****

**CE-580**

**COMPUTATIONAL TECHNIQUES**

**FOR**

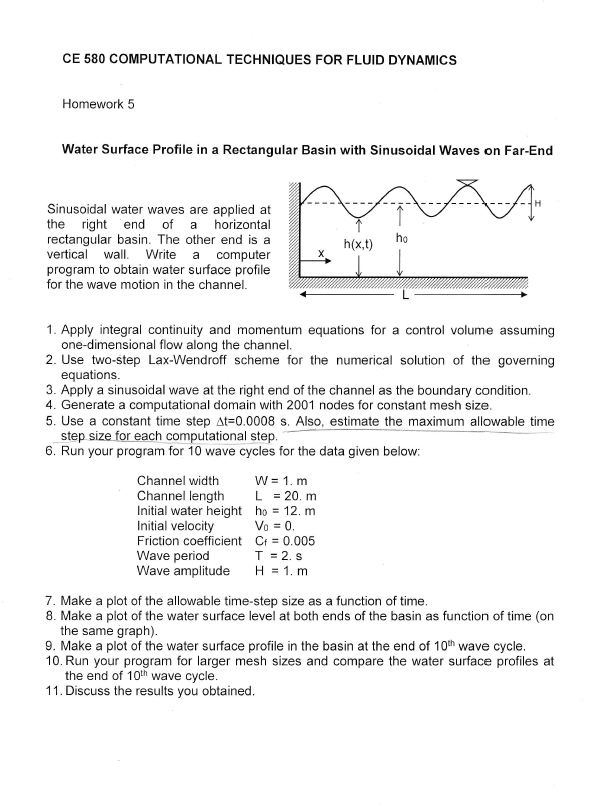
**FLUID DYNAMICS**

**HOMEWORK #5**

**Numerical Solution to Water Waves**

**Taha Yaşar Demir**

**1881978**



# **Computations**

Consider the control volume shown in figure 1

|  |
| --- |
|  |
| Figure 1: Control Volume |

***Assumptions:***

* Quasi-One-dimensional flow
* Velocity is uniform across vertically
* Incompressible fluid

***Governing Equations***

The integral continuity equation

Applied to above control volume

Note that, since the dependent variables above is not dependent on x we can take time derivative inside

Regrouping the continuity and dividing by

As goes to 0, continuity equation becomes

The integral momentum equation in x-direction is

Applying to the CV gives

The friction force can be related to the shear stress, and it also depends on flow direction

Where is the shear stress and is the wetted perimeter.

The wall shear stress can be expressed in terms of a friction coefficient based on the local cross sectional average velocity

Substituting into the momentum equation and dividing by

As goes to 0 yields momentum equation

The continuity equation (1) and momentum equation (2) represent the two equations to be solved to yield the water surface profile and the velocity distribution. For convenience we will introduce new variables for the parametric appearing in the equations

In terms of new variable, the equations simplify to

***Grid System***

|  |
| --- |
|  |
| Figure 2: Grid System |

***Discretization***

Two step Lax-Wendrof Method is used for numerical solution of these equations

The continuity equation

Solving for

The momentum equation

Discretized as

At completion of this step, one calculates velocity and other variables

The second step proceeds from n to n+1 using the variables at the intermediate step to evaluate the spatial derivatives. This is known as a ‘leapfrog’ step and suggests a central difference formulation for derivatives

The second step for continuity

And the second step for momentum gives

And then

***Boundary Conditions***

At two ends different discretization is necessary, since central difference on A and B is not possible

At the wall, i=1, forward differencing is used to obtain at intermediate time step

For second step to n+1i the values for h at the wall is

At the reservoir side,I=M, h is determined from input wave function

And A is determined such as

There is a definite limit on the magnitude of time step . Any small disturbance that occurs in the basin, such as local change in elevation of the water surface, will travel in both directions as a wave with velocity with respect to the water. If in addition the water in the base basin is moving with velocity u, the wave will travel at velocity with respect to the basin

In order to represent the physical problem correctly, the time step corresponding to the grid spacing must lie within the zone of influence

This requires that

***Solution Parameters***

* Channel width W = 1.0 m
* Channel length L = 20 m
* Initial water height
* Initial Velocity
* Friction coefficient
* Wave period
* Wave Amplitude = 1 m
* Gravitational acceleration
* Time step
* Solve for 10 cycles

# **Results and Discussion**

|  |
| --- |
|  |
| Figure 3: Allowable Time Step at Each Iteration (Time) |

In order to obey CFL condition for stability, time step of 0.0008 second is used in calculations. As seen from figure 3, maximum allowable time step on each iteration is slightly larger than 0.0008 seconds. A maximum time step can be calculated and used for each iteration to get a faster convergence. In addition, it is necessary to perform this operation when the time step is unknown, or to be determined.

|  |
| --- |
|  |
| Figure 4: Height of Water at both ends of Domain |

The water height is at the sea-side(reservoir side) is taken a sine function with wave amplitude of 1 m and period of 2 s. It is assumed that the reservoir is so strong that wont be affected from the reflected waves. The result is typical sine function.

On the other hand, from figure 4, The water height at the wall side is strongly affected from incoming and reflected outgoing waves. So that, water amplitude can reach higher and lower levels than the source. Note that there is no disturbance on water level until 2nd second. Which means the information from the reservoir reaches to wall side in 2 seconds. This confirms that the additional term in the CFL condition.

|  |
| --- |
|  |
| Figure 5: Surface profile at the end of 10th cycle |

|  |
| --- |
|  |
| Figure 6: Surface Profiles with Different Grid Numbers |

In order to prove that grid resolution is good enough, some different grid numbers are tried. 1001, 2001, 4001 and 8001 divisions are used with time step of 0.0016, 0.0008, 0.0004, and 0.0002 accordingly to stay in the stability region.

According to figure 6 water surface profiles are more or less the same, but with 1001 divisions there are high fluctuations in surface. And the fluctuations are getting smaller with increased grid numbers. With 8001 divisions and time step of 0.0002 s the fluctuations are completely removed, and solution is acceptable. The slight difference in the surface profiles is caused by these fluctuations, and represents erroneous solutions.

# Source Code

|  |
| --- |
| program WaterWaves  c..Taha Yaşar Demir /1881978  c..CE-580 - Homework #5  parameter(mx=20001)  common/para/ dx,dt,W,Pw(mx),rL,Cf,T,cycles,T\_max,Hi,Vi,g,time  common/flow/ u(mx),h(mx),a(mx),b(mx),c(mx),Pwt(mx),pi,N  common/midstep/ Utmp(mx),Htmp(mx),Atmp(mx),Btmp(mx),Ctmp(mx)  open(11,file='cfl.dat')  open(12,file='h.dat')  open(13,file='surface.dat')  call init  time = 0.  do while(time.le.T\_max)  call output(1)  time = time + 0.5\*dt  call mid\_continuity  call mid\_momentum  time = time + 0.5\*dt  call continuity  call momentum  print\*, time  end do  call output(2)  close(11)  close(12)  close(13)  stop  end  c-----------------------------------------------------------------------  subroutine init  parameter(mx=20001)  common/para/ dx,dt,W,Pw(mx),rL,Cf,T,cycles,T\_max,Hi,Vi,g,time  common/flow/ u(mx),h(mx),a(mx),b(mx),c(mx),Pwt(mx),pi,N  common/midstep/ Utmp(mx),Htmp(mx),Atmp(mx),Btmp(mx),Ctmp(mx)  pi = 22./7.  T = 2. ! s  cycles = 10.  T\_max = T\*cycles ! s  dt = 0.0008 ! s Change it according to grid number  W = 1. ! m - width of channel  rL = 20. ! m - lenght of channel  Hi = 12. ! m - initial water heigth  Vi = 0. ! m/s -initial velocity  N = 2001 ! grid number / Try other Numbers  Cf = 0.005  g = 9.81 ! N.m/s^2 gravitational acceleration  dx = rL/(N-1)  do i=1,N  Pw(i)= 2\*h(i) + W ! no need  u(i) = Vi  h(i) = Hi  a(i) = u(i)\*h(i)  b(i) = h(i)\*u(i)\*\*2 + 0.5\*g\*h(i)\*\*2 ! no need  c(i) = (Cf\*u(i)\*abs(u(i))\*Pw(i))/(2\*W) ! no need  enddo  return  end  c-----------------------------------------------------------------------  subroutine mid\_continuity  parameter(mx=20001)  common/para/ dx,dt,W,Pw(mx),rL,Cf,T,cycles,T\_max,Hi,Vi,g,time  common/flow/ u(mx),h(mx),a(mx),b(mx),c(mx),Pwt(mx),pi,N  common/midstep/ Utmp(mx),Htmp(mx),Atmp(mx),Btmp(mx),Ctmp(mx)  Htmp(N) = Hi + 0.5\*sin(2\*pi\*time/T) ! Resorvoir side is input wave function  Htmp(1) = h(1) - (0.5\*dt/dx)\*(a(2)-a(1)) ! forward difference for first node  do i=2,N-1  Htmp(i) = 0.5\*(h(i+1)+h(i-1)) - (0.25\*dt/dx)\*(a(i+1)-a(i-1))  c print\*, i,Htmp(i),h(i+1),h(i-1),a(i+1),a(i-1)  enddo  return  end  c-----------------------------------------------------------------------  subroutine mid\_momentum  parameter(mx=20001)  common/para/ dx,dt,W,Pw(mx),rL,Cf,T,cycles,T\_max,Hi,Vi,g,time  common/flow/ u(mx),h(mx),a(mx),b(mx),c(mx),Pwt(mx),pi,N  common/midstep/ Utmp(mx),Htmp(mx),Atmp(mx),Btmp(mx),Ctmp(mx)  Atmp(1) = 0. ! boundary condition  Utmp(1) = 0. ! boundary condition  Btmp(1) = 0.5\*g\*Htmp(1)\*\*2  Pwt(1) = 2\*Htmp(1) + W  Ctmp(1) = 0.  Atmp(N) = a(N) - (0.5\*dt/dx)\*(b(N)-b(N-1)) - 0.5\*dt\*c(N)  Utmp(N) = Atmp(N) / Htmp(N)  Btmp(N) = Htmp(N)\*Utmp(N)\*\*2 + 0.5\*g\*Htmp(N)\*\*2  Pwt(N) = 2\*Htmp(N) + W  Ctmp(N) = (Cf\*Utmp(N)\*abs(Utmp(N))\*Pwt(N))/(2\*W)  c print\*, time,Ctmp(N),Utmp(N),Btmp(N),Htmp(N)  do i=2,N-1  Atmp(i) = 0.5\*(a(i+1)+a(i-1))-(0.25\*dt/dx)\*(b(i+1)-b(i-1))  + -0.5\*dt\*c(i)  Utmp(i) = Atmp(i)/Htmp(i)  Btmp(i) = Htmp(i)\*Utmp(i)\*\*2 + 0.5\*g\*Htmp(i)\*\*2  Pwt(i) = 2\*Htmp(i) + W  Ctmp(i) = (Cf\*Utmp(i)\*abs(Utmp(i))\*Pwt(i))/(2\*W)  c print\*, time,i,Atmp(i),Utmp(i),Btmp(i),Ctmp(i)  enddo  return  end  c-----------------------------------------------------------------------  subroutine continuity  parameter(mx=20001)  common/para/ dx,dt,W,Pw(mx),rL,Cf,T,cycles,T\_max,Hi,Vi,g,time  common/flow/ u(mx),h(mx),a(mx),b(mx),c(mx),Pwt(mx),pi,N  common/midstep/ Utmp(mx),Htmp(mx),Atmp(mx),Btmp(mx),Ctmp(mx)  h(1) = h(1) - (dt/dx)\*(Atmp(2)-Atmp(1))  do i=2,N-1  h(i) = h(i) - (0.5\*dt/dx)\*(Atmp(i+1)-Atmp(i-1))  enddo  h(N) = Hi + 0.5\*sin(2\*pi\*time/T)  return  end  c-----------------------------------------------------------------------  subroutine momentum  parameter(mx=20001)  common/para/ dx,dt,W,Pw(mx),rL,Cf,T,cycles,T\_max,Hi,Vi,g,time  common/flow/ u(mx),h(mx),a(mx),b(mx),c(mx),Pwt(mx),pi,N  common/midstep/ Utmp(mx),Htmp(mx),Atmp(mx),Btmp(mx),Ctmp(mx)  a(1) = 0.  u(1) = 0.  b(1) = 0.5\*g\*h(1)\*\*2  c(1) = 0.  Pw(1) = 2\*h(1) + W  a(N) = a(N) - (dt/dx)\*(Btmp(N)-Btmp(N-1)) - dt\*Ctmp(N)  u(N) = a(N)/h(N)  b(N) = h(N)\*u(N)\*\*2 + 0.5\*g\*h(N)\*\*2  Pw(N)= 2\*h(N) + W  c(N) = (Cf\*u(N)\*abs(u(N))\*Pw(N-1))/(2\*W)  c print\*, time,a(N),b(N),u(N),h(N),Btmp(N),Btmp(N-1)  do i=2,N-1  a(i) = a(i) - (0.5\*dt/dx)\*(Btmp(i+1)-Btmp(i-1)) - dt\*Ctmp(i)  c print\*, Btmp(i+1),Btmp(i-1),Ctmp(i)  u(i) = a(i) / h(i)  c print\*, i,time,u(i),a(i),h(i)  b(i) = h(i)\*u(i)\*\*2 + 0.5\*g\*h(i)\*\*2  Pw(i)= 2\*h(i) + W  c(i) = (Cf\*u(i)\*abs(u(i))\*Pw(i))/(2\*W)  enddo    return  end  c-----------------------------------------------------------------------  subroutine output(m)  parameter(mx=20001)  common/para/ dx,dt,W,Pw(mx),rL,Cf,T,cycles,T\_max,Hi,Vi,g,time  common/flow/ u(mx),h(mx),a(mx),b(mx),c(mx),Pwt(mx),pi,N  common/midstep/ Utmp(mx),Htmp(mx),Atmp(mx),Btmp(mx),Ctmp(mx)  real cfl(N),x  do i=1,N  cfl(i) = dx/(abs(u(i))+sqrt(g\*h(i)))  enddo  write(11,\*) time,minval(cfl)  write(12,\*) time,h(1),h(N)  x = 0.  if (m.eq.2) then  do i=1,N  write(13,\*) x,h(i)  x = x + dx  enddo  endif  return  end  c----------------------------------------------------------------------- |
| Fortran Code Used for Calculations |