

LLENADO DE UN TANQUE

Trabajo práctico especial.

Integrantes: Jeremias Caballero Celan, Cristobal iglesias.

Email : jeremiascaballerocean@gmail.com

Fecha : 13/06/2020

Índice

Introducción	3
Algoritmo Implementado	4
Casos de Prueba:	8
Caso N°1	8
Caso N°2	9
Caso N°3	10
Caso N°4	11

Introducción

Este informe presenta el sistema físico correspondiente a un llenado de un tanque de agua, dicho documento muestra diferentes etapas y variables que hacen modificar la resolución de este sistema.

De el tanque se sabe que tiene forma cilíndrica, un área transversal(A), y una altura(C), ambas medidas en m(metros), a sí mismo también se conoce la entrada de agua(E), la cual se introduce por la parte superior del tanque y la salida(S) que se produce por la parte inferior esta última puede ser o no constante es decir ser siempre igual o cambiar (aumentar o disminuir) con el paso del tiempo.

Por último tenemos la columna de agua(H) es decir el agua alojada internamente que al igual que los demás datos se mide en m(Metros).

La resolución de dicho sistema físico consiste en realizar ciertas observaciones(AT), en un lapso de tiempo y ver como la columna de agua(H) dependiendo de la entrada y salida evoluciona tanto incrementando o decrementando su nivel.

Para calcular H, es necesario la siguiente fórmula $h(t_i) = h(t_{i-1}) + (E(t_i) - S(t_i)) * \Delta t / A$

Donde, $h(t_i)$ = Columna de agua en el tiempo actual.

$h(t_{i-1})$ = la columna de agua que había en la observación anterior.

$E(t_i)$ = la entrada de agua en el tiempo actual.

$S(t_i)$ = La salida de agua en el tiempo actual.

(Δt) = el tiempo entre observaciones.

(A) = El área del tanque.

Por último el programa aloja los siguientes resultados:

- I. Tiempo t_i , en el que el tanque se llena por completo.
- II. Tiempo t_i , en el que el tanque se vacía por completo.
- III. Primer tiempo t_i , en el que la columna de agua es igual o superior a $C/2$.

Algoritmo Implementado

Al iniciar el programa se cuenta con un menú el cual pide ingresar al usuario un número con la tarea que desea realizar figura 1.1.

```
===== Bienvenido =====
1) Simular con E y S Constantes
2) Simular con  $S(t) = K * t$ 
3) Simular con  $S(t) = W * t * t$ 
4) Simular con  $S(t) = G/A * h(t)$ 
Ingrese una opcion :|
```

Una vez ingresado una opción válida (entre 1 y 4 inclusive) el programa le pedirá al usuario los datos necesarios para realizar el experimento en el caso de que la opción no sea válida el programa terminará.

En este ejemplo el usuario ingresa 1 y la salida que obtiene es la siguiente

```
===== Bienvenido =====
1) Simular con E y S Constantes
2) Simular con  $S(t) = K * t$ 
3) Simular con  $S(t) = W * t * t$ 
4) Simular con  $S(t) = G/A * h(t)$ 
Ingrese una opcion :1
Ingrese At:
```

Una vez ingresado todos los datos que el programa requiere estos mismos son pasados a una función interna del programa que se encargará de realizar y obtener los resultados del experimento.

Explicación ejercicio uno.

Para la resolución de la primera parte donde la salida y la entrada son constantes, el programa va utilizar la fórmula antes mencionada para calcular h en cualquier tiempo.

Además de los datos ingresado por el usuario se usan 5 reservas de memoria (variables) para almacenar los resultados que el programa vaya alojando las cuales son:

double tiempo

La variable tiempo declarada como double se encargará principalmente de almacenar el tiempo empleado desde el inicio hasta el final del experimento, esta es

muy útil porque nos permite saber en qué tiempo (medido en segundos) pasó tal cosa.

int iteraciones

La siguiente variable es usada para almacenar las iteraciones, muy útil para saber qué iteración se realizó una acción.

double valor

Valor se encarga principalmente de obtener el resultado que aloje tiempo cuando el tanque pase la mitad de agua, esta variable principalmente tendrá un valor cuando h sea mayor a $c/2$.

Cabe aclarar que permite que tiempo no se pise y pueda seguir incrementando.

boolean tanqueVacio;

boolean TanqueLleno;

Por último las variables booleanas alojarán verdadero o falso cuando el tanque esté lleno o vacío, se encargará principalmente de conocer si el tanque está vacío o lleno.

Una vez conocidas las variables utilizando por el programa se cuenta con una ciclo repetitivo while(Mientras), el cual va a iterar hasta que el tanque este vacio o se llegue a un millón de iteraciones, cuando lo último suceda el ciclo cortara y H almacenará la cantidad de agua hasta ese momento.

```
while (tanqueVacio == false) {
```

Por cada iteración dentro del ciclo repetitivo se obtiene h la cual se almacena en la variable resultado, a su vez tiempo se irá incrementado con el valor de AT.

El bucle dentro cuenta con condiciones lógicas(if)

La primera condición indica si H en este caso resultado pasó la mitad del tanque, a sí mismo cuenta con otra condición la cual nos permite manipular solo la primer vez que esto ocurra, sin esta condición en cada iteración que la columna de agua supere la mitad el programa nos informa, lo cual solo necesitamos saber la primer vez que ocurra.

por ultimo valor guardará el tiempo exacto en el que esto ocurrió y por consola saldrá un mensaje indicando los segundos y las iteraciones en el que esto sucedió.

```
if ((resultado >= c/2)&&(valor == 0 )){  
    valor = tiempo;  
    System.out.println("El tanque llevo a C/2 en : "+ valor + " segundos y "+iteraciones+" iteraciones");  
}
```

La siguiente condición permite saber cuando el tanque se llenó, para ello utiliza la lógica si H es mayor o igual a C y si el tanque no estaba lleno la última es empleada para cuando h sea mayor a c solo almacene la primer que se cumpla.

```
if((h>=c)&&(TanqueLleno == false)) {  
    System.out.println("El tanque se lleno en: "+ tiempo + " segundos y "+iteraciones+" iteraciones");  
    e = 0; // corto la entrada  
    TanqueLleno = true;  
}
```

Para saber en qué tiempo y iteración el tanque se vacía, la lógica empleada es si H es menor o igual a C y el tanque no está vacío.

Si esto sucede la salida por consola mostrará el mensaje con el tiempo y las iteraciones que sucedió dicho evento.

Por último la condición más básica que cuenta el algoritmo empleado es si las iteraciones son iguales a un millón, si esto sucede el programa deberá finalizar su ejecución, esto se emplea cuando el tanque nunca se llena o nunca se vacía.

Para los demás problemas en donde la salida no es constante el algoritmo es igual cambiando en cada caso, cuando la salida se mide con una función lineal (K) es necesario multiplicarla en cada iteración por el tiempo (T) actual, esta última se calcula incrementando AT

-K es solicitada al usuario antes de resolver el problema.

Cuando la salida se mide en una función cuadrática al usuario se le pide que ingrese W (medida en metros cúbicos).

A su vez el algoritmo necesita otra variable más llamada "Tiempo2" la cual comienza en 1

Para calcular H en cada iteración es necesario multiplicar el Tiempo2 por AT elevado al cuadrado.

En java la función `math.pow` nos devuelve la base elevada al exponente

`h+((e-w*Math.pow(Tiempo2 * at,2)) * at / a);`

Es necesario incrementar Tiempo2 por cada iteración.

```
private static void Realizar_ejercicio_tres(double at, double h, double e, double a, double c, double w) {  
    double tiempo= 0;  
    double tiempo2 = 1;  
    int iteraciones = 0;  
    double valor = 0;  
    boolean tanqueVacio = false;  
    boolean tanquelleno = false;  
    while(tanqueVacio == false) {  
        double resultado = h+((e-w*Math.pow(tiempo2*at,2))*at/a);  
        tiempo2++;  
        tiempo+=at;  
    }
```

Para la resolución cuando la salida es proporcional a la altura es necesario pedirle al usuario dicha salida (G).

Para calcular H por cada iteración es necesario la siguiente

fórmula: $h + (e - (g/a) * h) * at / a$, donde la salida se calcula de dividir área multiplicada por la columna de agua que había en el tiempo anterior.

Sabiendo que la salida es proporcional a H la cual está también baja en forma proporcional, por ende nunca será 0. Si deseamos saber cuando el tanque se vacía es necesario preguntar si $H \leq 0.000000001 (10^{-9})$. Si esto se cumple el tanque está vacío.

```
}  
if((h<=0.000000001)&&(tanqueVacio == false)) {  
    System.out.println("El tanque se vacio en: " + tiempo + " segundos y "+iteraciones+" iteraciones");  
    tanqueVacio = true;|  
}
```

Casos de Prueba:

Caso N°1

Ejemplo ejercicio **N°1** donde la entrada y la salida son constantes le ingresamos al programa los siguientes datos:

(S) = $(2\text{m}^3.\text{seg} / \text{s})$

(E) = $(5\text{m}^3.\text{seg} / \text{s})$

(H) = (2.4m)

(AT) = $(0,5\text{seg})$

(C) = (6m)

(A) = (3m^2)

Una vez ingresados los datos el programa comenzará a trabajar y aloja los siguientes resultados:

```
=====
El tanque llego a C/2 en : 1.0 segundos y 2 iteraciones
El tanque se lleno en: 4.0 segundos y 8 iteraciones
El tanque se vacio en: 14.0 segundos y 28 iteraciones
Total de iteraciones 28|
```


Caso N°2

Salida lineal,

Datos ingresados por consola:

(**K**) = $(2\text{m}^3.\text{seg} / \text{s})$
(**E**) = 0(El tanque comienza lleno)
(**H**) = (5m)
(**AT**) = (0.1seg)
(**C**) = (5m)
(**A**) = $(0,5\text{m}^2)$

Resultado:

```
=====
El tanque llego a C/2 en : 0.1 segundos y 1 iteraciones|
El tanque se lleno en: 0.1 segundos y 1 iteraciones
El tanque se vacio en: 1.7000000000000004 segundos y 17 iteraciones
Total de iteraciones  17
```

Caso N°3

Salida cuadrática

Datos ingresados por consola:

(W) = $(2\text{m}^3 \cdot \text{seg}^2)$

(E) = $(12 \text{ m}^3 \cdot \text{seg})$

(H) = (8m)

(AT) = (0.5seg)

(C) = (8m)

(A) = (2m^2)

Resultado

:

```
El tanque llego a la mitad en :1.0 segundos y 2 iteraciones
El tanque se lleno en: 2.0 segundos y 4 iteraciones
El tanque se vacio en: 3.5 segundos y 7 iteraciones
Total de iteraciones 7
```

Caso N°4

Salida proporcional a la altura

Datos ingresados por consola:

(G) = $(0.25\text{m}^4\cdot\text{seg})$

(E) = $(1\text{ m}^2\cdot\text{seg})$

(H) = (0.5m)

(AT) = (2seg)

(C) = (3.75m)

(A) = (1m^2)

Resultado:

```
El tanque llego a la mitad en :2.0 segundos y1 iteraciones
El tanque se lleno en: 8.0 segundos y 4 iteraciones
El tanque se vacio en: 72.0 segundos y 36 iteraciones
Total de iteraciones 36
```