

## DAFTAR PUSTAKA

- Berthon, C., Cordier, O., & Le, M. (2012). An analytical solution of the shallow water system coupled to the Exner Equation. *Comptes Rendus Mathematique, Elsevier Masson*, 183-186.
- Castro Diaz, M., Fernandez-Nieto, E., & Ferreiro, A. (2008). Sediment Transport Models in Shallow Water Equations and Numerical Approach by High Order Finite Volume Methods. *Computer & Fluids*, 299-316.
- Caviedes-Voullième, D., Morales-Hernández, M. J., Lacasta, A., & García-Navarro, P. (2017). Two-Dimensional Numerical Simulation of Bed-Load Transport of a Finite-Depth Sediment Layer: Applications to Channel Flushing. *Journal Hydraulic Engineering*, 04017034.
- de Gennes, P. (2002). On Fluid/Wall Slippage. *Langmuir*, 3413-3414.
- Goutiere, L., Soares-Fraza, S., & Zech, Y. (2011). Dam-break flow on mobile bed in abruptly widening channel: experimental data. *Journal of Hydraulic Research*, 367-371.
- Gunawan, P., & Lhebrard, X. (2015). Hydrostatic Relaxation Scheme for the 1D Shallow Water - Exner Equations in Bedload Transport. *Computers & Fluids*, 44-50.
- Hoffmann, K., & Chiang, S. (2000). *Computational Fluid Dynamics* (Vol. I). Wichita, Kansas: Engineering Education System.
- Hudson, J., & Sweby, P. (2003). Formulations for Numerically Approximating Hyperbolic Systems Governing Sediment Transport. *Journal of Scientific Computing*, 225-252.
- Juez, C., Murillo, J., & García-Navarro, P. (2014). A 2D Weakly-Coupled and Efficient Numerical Model for Transient Shallow Flow and Movable Bed. *Advances in Water Resources*, 93-109.
- Kowalik, Z., & Murty, T. (1993). *Numerical Modelling of Ocean Dynamics*. Singapore: World Scientific.

- Liu, X., & Beljadid, A. (2017). A coupled numerical model for water flow, sediment transport and bed erosion. *Computers & Fluids*, 273-284.
- MacCormack, R. (1969). The Effect of Viscosity in Hypervelocity Impact Cratering. *AIAA Hypervelocity Impact Conference*, 69-354.
- Magdalena, I., & Pebriansyah, M. (2022). Numerical treatment of finite difference method for solving dam break model on a wet-dry bed withan obsatcle. *Results in Engineering*, 100382.
- Martinez-Aranda, S., Murillo, J., & Garcia-Navarro, P. (2019). A 1D Numerical Model for the Simulation of Unsteady and Highly Erosive FLOws in Rivers. *Computers & Fluids* , 8-13.
- Martinez-Aranda, S., Murillo, J., & Garcia-Navarro, P. (2021). Comparison of new efficient 2D models for the simulation of bedload transport using the augmented Roe approach. *Advances in Water Resources*, 103931.
- Meyer-Peter, E., & Muller, R. (1948). Formulas for bed-load transport. *In: Report on 2nd meeting on international association on hydraulic structures research*, 39-64.
- Said, I., Ribal, A., & Mahie, A. (2017). Numerical investigation of long-wave breaking on even bottom with linear friction. *Far East Journal of Mathematical Sciences*, 595-606.
- Spinewine, B., & Zech, Y. (2007). Small-scale laboratory dam-break waves on movable beds. *Journal of Hydraulic Research*, 77-86.
- Vreugdenhil, C. (1994). *Numerical Methods for Shallow-Water Flow*. Dordrecht: Kluwer Academic Publishers.
- Wu, W. (2007). *Computational River Dynamics*. London: Taylor & Francis.
- Zhang, S., & Duan, J. (2011). 1D finite volume model of unsteady flow over mobile bed. *Journal of Hydrology*, 57-68.

**LAMPIRAN**

## - Sintaks program solusi analitik

```

clc
clear

alpha = 0.005;
beta = 0.005;
gamma = 2;
q = 1;
Ag = 0.01;
g = 9.81;
p = 2/3;
x = 10;
dx = 0.1;
del_x = (x/dx)+1;
t = 10;
for i=1:del_x
    u(i,1) = sqrt(((alpha*((i-1)*dx)+beta)/Ag).^p);
    h(i,1) = q/u(i);
    U(i,1) = u(i).^3;
    z0(i,1) = gamma - ((2*g*q+U(i))/(2*g*u(i)));
    Z0(i,1) = z0(i,1) - alpha*t;
end
X = linspace(0,x,del_x);
% plot (X,h+Z0,'-', 'LineWidth',2, 'DisplayName', "Keadaan Awal h+Z_b
t = "+t+" s")
% hold on
% plot (X,Z0,'-', 'LineWidth',2, 'DisplayName', "Keadaan Awal Z_b t =
"+t+" s")
% hold on
figure (1)
% plot (X,h+Z0,'--k', 'LineWidth',3, 'DisplayName', "Keadaan Awal")
plot (X,h+Z0,'-k', 'LineWidth',3, 'DisplayName', "Solusi Analitik t =
"+t+" s")
grid on
hold on
axis ([0 x 1.75 2])
yticks([1.75 1.8 1.85 1.9 1.95 2])
xticks([0 1 2 3 4 5 6 7 8 9 10])
xlabel ("x (m)", 'FontSize', 24)
ylabel (" h+Z_b (m)", 'FontSize', 24)
set(gca, 'FontSize',24)

figure (2)
% plot (X,Z0,'--k', 'LineWidth',3, 'DisplayName', "Keadaan Awal")
plot (X,Z0,'-k', 'LineWidth',3, 'DisplayName', "Solusi Analitik t =
"+t+" s")
grid on
hold on
axis ([0 x 0.65 1.35])
yticks([.7 .9 1.1 1.3])
xticks([0 1 2 3 4 5 6 7 8 9 10])
xlabel ("x (m)", 'FontSize', 24)
ylabel ("Z_b (m)", 'FontSize', 24)
set(gca, 'FontSize',24)

```

## - Sintaks program metode MacCormack

```

clc
clear

t = 10;%time
g = 9.81;%gravity
x = 10;%space

M = 0;

e = 0.47;%porosity
p = 1/(1-e);
G0 = 0.01;
G1 = 0.01;

alpa1 = 0;%alpha const
beta1 = alpa1/2;
n = 0.0;%friction coef

dx = 0.1;%dx
dt = 0.01;%dt

del_x = (x/dx)+1;%length grid x
del_t = t/dt;%length grid t
C = dt/dx;

q = ones(del_x,2);
u = zeros(del_x,2);
h = zeros(del_x,2);
Z = zeros(del_x,2);

qq = zeros(del_x,1);
uu = zeros(del_x,1);
hh = zeros(del_x,1);
ZZ = zeros(del_x,1);

Q = 1;
Ag = 0.01;
p= 2/3;
alpha = 0.005;
beta = 0.005;
gamma = 2;

for i=1:del_x
    u(i,1) = sqrt(((alpha*((i-1)*dx)+beta)/Ag).^p);
    h(i,1) = Q/u(i);
    U(i) = u(i);
    Z(i,1) = gamma-((2*g*Q+u(i).^3)/(2*g*u(i)));
end

k = 0;
while k<del_t
    %prediktor step
    F = q.*u+0.5*g*h.^2;

```

```

for i = 2:del_x-1
    hh(i,1) = h(i,1) - C*(q(i+1,1)-q(i,1));%mass conserv

    if hh(i,1)>M
        qq(i,1) = q(i,1) - C*(F(i+1,1)-F(i,1))...
            -C*g*h(i,1)*(Z(i+1,1)-Z(i,1));%momentum conserv
        uu(i,1) = qq(i,1)/hh(i,1);%velocity
    else
        qq(i,1) = 0;
        uu(i,1) = 0;
    end
    %G0 = MPM_1D(h(i,1),u(i,1));
    %G1 = MPM_1D(h(i+1,1),u(i+1,1));

    ZZ(i,1) = Z(i,1)-C*(G1*u(i+1,1)^3-G0*u(i,1)^3);
end
%boundary condition
qq(1,1) = 1;
qq(end,1) = 2*qq(end-1,1)-qq(end-2,1);
uu(1,1) = 2*uu(2,1)-uu(3,1);
uu(end,1) = 2*uu(end-1,1)-uu(end-2,1);
hh(1,1) = 2*hh(2,1)-hh(3,1);
hh(end,1) = .5665;
ZZ(1,1) = 2*ZZ(2,1)-ZZ(3,1);
ZZ(end,1) = 2*ZZ(end-1,1)-ZZ(end-2,1);

%korektor step

F = qq.*uu+0.5*g*hh.^2;
for i = 2:del_x-1

    h(i,2) = 0.5*(h(i,1) + hh(i,1) - C*(qq(i,1)-qq(i-1,1)))...
        +(C/dx)*alpha1*(h(i+1,1)-2*h(i,1)+h(i-1,1));%mass
conserv with filter

    if h(i,2)>M
        q(i,2) = 0.5*(q(i,1) + qq(i,1) - C*(F(i,1)-F(i-
1,1))...
            -C*g*hh(i,1)*(ZZ(i,1)-ZZ(i-1,1)));%momentum
conserv
        u(i,2) = q(i,2)/h(i,2);%velocity
    else
        q(i,2) = 0;
        u(i,2) = 0;
    end
    %G0 = MPM_1D(h(i,1),u(i,1));
    %G1 = MPM_1D(h(i-1,1),u(i-1,1));

    Z(i,2) = 0.5*(Z(i,1)+ZZ(i,1)-C*(G0*uu(i,1)^3-G1*uu(i-
1,1)^3))...
        +(C/dx)*alpha1*(Z(i+1,1)-2*Z(i,1)+Z(i-1,1));

end
%boundary condition
q(1,2) = 1;
q(end,2) = 2*q(end-1,2)-q(end-2,2);
u(1,2) = 2*u(2,2)-u(3,2);

```

```

u(end,2) = 2*u(end-1,2)-u(end-2,2);
h(1,2) = 2*h(2,2)-h(3,2);
h(end,2) = .5665;
Z(1,2) = 2*Z(2,2)-Z(3,2);
Z(end,2) = 2*Z(end-1,2)-Z(end-2,2);

%rewrite result
h(:,1) = h(:,2);
q(:,1) = q(:,2);
u(:,1) = u(:,2);
Z(:,1) = Z(:,2);

k = k+1;%part of "while-end" condition
end

X = linspace(0,x,del_x);
figure (1)
plot(X,Z(:,1)+h(:,1), 'vb', 'Linewidth',2, 'DisplayName', "Solusi
MacCormack t = "+t+" s")
axis tight
% axis ([0 x 0. 1.1])
hold on
figure (2)
plot(X,Z(:,1), 'vr', 'Linewidth',2, 'DisplayName', "Solusi MacCormack
t ="+t+" s")
%axis ([0-1 x-1 -0.005 .04])
hold on
axis tight
lgd = legend;
lgd.FontSize = 18;

```