REAL      UPB,C,ERRREL
C
      OPEN(UNIT=20,FILE='eenl.in')
      READ (20,*) UPB
      READ (20,*) N
      READ (20,*) C
      READ (20,*) ITMAX
      READ (20,*) M
      READ (20,*) ERRREL
      RETURN
      END
C
C     ------------------------------------------------------------
C
      SUBROUTINE MEQDR(X,UPB,N,DX)
C
C     deze routine zorgt dat de stapgrootte DX bepaald wordt en
C     maakt equidistant rooster X(-N) t/m X(N) op interval [-UPB,UPB]
C
      IMPLICIT  NONE
      INTEGER   NMIN,NMAX
      PARAMETER (NMIN=-250,NMAX=250)
      INTEGER   N,TELLER
      REAL      UPB,DX
      REAL      X(NMIN:NMAX)
C
      DX=UPB/N
      X(0)=0.0
      DO 10 TELLER=1,N
         X(TELLER)=TELLER*DX
         X(-TELLER)=-X(TELLER)
   10 CONTINUE
      RETURN
      END
C
C     ------------------------------------------------------------
C
      SUBROUTINE MAAKTAB(HLNTH,N,DX)
C
C     deze routine voorkomt onnodig rekenwerk en maakt een tabel. De
C     waarde DX*ln(tanh(i*DX)) wordt in array-element HLNTH(i) opgesla-
C     gen.
C
      IMPLICIT  NONE
      INTEGER   DNMAX
      PARAMETER (DNMAX=500)
      INTEGER   N,TELLER
      REAL      DX
```

```
      REAL       HLNTH(DNMAX)
C
      INTRINSIC LOG,TANH
C
      DO 10 TELLER=1,N
         HLNTH(TELLER)=DX*LOG(TANH(TELLER*DX))
   10 CONTINUE
      RETURN
      END
C
C     ---------------------------------------------------------------
C
      SUBROUTINE INIT(Z,ROOS,N,C)
C
C     deze subroutine berekent het startprofiel voor de iteratie,
C     d.w.z., Z(i):=(C**2-1)/(cosh**2(sqrt(3(C**2-1)/4)*ROOS(i)))
C     N.B.  Z(0)=(Z(0)+Z(1))/2!
C
      IMPLICIT  NONE
      INTEGER   NMIN,NMAX
      REAL      PI
      PARAMETER (NMIN=-250,NMAX=250,PI=3.141592653590)
      INTEGER   N,TELLER
      REAL      C,ALFA,BETA
      REAL      Z(NMIN:NMAX),ROOS(NMIN:NMAX)
C
      INTRINSIC COSH,SQRT
C
      ALFA=(C**2-1)
      BETA=SQRT(0.75*ALFA)
      DO 10 TELLER=-N,N
         Z(TELLER)=ALFA/(COSH(BETA*ROOS(TELLER))**2)
   10 CONTINUE
      Z(0)=0.5*(Z(1)+Z(0))
      RETURN
      END
C
C     ---------------------------------------------------------------
C
      SUBROUTINE BERPROF(Z,F,HLNTH,N,H,M,ERRREL,KLAAR)
C
C     deze routine berekent het nieuwe golfprofiel (zeta) gegeven DFI
C     en fixeert zeta(M) op zijn vaste geinitialiseerde waarde.
C     Tevens wordt gekeken of er aan het stopcriterium met de rela-
C     tieve fout wordt voldaan.
C
      IMPLICIT  NONE
      INTEGER   NMIN,NMAX,DNMAX
      REAL      PI
      PARAMETER (NMIN=-250,NMAX=250,DNMAX=500,PI=3.141592653590)
      INTEGER   I,N,M,KLAAR
      REAL      H,SUM,SCHF,TWNORM,TWNZ,ERRREL
      REAL      Z(NMIN:NMAX),ZOUD(NMIN:NMAX),F(NMIN:NMAX),
     +          HLNTH(DNMAX)
C
      INTRINSIC SQRT,ABS
C
      DO 5 I=-N,N
         ZOUD(I)=Z(I)
```

```
    5 CONTINUE
      Z(-N)=0.25*PI*F(-N)-(0.5*HLNTH(2*N)*(F(N)-F(-N))+
     +                                SUM(N,-N,F,HLNTH))/PI
      Z(N)=0.25*PI*F(N)-(0.5*HLNTH(2*N)*(F(-N)-F(N))+
     +                        SUM(N,N,F,HLNTH))/PI
      DO 10 I=-N+1,N-1
         Z(I)=0.25*PI*F(I)-(0.5*HLNTH(I+N)*(F(-N)-F(I))+
     +                        0.5*HLNTH(N-I)*(F(N)-F(I))+
     +                                   SUM(N,I,F,HLNTH))/PI
   10 CONTINUE
      SCHF=ZOUD(M)/Z(M)
      DO 20 I=-N,M
         Z(I)=SCHF*Z(I)
   20 CONTINUE
C     kijk of aan stopcriterium voldaan wordt
      TWNORM=0.0
      TWNZ=0.0
      DO 30 I=-N,N
         TWNORM=TWNORM+(Z(I)-ZOUD(I))**2
         TWNZ=TWNZ+(ZOUD(I))**2
   30 CONTINUE
      TWNORM=SQRT(TWNORM)
      TWNZ=SQRT(TWNZ)
      IF ((ABS(SCHF-1.0).LE.1.0E-6).AND.((TWNORM/TWNZ).LE.ERRREL)) THEN
      KLAAR=1
      ENDIF
      RETURN
      END
C
C     ---------------------------------------------------------------
C
      REAL FUNCTION SUM(N,M,F,K)
C
C     deze functie is een hulpfunctie voor de vorige routine om te
C     helpen met het berekenen van de integraal.
C
      IMPLICIT  NONE
      INTEGER   NMIN,NMAX,DNMAX
      PARAMETER (NMIN=-250,NMAX=250,DNMAX=500)
      INTEGER   N,M,TELLER
      REAL      HULP
      REAL      F(NMIN:NMAX),K(DNMAX)
C
      INTRINSIC ABS
C
      HULP=0.0
      DO 10 TELLER=-N+1,N-1
         IF (TELLER.NE.M) THEN
         HULP=HULP+K(ABS(TELLER-M))*(F(TELLER)-F(M))
         ENDIF
   10 CONTINUE
      SUM=HULP
      RETURN
      END
C
C     ---------------------------------------------------------------
C
      SUBROUTINE BERSNEL(Z,F,N,C)
C
```

```
C     deze routine berekent de nieuwe dfi(F) gegeven zeta(Z)
C
      IMPLICIT  NONE
      INTEGER   NMIN,NMAX
      REAL      PI
      PARAMETER (NMIN=-250,NMAX=250,PI=3.141592653590)
      INTEGER   N,TELLER
      REAL      Z(NMIN:NMAX),F(NMIN:NMAX)
      REAL      C,BETA
C
      BETA=4/(PI*(C**2))
      DO 20 TELLER=-N,N
         F(TELLER)=BETA*Z(TELLER)*(1+1.5*Z(TELLER))
   20 CONTINUE
      RETURN
      END
C
C     ------------------------------------------------------------
C
      SUBROUTINE TERUG(RO,Z,F,N,H,C)
C
C     deze routine zorgt dat de oplossing van het probleem (gesteld in
C     p-variabele) wordt terugvertaald naar de dimensieloze plaatsvari-
C     abele X.
C
      IMPLICIT  NONE
      INTEGER   NMIN,NMAX
      REAL      PI
      PARAMETER (NMIN=-250,NMAX=250,PI=3.141592653590)
      INTEGER   N,TELLER,T1,T2
      REAL      H,C
      REAL      RO(NMIN:NMAX),Z(NMIN:NMAX),F(NMIN:NMAX)
C
      RO(0)=0.0
      DO 10 T1=1,N
         RO(T1)=T1+0.5*(Z(0)+Z(T1))
         RO(-T1)=T1+0.5*(Z(0)+Z(-T1))
         DO 20 T2=1,T1-1
            RO(T1)=RO(T1)+Z(T2)
            RO(-T1)=RO(-T1)+Z(-T2)
   20    CONTINUE
         RO(T1)=4*H*RO(T1)/PI
         RO(-T1)=-4*H*RO(-T1)/PI
   10 CONTINUE
      DO 25 TELLER=-N,N
         F(TELLER)=Z(TELLER)*(1+1.5*Z(TELLER))/((1+Z(TELLER))*(C**2))
   25 CONTINUE
      RETURN
      END
C
C     ------------------------------------------------------------
C
      SUBROUTINE SCHRIJF(UPB,N,C,ITMAX,M,ERRREL,ST,RO,Z,F)
C
C     deze subroutine schrijft de uitvoer weg naar de file "eenl.uit".
C
      IMPLICIT  NONE
      INTEGER   NMIN,NMAX
      PARAMETER (NMIN=-250,NMAX=250)
```

```
      INTEGER    N,ITMAX,M,ST,TELLER
      REAL       UPB,C,ERRREL
      REAL       RO(NMIN:NMAX),Z(NMIN:NMAX),F(NMIN:NMAX)
C
      OPEN(UNIT=21,FILE='eenl.uit')
      REWIND(21)
      WRITE(21,*) UPB
      WRITE(21,*) N
      WRITE(21,*) C
      WRITE(21,*) ITMAX
      WRITE(21,*) M
      WRITE(21,*) ERRREL
      WRITE(21,*) ST
      DO 10 TELLER=-N,N
         WRITE(21,'(3F10.6)') RO(TELLER),Z(TELLER),F(TELLER)
   10 CONTINUE
      RETURN
      END
```

Ad footnote (7):

```
      PROGRAM ONTWGP
C
C     **************************************************************
C     VERSIE DD 16-1-91
C
C     Dit programma rekent bewegingsvergelijkingen door die de ontwik-
C     keling van een golfprofiel boven een vlakke bodem en de stroom-
C     snelheid aan het vrije oppervlakte beschrijven.
C
C     De gebruiker dient als invoer te geven:
C     (1) de bovengrens van het interval waarop hij wenst te werken
C         (de ondergrens zal altijd 0 zijn).
C         Er wordt aangenomen dat buiten dit interval alles in de
C         rusttoestand is.
C     (2) het aantal roosterpunten op dit interval
C         (op deze roosterpunten wordt in de plaats gediscretiseerd)
C     (3) de tijdstap
C         (dit is de stap die voor de integratie in de tijd genomen
C          wordt)
C     (4) het maximaal aantal stappen dat in de tijd gemaakt wordt.
C         (na dit aantal stappen zal het programma zeker stoppen of
C          al gestopt zijn. Dit laatste zal het geval zijn als er
C          niet meer aan de voorwaarde wordt voldaan dat buiten het
C          werkinterval alles in de rusttoestand is.)
C     (5) het aantal te plotten golfprofielen (deler van (4))
C     (6) de begintoestand voor het golfprofiel en de snelheid aan het
C         oppervlakte in de roosterpunten op tijdstip t=0
C
C     **************************************************************
C
C
C
C     **************************************************************
C     declaraties
C     **************************************************************
      IMPLICIT  NONE
      INTEGER   NMAX,TMAX
      PARAMETER (NMAX=500,TMAX=200)
      INTEGER   N,T,I,J,AANTPL,K,PLAANT
      REAL      UPB,TSTAP,XSTAP,BODEM,BERMAS,BERKIN,BERPOT,TR,UPBM
      REAL      ROOS(0:NMAX),ZETA(0:NMAX,0:TMAX),V(0:NMAX,0:TMAX),
     +          HZO(0:NMAX),HZN(0:NMAX),HVO(0:NMAX),HVN(0:NMAX),
     +          HEO(0:NMAX),HEN(0:NMAX),HZT(0:NMAX),BOD(0:NMAX),
     +          MAS(0:TMAX),ENER(0:TMAX),KINENER(0:TMAX),
     +          POTENER(0:TMAX),HULPIT(0:TMAX)
C
      INTRINSIC ABS,MOD
C
C     **************************************************************
C     hoofdprogramma
C     **************************************************************
      CALL LEESIN(UPB,N,TSTAP,T,AANTPL,HZN,HVN)
      CALL MEQDR(ROOS,UPB,N,XSTAP)
      DO 5 J=0,N
         ZETA(J,0)=HZN(J)
```

```
         V(J,0)=HVN(J)
         BOD(J)=BODEM(ROOS(J))
         HEN(J)=HZN(J)-BOD(J)
  5   CONTINUE
      PLAANT=0
      MAS(0)=BERMAS(N,HZN,XSTAP)
      KINENER(0)=BERKIN(N,HVN,HEN,XSTAP)
      POTENER(0)=BERPOT(N,HZN,XSTAP)
      ENER(0)=KINENER(0)+POTENER(0)
      DO 10 I=1,T
         DO 15 J=0,N
            HZO(J)=HZN(J)
            HVO(J)=HVN(J)
            HEO(J)=HEN(J)
 15      CONTINUE
         IF (ABS(HZO(0)).GE.(1.0E-4)) THEN
         GOTO 25
         ELSEIF (ABS(HZO(N)).GE.(1.0E-4)) THEN
         GOTO 25
         ELSEIF (ABS(HVO(0)).GE.(1.0E-4)) THEN
         GOTO 25
         ELSEIF (ABS(HVO(N)).GE.(1.0E-4)) THEN
         GOTO 25
         ELSE
         CALL BERZETA(N,HZO,HVO,HEO,HZN,HEN,HZT,XSTAP,TSTAP)
         CALL BERV(N,HVO,HZO,HEO,HZT,HVN,XSTAP,TSTAP)
         IF (MOD(I,T/AANTPL).EQ.0) THEN
         PLAANT=PLAANT+1
         K=I*AANTPL/T
         MAS(K)=BERMAS(N,HZN,XSTAP)
         KINENER(K)=BERKIN(N,HVN,HEN,XSTAP)
         POTENER(K)=BERPOT(N,HZN,XSTAP)
         ENER(K)=KINENER(K)+POTENER(K)
         DO 35 J=0,N
            ZETA(J,K)=HZN(J)
            V(J,K)=HVN(J)
 35      CONTINUE
         ENDIF
         ENDIF
 10   CONTINUE
 25   IF (T.GT.I) THEN
      T=I-1
      AANTPL=PLAANT
      ENDIF
      CALL SCHRIJF(UPB,N,TSTAP,T,AANTPL,ZETA,V,MAS,ENER,POTENER,
     +                                               KINENER)
C
C     ga nu plaatjes maken
C
      CALL GINO
C     CALL HPGTER
      CALL PLOTDEVICE(3)
      CALL HP7550
      CALL AXISCA(3,10,0.0,UPB,1)
      CALL AXISCA(3,12,-1.0,11.0,2)
      CALL AXIDRA(1,1,1)
C     CALL AXIDRA(-1,-1,2)
      CALL AXNSTR('x/depth',4.0,-1,0)
      CALL GRACUR(ROOS,BOD,N+1)
```

```
      DO 20 I=0,AANTPL
         DO 30 J=0,N
            HEN(J)=ZETA(J,I)+(10.0/AANTPL)*I
 30      CONTINUE
         CALL GRACUR(ROOS,HEN,N+1)
 20   CONTINUE
      CALL DEVEND
C
C     maak massa en energieplaatjes
C
C     CALL HPGTER
C     CALL PICCLE
      CALL PLOTDEVICE(3)
      CALL HP7550
      TR=T
      UPBM=MAS(0)+0.1
      CALL AXISCA(3,10,0.0,TR,1)
      CALL AXISCA(3,10,0.0,UPBM,2)
      CALL AXIDRA(-1,-1,2)
      CALL AXNSTR('iteration step',4.0,-1,0)
      DO 70 I=0,AANTPL
            HULPIT(I)=I*T/AANTPL
 70   CONTINUE
      CALL GRACUR(HULPIT,MAS,AANTPL+1)
      CALL BROKEN(1)
      CALL GRACUR(HULPIT,POTENER,AANTPL+1)
      CALL BROKEN(3)
      CALL GRACUR(HULPIT,KINENER,AANTPL+1)
      CALL BROKEN(5)
      CALL GRACUR(HULPIT,ENER,AANTPL+1)
      CALL AXIDRA(2,1,1)
      CALL DEVEND
      CALL GINEND
      STOP
      END
C
C     ****************************************************************
C     subroutines en functies
C     ****************************************************************
      SUBROUTINE LEESIN(UPB,N,DT,T,AANT,Z,V)
C
C     deze routine leest de invoer. Achtereenvolgens zijn dat de boven-
C     grens van het interval UPB, het aantal roosterpunten N, de tijd-
C     stap DT, het maximale aantal stappen in de tijd T,
C     het aantal te plotten golfprofielen en de begin-
C     toestanden Z (golfprofiel) en V (snelheid aan het oppervlakte)
C     in de roosterpunten. De invoer dient in de file 'ontwgp.in' te
C     staan.
C
      IMPLICIT  NONE
      INTEGER   NMAX
      PARAMETER (NMAX=500)
      INTEGER   N,I,T,AANT
      REAL      UPB,DT
      REAL      Z(0:NMAX),V(0:NMAX)
C
      OPEN(UNIT=18,FILE='ontwgp.in')
      READ (18,*) UPB
      READ (18,*) N
```

```
      READ (18,*) DT
      READ (18,*) T
      READ (18,*) AANT
      DO 10 I=0,N
         READ (18,*) Z(I),V(I)
   10 CONTINUE
      RETURN
      END
C
C     --------------------------------------------------------------
C
      SUBROUTINE MEQDR(RO,UPB,N,DX)
C
C     deze routine maakt op [0,UPB] een equidistant rooster RO(0) t/m
C     RO(N). Verder wordt in DX de afstand tussen twee roosterpunten
C     opgeleverd.
C
      IMPLICIT   NONE
      INTEGER    NMAX
      PARAMETER (NMAX=500)
      INTEGER    N,I
      REAL       UPB,DX
      REAL       RO(0:NMAX)
C
      DX=UPB/N
      RO(0)=0.0
      DO 10 I=1,N
         RO(I)=I*DX
   10 CONTINUE
      RETURN
      END
C
C     --------------------------------------------------------------
C
      REAL FUNCTION BODEM(X)
C
C     de functie geeft de positie van de bodem op plaats X t.o.v. het
C     y=0-nivo aan.
C
      IMPLICIT   NONE
      REAL       X
C
      BODEM=-1.0
      RETURN
      END
C
C     --------------------------------------------------------------
C
      REAL FUNCTION BERMAS(N,Z,H)
C
C     deze functie berekent de integraal van Z (zeta) over het interval
C     [0,UPB] met de gerepeteerde trapeziumregel. Deze grootheid moet
C     constant zijn in de tijd.
C
      IMPLICIT   NONE
      INTEGER    NMAX
      PARAMETER (NMAX=500)
      INTEGER    N,I
      REAL       H,SUM
```

```
      REAL        Z(0:NMAX)
C
      SUM=0.0
      DO 10 I=0,N
         IF ((I.EQ.0).OR.(I.EQ.N)) THEN
         SUM=SUM+0.5*H*Z(I)
         ELSE
         SUM=SUM+H*Z(I)
         ENDIF
   10 CONTINUE
      BERMAS=SUM
      RETURN
      END
C
C     ---------------------------------------------------------------
C
      REAL FUNCTION BERKIN(N,V,E,H)
C
C     deze functie berekent de kinetische energie als functie van V en
C     E (eta). N is weer het aantal roosterpunten en H de stapgrootte
C     in de plaats.
C
      IMPLICIT   NONE
      INTEGER    NMAX
      REAL       PI
      PARAMETER  (NMAX=500,PI=3.141592653590)
      INTEGER    N,I,J
      REAL       H,TUSSEN
      REAL       V(0:NMAX),E(0:NMAX),TAB1(0:NMAX),TAB2(0:NMAX)
C
      INTRINSIC LOG,TANH,ABS
C
      TUSSEN=0.0
      DO 10 I=0,N
         IF ((I.EQ.0).OR.(I.EQ.N)) THEN
         TUSSEN=TUSSEN+0.25*H*(E(I)*(V(I)**2))
         ELSE
         TUSSEN=TUSSEN+0.5*H*(E(I)*(V(I)**2))
         ENDIF
   10 CONTINUE
      TAB1(0)=0.0
      TAB2(0)=0.0
      DO 20 I=1,N
         TAB1(I)=0.5*H*((1.0)/E(I-1)+(1.0)/E(I))
         TAB2(I)=TAB1(I)+TAB2(I-1)
   20 CONTINUE
      DO 30 I=0,N
         DO 50 J=0,N
            IF (J.LT.I) THEN
            TAB2(J)=TAB2(J)+TAB1(I)
            ELSEIF (J.EQ.I) THEN
            TAB2(J)=0.0
            ELSE
            TAB2(J)=TAB2(J)-TAB1(I)
            ENDIF
   50    CONTINUE
         DO 40 J=0,N
            IF (J.EQ.I) THEN
            TUSSEN=TUSSEN
```

```
          ELSEIF (((J.EQ.0).AND.(I.EQ.N)).OR.((J.EQ.N).AND.(I.EQ.0)))
     +    THEN
          TUSSEN=TUSSEN-0.0625*(1.0/PI)*(H**2)*(V(I)-E(J)*V(J)/E(I))*
     +         (V(J)-E(I)*V(I)/E(J))*LOG(TANH(0.25*PI*ABS(TAB2(J))))
          ELSEIF ((((J.EQ.0).OR.(J.EQ.N)).OR.(I.EQ.0)).OR.(I.EQ.N))
     +    THEN
          TUSSEN=TUSSEN-0.125*(1.0/PI)*(H**2)*(V(I)-E(J)*V(J)/E(I))*
     +         (V(J)-E(I)*V(I)/E(J))*LOG(TANH(0.25*PI*ABS(TAB2(J))))
          ELSE
          TUSSEN=TUSSEN-0.25*(1.0/PI)*(H**2)*(V(I)-E(J)*V(J)/E(I))*
     +         (V(J)-E(I)*V(I)/E(J))*LOG(TANH(0.25*PI*ABS(TAB2(J))))
          ENDIF
 40     CONTINUE
 30   CONTINUE
      BERKIN=TUSSEN
      RETURN
      END
C
C     ----------------------------------------------------------------
C
      REAL FUNCTION BERPOT(N,Z,H)
C
C     deze functie berekent de potentiele energie als functie van Z
C     (zeta). N is weer het aantal roosterpunten en H de stapgrootte
C     in de plaats.
C
      IMPLICIT   NONE
      INTEGER    NMAX
      PARAMETER (NMAX=500)
      INTEGER    N,I
      REAL       H,TUSSEN
      REAL       Z(0:NMAX)
C
      TUSSEN=0.0
      DO 10 I=0,N
        IF ((I.EQ.0).OR.(I.EQ.N)) THEN
        TUSSEN=TUSSEN+0.25*H*((Z(I)**2))
        ELSE
        TUSSEN=TUSSEN+0.5*H*((Z(I)**2))
        ENDIF
 10   CONTINUE
      BERPOT=TUSSEN
      RETURN
      END
C
C     ----------------------------------------------------------------
C
      SUBROUTINE BERZETA(N,ZOUD,V,ETAOUD,ZNEW,ETANEW,ZT,H,DT)
C
C     deze routine berekent voor de gegeven ZOUD,V en ETAOUD de ZT, en
C     ZNEW, ETANEW op tijdstip DT later. De N staat weer voor het aan-
C     tal roosterpunten en H is de stapgrootte in de plaats.
C
      IMPLICIT   NONE
      INTEGER    NMAX
      REAL       PI
      PARAMETER (NMAX=500,PI=3.141592653590)
      INTEGER    N,I,J
      REAL       H,TWEEH,DT
```

```
      REAL        TAB1(0:NMAX),TAB2(0:NMAX),ZOUD(0:NMAX),V(0:NMAX),
     +            ETAOUD(0:NMAX),ZNEW(0:NMAX),ZT(0:NMAX),ETANEW(0:NMAX),
     +            INT(0:NMAX)
C
      INTRINSIC LOG,TANH,ABS
C
      TWEEH=2.0*H
      TAB1(0)=0.0
      TAB2(0)=0.0
      DO 10 I=1,N
         TAB1(I)=0.5*H*((1.0/ETAOUD(I-1))+(1.0/ETAOUD(I)))
         TAB2(I)=TAB1(I)+TAB2(I-1)
   10 CONTINUE
      DO 20 I=0,N
         INT(I)=-PI*ETAOUD(I)*V(I)
         DO 30 J=0,N
            IF (J.LT.I) THEN
            TAB2(J)=TAB2(J)+TAB1(I)
            ELSEIF (J.EQ.I) THEN
            TAB2(J)=0.0
            ELSE
            TAB2(J)=TAB2(J)-TAB1(I)
            ENDIF
            IF (J.EQ.I) THEN
            INT(I)=INT(I)
            ELSE
              IF ((J.EQ.0).OR.(J.EQ.N)) THEN
              INT(I)=INT(I)+0.5*(V(J)-ETAOUD(I)*V(I)/ETAOUD(J))*
     +                 LOG(TANH(PI*0.25*ABS(TAB2(J))))*H
              ELSE
              INT(I)=INT(I)+(V(J)-ETAOUD(I)*V(I)/ETAOUD(J))*
     +                 LOG(TANH(PI*0.25*ABS(TAB2(J))))*H
              ENDIF
            ENDIF
   30    CONTINUE
         INT(I)=INT(I)/PI
   20 CONTINUE
      ZT(0)=(INT(1)-INT(0))/H
      DO 40 I=1,N-1
         ZT(I)=(INT(I+1)-INT(I-1))/TWEEH
   40 CONTINUE
      ZT(N)=(INT(N)-INT(N-1))/H
      DO 50 I=0,N
         ZNEW(I)=ZOUD(I)+DT*ZT(I)
         ETANEW(I)=ETAOUD(I)+ZNEW(I)-ZOUD(I)
   50 CONTINUE
      RETURN
      END
C
C     -----------------------------------------------------------------
C
      SUBROUTINE BERV(N,VOUD,Z,ETA,ZT,VNEW,H,DT)
C
C     deze routine berekent m.b.v. Z,ZT,VOUD en ETA de VT en daarmee de
C     V op tijdstip DT later. N geeft weer het aantal roosterpunten aan
C     en H de stapgrootte in de plaats.
C
      IMPLICIT   NONE
C
```

```
      INTEGER    NMAX
      REAL       PI
      PARAMETER (NMAX=500,PI=3.141592653590)
      INTEGER    N,I,J
      REAL       H,TWEEH,DT
      REAL       VOUD(0:NMAX),Z(0:NMAX),ETA(0:NMAX),ZT(0:NMAX),
     +           VNEW(0:NMAX),ZX(0:NMAX),ETAX(0:NMAX),VT(0:NMAX),
     +           HULP(0:NMAX),TAB1(0:NMAX)
C
      TWEEH=2.0*H
      TAB1(0)=0.0
      DO 10 I=1,N
         TAB1(I)=0.5*(VOUD(I)*ETA(I)*ZT(I)+
     +                VOUD(I-1)*ETA(I-1)*ZT(I-1))*H
 10   CONTINUE
      ZX(0)=Z(1)/TWEEH
      ETAX(0)=(ETA(1)-1.0)/TWEEH
      DO 20 I=1,N-1
         ZX(I)=(Z(I+1)-Z(I-1))/TWEEH
         ETAX(I)=(ETA(I+1)-ETA(I-1))/TWEEH
 20   CONTINUE
      ZX(N)=-Z(N-1)/TWEEH
      ETAX(N)=(1.0-ETA(N-1))/TWEEH
      DO 30 I=0,N
         VT(I)=-ZX(I)+VOUD(I)*ZT(I)/ETA(I)
         HULP(I)=ETAX(I)/(ETA(I)**3)
         DO 40 J=1,N
            IF (J.LE.I) THEN
            VT(I)=VT(I)-TAB1(J)*HULP(I)
            ELSE
            VT(I)=VT(I)+TAB1(J)*HULP(I)
            ENDIF
 40      CONTINUE
 30   CONTINUE
      DO 50 I=0,N
         VNEW(I)=VOUD(I)+DT*VT(I)
 50   CONTINUE
      RETURN
      END
C
C     ---------------------------------------------------------------
C
      SUBROUTINE SCHRIJF(UPB,N,DT,T,AANT,Z,V,MASSA,ENERG,POT,KIN)
C
C     deze routine schrijft de uitvoer naar de file 'ontwgp.uit'. Ach-
C     tereenvolgens zijn dat de ingevoerde bovengrens UPB, het aantal
C     roosterpunten N, de tijdstap DT, het gemaakte aantal stappen in
C     de tijd en de geplotte Z en V-vectoren. Verder worden ook
C     de berekende grootheden massa en energie weggeschreven.
C
      IMPLICIT  NONE
      INTEGER    NMAX,TMAX
      PARAMETER (NMAX=500,TMAX=200)
      INTEGER    N,I,J,AANT,T
      REAL       UPB,DT
      REAL       Z(0:NMAX,0:TMAX),V(0:NMAX,0:TMAX),MASSA(0:TMAX),
     +           ENERG(0:TMAX),POT(0:TMAX),KIN(0:TMAX)
C
      OPEN(UNIT=19,FILE='ontwgp.uit')
```

```
      REWIND (19)
      WRITE (19,*) UPB
      WRITE (19,*) N
      WRITE (19,*) DT
      WRITE (19,*) T
      DO 10 I=0,AANT
         DO 20 J=0,N
            WRITE (19,'(I4,2F10.6)') J,Z(J,I),V(J,I)
20       CONTINUE
         WRITE (19,*) '----------------------------'
10    CONTINUE
      DO 30 I=0,AANT
         WRITE (19,'(4F10.6)') MASSA(I),ENERG(I),POT(I),KIN(I)
30    CONTINUE
      CLOSE(19)
      RETURN
      END
```