Universidad Nacional del Centro de la Provincia de Buenos Aires
Facultad de Ciencias Exactas
Tecnicatura Universitaria en Desarrollo de Aplicaciones Informáticas

Trabajo Práctico Especial

Programa computacional para el llenado de un tanque de agua

Taller de Matemática Computacional

Notti Agustina

LU 250565

Tandil, junio 2020

INTRODUCCIÓN

El siguiente informe se realiza en el marco de la catedra Taller de Matemática Computacional perteneciente a la carrera Tecnicatura Universitaria en Desarrollo de Aplicaciones Informáticas de la UNCPBA.

El mismo tiene como objetivo presentar los resultados y conclusiones obtenidos a partir de la realización de un modelo computacional para el llenado de un tanque de agua.

La implementación del programa se centró en la simulación de la evolución temporal de la altura de una columna de agua a partir de una serie de parámetros ingresados por el usuario, como: entrada de agua (E), salida de agua, altura del tanque (C), altura de la columna de agua (h), paso del tiempo (Δt) y área transversal del tanque (A).

A partir de ello, se debía calcular:

1. Tiempo en que se llena el tanque.
2. Tiempo de vaciado del tanque.
3. Tiempo exacto en que el agua llega a mitad del tanque.

La realización del programa constó de dos partes. Una primera instancia en la cual se poseía una salida de agua constante (S); y una segunda instancia donde la salida de agua variaba según tres funciones:

A. $S(t) = K * t$
B. $S(t) = W * t^2$
C. $S(t) = G/A * h(t)$

En los tres casos, K, W y G, son valores ingresados por el usuario.

DESARROLLO

➢ Metodología:

Para la implementación del problema planteado se utilizo el lenguaje Java, que consiste un lenguaje tipado y compilado, el cual fue aprendido durante la cursada de la cátedra de Programación I.

Se busco llevar a cabo la solución del problema abstrayendo la funcionalidad del calculo del tiempo de llenado del tanque a través de un método que recibe como parámetros los valores ingresados por el usuario previamente.

```
calcularTiempoDeLlenado(alturaTanque, areaTransversal, alturaAgua, tiempo, entradaAgua, tipoSalidaCaudal, salidaAgua);
```
(Método de la resolución del problema)

En dicho método, se utilizó una estructura de control *while* para simular las diferentes iteraciones en las cuales se va llenando el tanque y calculando los nuevo valores: cantidad de iteraciones, tiempo transcurrido y la altura actual del agua, a través de la formula h = h+((E-S)*t/A)).

A su vez, la estructura mencionada anteriormente se utilizó estableciendo como condiciones de control el tiempo limite propuesto por la catedra (tiempo transcurrido < 1.000.000 segundos), como así también el nivel máximo de agua en el tanque (h < C).

Dentro de este método también se lleva a cabo la evaluación cuando el agua llega a mitad de tanque. Una vez que el taque es llenado se procede a la llamada de otro método, similar al del llenado del tanque, el cual calcula el tiempo de vaciado del tanque teniendo en cuenta las iteraciones y el tiempo transcurrido durante el llenado.

```
calcularTiempoDeVaciado(C, A, C, t,  0, tipoSalida, salida, iteraciones, tiempo);
```

A diferencia del método que calcula el tiempo de llenado, en este método se envía el parámetro de entrada de caudal de agua en 0, entendiendo que al llenarse el tanque se frena la entrada de agua al mismo.

Por otro lado, la condición del *while* para finalizar la ejecución corrobora que la altura de agua en el tanque llegue a 0 o, en el caso del tipo de salida G/A*h(t), se utilizo como limite 0.000000001, ya que por las propiedades de la función, esta se acerca a 0 pero nunca alcanzara ese valor.

A lo largo de la resolución del problema, se fue haciendo necesaria la declaración de constantes, ya sea para controlar el tiempo límite (o límite de iteraciones) en caso de que no se llegue al objetivo plateado en un tiempo determinado, como así también, la utilización de las mismas para evaluar el tipo de salida de agua deseada (S, K, G y W).

```
public static double obtenerSalidaDeAgua(double salidaAgua, double A, double h, int tipoSalida, double tiempo) {
    double salida = 0;

    if(tipoSalida == SALIDA_CONSTANTE) {
        salida = salidaAgua;
    } else if(tipoSalida == SALIDA_K) {
        salida = salidaAgua*tiempo;
    }else if(tipoSalida == SALIDA_W) {
        salida = salidaAgua*(tiempo*tiempo);
    }else if(tipoSalida == SALIDA_G) {
        salida = salidaAgua/A*h;
    }

    return salida;
}
```

Imagen 1: El método calcula la salida de agua, dependiendo del tipo de salida seleccionada por el usuario.

➢ Resultados:

Ejemplo 1:

- Entrada de agua: 4 m^3/s
- Salida de agua: 4 m^3/s
- Altura inicial: 1.5 m
- Paso del tiempo: 0.5 s
- Altura del contenedor: 6 m
- Área del contenedor: 4 m^2

TIPO DE SALIDA	TANQUE LLENO		MEDIO TANQUE		TANQUE VACIO		TOTAL ITERACIONES
	Iteraciones	Tiempo	Iteraciones	Tiempo	Iteraciones	Tiempo	
S (contante)	-	-	-	-	2000003	1000001.5	2000004

$S(t) = K * t$	-	-	-	-	2000000	1000000	2000001
$S(t) = W * t^2$	-	-	-	-	2000000	1000000	2000001
$S(t) = G/A * h(t)$	-	-	7	3.5	2000166	1000083	2000167

OBSERVACIONES:

- **Entrada y salida iguales:** Para el caso donde la entrada y salida son iguales, y esta última es constante, se observa que el tanque nunca llega a llenarse, manteniendo la altura inicial de agua y, una vez llega al millón de segundos, comienza a vaciarse desde esa altura.
- **Salida mayor a la entrada:** Para los casos de $S(t) = K * t$ y $S(t) = W * t^2$ se da que la salida, por las características de las funciones, se convierte en mayor que la entrada. Esto genera que el tanque nunca llega a llenarse, porque sale más agua que la que entra, y cuando llega a la condición del millón de segundos el tanque se encuentre vacío.
- **Salida inicialmente menor que la entrada:** Para el caso de $S(t) = G/A * h(t)$ se da que la salida, por la característica de la función, comienza siendo menor que el caudal de entrada (ejemplo $S(1) = 1.5$). A medida que aumenta h, el caudal de salida aumenta. Se puede corroborar que llego al medio tanque en la iteración numero 7. Sin embargo, cuando el caudal de salida llega al mismo caudal que la entrada, sucede igual que el primer caso: el tanque nunca llega a llenarse, ya que la entrada y salida de agua son iguales, por lo que mantiene el nivel del agua hasta que se cumple la condición de corte por el tiempo.

Ejemplo 2:

- Entrada de agua: 8 m^3/s
- Salida de agua: 3 m^3/s
- Altura inicial: 1.5 m
- Paso del tiempo: 0.5 s
- Altura del contenedor: 6 m
- Área del contenedor: 4 m^2

TIPO DE SALIDA	TANQUE LLENO		MEDIO TANQUE		TANQUE VACIO		TOTAL ITERACIONES
	Iteraciones	Tiempo	Iteraciones	Tiempo	Iteraciones	Tiempo	
S (contante)	8	4	3	1.5	24	12	25
S(t) = K * t	-	-	3	1.5	2000000	1000000	2000001
S(t) = W * t²	-	-	2	1	2000000	1000000	2000001
S(t) = G/A * h(t)	7	3.5	2	1	236	118	237

OBSERVACIONES:

- **Entrada mayor a la salida:** para los casos de salida de caudal de agua S y G la entrada se mantiene mayor que el caudal de salida, lo que permite llegar al objetivo de que el tanque se llene antes de que se ejecute la condición de corte por el tiempo límite. Se puede observar que en el caso de salida G, a medida que el tanque se vacía y que la altura de agua disminuye, el caudal de salida se ira haciendo cada vez menor.

Ejemplo 3:

- Entrada de agua: 8 m³/s
- Salida de agua: 3 m³/s
- Altura inicial: 0 m
- Paso del tiempo: 3 s
- Altura del contenedor: 6 m
- Área del contenedor: 4 m²

TIPO DE SALIDA	TANQUE LLENO		MEDIO TANQUE		TANQUE VACIO		TOTAL ITERACIONES
	Iteraciones	Tiempo	Iteraciones	Tiempo	Iteraciones	Tiempo	
S (contante)	2	6	1	3	5	15	6
S(t) = K * t	-	-	-	-	333334	1000002	333335
S(t) = W * t²	-	-	-	-	333334	1000002	333335
S(t) = G/A * h(t)	1	3	1	3	29	87	30

OBSERVACIONES:

- **Paso del tiempo mayor:** se puede observar (en S y G) que, a diferencia del ejemplo anterior, a mayor paso del tiempo, se dan menor cantidad de iteraciones para el llenado del tanque.

Ejemplo 4:

- Entrada de agua: 8 m^3/s
- Salida de agua: 3 m^3/s
- Altura inicial: 6 m
- Paso del tiempo: 3 s
- Altura del contenedor: 6 m
- Área del contenedor: 4 m^2

TIPO DE SALIDA	TANQUE LLENO		MEDIO TANQUE		TANQUE VACIO		TOTAL ITERACIONES
	Iteraciones	Tiempo	Iteraciones	Tiempo	Iteraciones	Tiempo	
S (contante)	-	-	-	-	3	9	4
$S(t) = K * t$	-	-	-	-	1	3	2
$S(t) = W * t^2$	-	-	-	-	1	3	2
$S(t) = G/A * h(t)$	-	-	-	-	28	84	29

OBSERVACIONES:

- **Inicio con tanque lleno:** se puede observar que, a diferencia de los ejemplos anteriores, en los cuales se comenzaba con el tanque vacío o con alguna altura inicial de agua, en estos casos, al iniciar con el tanque lleno, no se contabilizan las iteraciones de llenado en la cantidad de iteraciones totales.

A su vez, esto disminuye el tiempo transcurrido total calculado por el programa.

CONCLUSIONES

Las principales dificultades las presencia al momento de plantear inicialmente el problema al no saber como encarar la implementación del código llevando a cabo buenas prácticas, reutilización de código y modularizarían de las funcionalidades.

En dicho contexto otra punto observado fue la dificultad para la abstracción de código para ser reutilizado en diferentes casos. Esto, por ejemplo, para el caso de las diferentes salidas de caudal de agua, en el que se buscó que el usuario ingrese un solo valor de salida (S, K, W y G son la misma variable), lo cual fue solucionado sumando un parámetro (tipo de salida de agua) en la función.

Por otro lado, en un primer momento, al realizar los ejercicios de prueba, los resultados obtenidos no eran los correctos. Esto se debía a que no estaba teniendo en cuenta el incremento del tiempo en cada iteración.

A su vez, se descubrió que no se estaba teniendo en cuenta el tiempo y la cantidad de iteraciones del llenado del tanque para el calculo de las iteraciones totales y el tiempo total en el que el tanque se vacía.

Por último, para el caso donde se da que la salida se calcula mediante la función G/A * h(t), no se estaba teniendo en cuenta que, por las características de la función, esta nunca llegaría a 0, por lo que nunca se llegaba a la condición de corte. Esto se soluciono al incluir una condición donde, dado ese caso se dispuso un limite diferente de 0.