

Miniprojekt 2
Hydrostatiskt tryck vid Runge Falls

Peter Ankerstål

10 oktober 2007

Innehåll

1	Introduktion - DammTryck	3
2	Användargränssnitt	3
3	Körexempel 1	3
3.1	Grafer från körning	4
4	Tolkning av grafer	4
4.1	Tryck	4
4.2	Steglängd	5
4.3	Approximativa felet	5
5	Alternativ Placering	5
A	Källkod	6

1 Introduktion - DammTryck

DammTryck är ett program utvecklat för att på numerisk väg kalkylera det hydrostatiska trycket vid olika vattennivåer vid ett eventuellt dammbygge vid Runge Falls.

Programmet beräknar en integral för trycket med hjälp av den numeriska trapetsmetoden och använder sig av en variabel steglängd för att hålla det beräknade felet under kontroll.

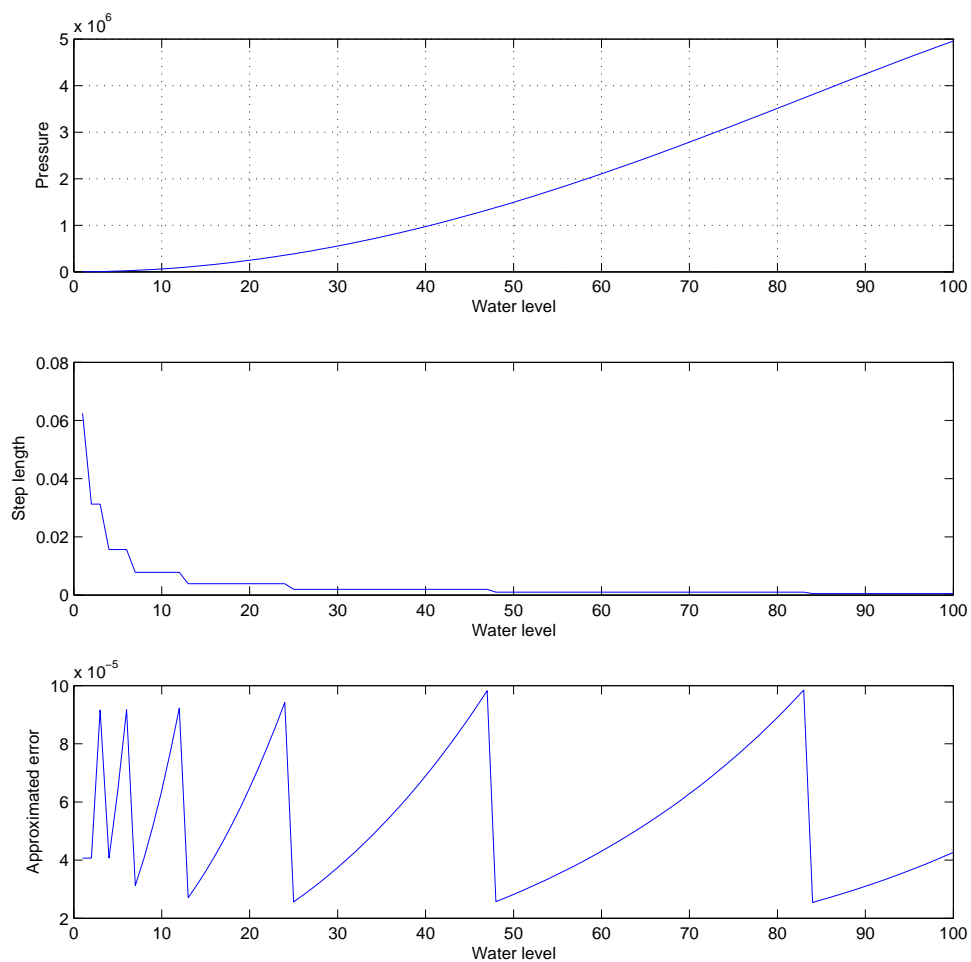
2 Användargränssnitt

Programmet är mycket enkelt och användas och startas (i matlab) genom att skriva kommandot DammTryck, sedan får man välja för vilken vattennivå (i feet) man skall beräkna trycket. Därefter beräknar programmet trycket för vattennivåer från 0ft till angiven nivå och visar resultatet som en graf på skärmen.

3 Körexempel 1

```
>> DammTryck
DammTryck 0.1
This program calculates the Hydrostatic-pressure
at any given water level at Oscillating Gorge
Enter maximum waterlevel to calculate pressure for: 100
Smallest step-length used: 0.00048828
```

3.1 Grafer från körning



Figur 1: Graf från körning av DammTryck

4 Tolkning av grafer

4.1 Tryck

Grafen för trycket som funktion av vattennivån är ganska enkelt att tolka, trycket ökar exponentiellt med vattennivån.

4.2 Steglängd

Steglängdsgrafen visar hur steglängden vid den numeriska integrationen minskar när vattennivån ökar för att kunna hålla det approximativa felet inom rimliga gränser. Anledningen till att steglängden måste minska när vattennivån ökar är att när beräkningarnas storleksordning ökar, ökar också differensen mellan iterationerna, vilket leder till att även det approximativa felet ökar, då felet beräknas som $T(n) - T(n - 1)$ där n är iterationsnummret.

4.3 Approximativa felet

Man ser tydligt att det approximativa felet har en stark koppling till steglängden. Detta beror på att när steglängden halveras så minskar det approximativa felet kraftigt, men sedan kommer felet att börja gå upp igen allteftersom storleksordningen i beräkningarna ökar. Därefter när det beräknade felet till slut är större än det tillåtna halveras steglängden igen och det approximativa felet minskar återigen kraftigt.

5 Alternativ Placering

Det finns en alternativ plats för dammen några kilometer längre uppströms, nära Newton's Mill. Där har Oscillating Gorge en ojämnare form, så att det inte går att uttrycka ravinens vidd som en enkel funktion av höjden. För beskrivningen av ravinens form har man i stället tillgång till några mätningar från en geologisk undersökning.

Det finns åtminstone två sätt att ta sig an det här problemet. Ett sätt är att numeriskt integrera med trapezformeln direkt mot de givna data som vi fått från den geologiska undersökningen. Det andra alternativet kan vara att först försöka finna någon funktion utifrån de data som finns, antingen med minstakvadrat-approximation eller interpolering. Antagligen tjänar man på att använda sig av styckvis interpolering då vanlig interpolation över all mätdata kan orsaka kraftliga oscillationer mellan de punkter som polynomet anpassas till. När detta sedan är gjort kan man integrera de här funktionerna.

Man bör också notera att de data man kommit fram till är nästan linjära på västra sidan, detta är något som man skulle kunna utnyttja vid en ev. kurvanpassning. På östra sidan däremot så förefaller raven vara lite mer ojämn och man bör därför inte använda sig av linjär kurvanpassning i det fallet.

Appendix

A Källkod

```
% Miniprojekt 2.
%
% Author: Peter Ankerstal.
% 850826-xxxx
% Licence: Beerware.
%
%
clc
disp('DammTryck 0.1');
disp('');
disp('This program calculates the Hydrostatic-pressure');
disp('at any given water level at Oscillating Gorge');
disp('');
% Predefine some variables.
p = 62.5;
n_values = input('Enter maximum waterlevel to calculate pressure for: ');
F(1) = 0;
H(1) = 0;
E(1) = 0;
D = [1:n_values];

% Main loop where we approximate the integrals for waterlevels 1 to 'n_values' feet.
for ii=1:n_values
err = 1;
h = 1/2;
X1 = [0:h:D(ii)];
Y1 = p.*(D(ii)-X1).*(40-20.*exp((-0.01.*X1).^2));
I1 = trapz(X1,Y1);

% Checks if the approx error is small enough usning 'Richardsonextrapolation'.
while (err > 0.0001)
h = h/2;
X2 = [0:h:D(ii)];
Y2 = (p.*(D(ii)-X2).*(40-20.*exp((-0.01.*X2).^2)));
I2 = trapz(X2,Y2);
err = abs((I1-I2)/3);

I1 = I2;
end
```

```

F(ii) = I2;
H(ii) = h;
E(ii) = err;
end

% Plot of the pressure depending on water level.
subplot(3,1,1), plot(D,F);
xlabel('Water level');
ylabel('Pressure');
grid on

% Plot for Step-length used at different waterlevels.
subplot(3,1,2), plot(D,H);
xlabel('Water level');
ylabel('Step length');

% Plot for calculated error as a function of Water level.
subplot(3,1,3), plot(D,E);
xlabel('Water level');
ylabel('Approximated error');
disp(['Smallest step-length used: ' num2str(min(H));]);

% Clear all variables.
clear

```