



MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÃO
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

sid.inpe.br/mtc-m21d/2023/12.21.12.57-RPQ

**ÁGUA NA REPRESA: UMA APLICAÇÃO DO
PARADIGMA DINÂMICA DE SISTEMAS COM
MODELAGEM MATEMÁTICA EM CÓDIGO LUA**

Meiriele Alvarenga Cumplido
Raiane Aparecida Lopes Neves
Rita de Cassya Almeida Sousa

CST-323: Introdução à Modelagem
do Sistema Terrestre Professor: Dr.
Pedro Ribeiro de Andrade Neto

URL do documento original:

<http://urlib.net/8JMKD3MGP3W34T/4ADQREL>

INPE
São José dos Campos
2023

PUBLICADO POR:

Instituto Nacional de Pesquisas Espaciais - INPE
Coordenação de Ensino, Pesquisa e Extensão (COEPE)
Divisão de Biblioteca (DIBIB)
CEP 12.227-010
São José dos Campos - SP - Brasil
Tel.:(012) 3208-6923/7348
E-mail: pubtc@inpe.br

CONSELHO DE EDITORAÇÃO E PRESERVAÇÃO DA PRODUÇÃO INTELLECTUAL DO INPE - CEPPII (PORTARIA Nº 176/2018/SEI-INPE):

Presidente:

Dra. Marley Cavalcante de Lima Moscati - Coordenação-Geral de Ciências da Terra (CGCT)

Membros:

Dra. Ieda Del Arco Sanches - Conselho de Pós-Graduação (CPG)
Dr. Evandro Marconi Rocco - Coordenação-Geral de Engenharia, Tecnologia e Ciência Espaciais (CGCE)
Dr. Rafael Duarte Coelho dos Santos - Coordenação-Geral de Infraestrutura e Pesquisas Aplicadas (CGIP)
Simone Angélica Del Ducca Barbedo - Divisão de Biblioteca (DIBIB)

BIBLIOTECA DIGITAL:

Dr. Gerald Jean Francis Banon
Clayton Martins Pereira - Divisão de Biblioteca (DIBIB)

REVISÃO E NORMALIZAÇÃO DOCUMENTÁRIA:

Simone Angélica Del Ducca Barbedo - Divisão de Biblioteca (DIBIB)
André Luis Dias Fernandes - Divisão de Biblioteca (DIBIB)

EDITORAÇÃO ELETRÔNICA:

Ivone Martins - Divisão de Biblioteca (DIBIB)
André Luis Dias Fernandes - Divisão de Biblioteca (DIBIB)



MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÃO
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

sid.inpe.br/mtc-m21d/2023/12.21.12.57-RPQ

**ÁGUA NA REPRESA: UMA APLICAÇÃO DO
PARADIGMA DINÂMICA DE SISTEMAS COM
MODELAGEM MATEMÁTICA EM CÓDIGO LUA**

Meiriele Alvarenga Cumplido
Raiane Aparecida Lopes Neves
Rita de Cassya Almeida Sousa

CST-323: Introdução à Modelagem
do Sistema Terrestre Professor: Dr.
Pedro Ribeiro de Andrade Neto

URL do documento original:

<<http://urlib.net/8JMKD3MGP3W34T/4ADQREL>>

INPE
São José dos Campos
2023



Esta obra foi licenciada sob uma Licença Creative Commons Atribuição-NãoComercial 3.0 Não Adaptada.

This work is licensed under a Creative Commons Attribution-NonCommercial 3.0 Unported License.

Sumário

1. Apresentação do Exercício.....	3
2. Desenvolvimento do Modelo	3
3. Respostas para os Cenários	5
4. Limitações do Modelo e Conclusões	7

1. Apresentação do Exercício

Como forma de avaliar o aprendizado sobre o paradigma Dinâmica de Sistemas, o problema apresentado no exercício intitulado *Water in the Dam* (Água na Represa) foi resolvido através de modelagem matemática, com as equações resultantes transcritas para um código (*script*) em linguagem de programação LUA nas interfaces de desenvolvimento ZeroBrane Studio e TerraME.

O enunciado do exercício menciona que, no ano de 1950, uma determinada cidade possui uma população de 100.000 (1E5 em notação científica) habitantes e uma represa com capacidade de 5.000.000.000 (5E9) m³ de água, a qual produz energia hidrelétrica para toda essa cidade. Há duas estações chuvosas por ano, sendo que as chuvas adicionam 2.000.000.000 (2E9) m³ de água à represa na primeira estação e 1.500.000.000 (1.5E9) m³ na segunda. Considera-se que a represa está cheia no início de 1950, cada habitante consome em média 10 kWh de energia por mês, e cada kWh requer 100 m³ de água para ser produzido. Além disso, o consumo de energia aumenta em média 5% por habitante a cada ano.

Diante das informações acima apresentadas é então requerido desenvolver um modelo para investigar cinco cenários futuros (listados de 1 a 5 abaixo) para a represa, de modo a responder ao seguinte questionamento para cada um deles: quanto tempo levará até que a represa não seja mais capaz de fornecer toda a energia necessária à cidade?

Cenário 1: Se nada mais acontecer.

Cenário 2: Se as turbinas requeressem apenas 80 m³ de água para gerar 1 kWh.

Cenário 3: Se o crescimento do consumo reduzisse pela metade.

Cenário 4: Se a chuva total caíssem pela metade a partir de 1970.

Cenário 5: Se os cenários (2), (3) e (4) ocorressem simultaneamente.

2. Desenvolvimento do Modelo

Para o desenvolvimento do modelo, e obtenção das respostas para cada cenário futuro proposto, as seguintes equações matemáticas foram implementadas no programa (as variáveis das equações são aqui apresentadas na sintaxe que constam no código do programa, de acordo com os parâmetros definidos para o modelo e os parâmetros definidos localmente nos eventos):

Equação 1: $\text{consumeEnergy} = \text{model.population} * \text{model.energy} * 12$ [kWh/year]

Equação 2: $\text{consumeWater} = \text{consumeEnergy} * \text{model.waterkWh}$ [m³/year]

- Equação 3: $\text{model.waterDam} = \text{model.waterDam} - \text{consumeWater}$ [m³/year]
- Equação 4: $\text{model.waterDam} = \text{model.waterDam} + (\text{model.rain1} + \text{model.rain2})$ [m³/year]
- Equação 5: $\text{model.rain1} = \text{model.rain1} / 2$
 $\text{model.rain2} = \text{model.rain2} / 2$ [m³/year]
- Equação 6: $\text{model.energy} = \text{model.energy} * (1 + \text{model.increaseEnergy})$ [kWh/month.inhabitant]

As Equações 1 e 2 indicam, respectivamente, o consumo de energia da população em um ano e o consumo de água para a geração dessa energia. A Equação 3 indica a quantidade de água na represa descontando o consumo da população e a Equação 4 mostra a quantidade de água na represa somando a contribuição das chuvas na primeira e segunda estações. A Equação 5 fornece um novo valor da contribuição das chuvas em cada estação. Por fim, a Equação 6 considera o aumento anual de 5% no consumo de energia da população.

Os parâmetros do modelo, as equações acima descritas e outros elementos/comandos que compõem os eventos criados dentro da função do modelo (*if statement*, *flag* de ativação, por exemplo), bem como *inputs* específicos para cada cenário e comandos de geração de gráficos de resultados, podem ser visualizados na Figura 1. O código na íntegra também segue em anexo sob a denominação ‘waterdam.lua’.

Figura 1: Programa do modelo na interface de desenvolvimento ZeroBrane Studio (parte 1/2).

```

6 Exercise #1: Water in the Dam]]
7
8 Dam = Model{
9   population = 1E5, -- inhabitants
10  waterDam = 5E9, -- m3 of water (threshold)
11  rain1 = 2E9, -- m3 of water from rains in the first season
12  rain2 = 1.5E9, -- m3 of water from rains in the second season
13  energy = 10, -- kWh of energy per month each inhabitant
14  waterkWh = 100, -- m3 of water each kWh
15  increaseEnergy = 0.05, -- 5% increase in consumption of energy per inhabitant each year
16  after1970 = false, -- to switch the amount of overall rain from 1970 onwards
17  dt = 1, -- year(s)
18  finalTime = 2010, -- years
19
20  init = function(model)
21    model.chart = Chart{
22      target = model,
23      select = "waterDam"
24    } -- Chart
25
26    model.timer = Timer{
27      Event{start = 1950, action = function()
28        local consumeEnergy = model.population * model.energy * 12 -- kWh/year
29        local consumeWater = consumeEnergy * model.waterkWh -- m3/year
30
31        model.waterDam = model.waterDam - consumeWater -- m3/year
32        model.waterDam = model.waterDam + (model.rain1 + model.rain2) -- m3/year
33
34        if model.waterDam < 0 then
35          model.waterDam = 0 -- disregarding negative amounts of water
36        elseif model.waterDam > 5E9 then
37          model.waterDam = 5E9 -- disregarding amounts of water greater than the dam capacity
38        end -- if
39      end}, -- Event
40

```

Parâmetros do modelo

Equações: 1, 2, 3, 4

if statement

Figura 1: Programa do modelo na interface de desenvolvimento ZeroBrane Studio (parte 2/2).

```

41     Event{start = 1970, action = function()
42         if model.after1970 == true then
43             model.rain1 = model.rain1 / 2 -- m3/year
44             model.rain2 = model.rain2 / 2 -- m3/year
45             return false
46         end -- if
47     end}, -- Event
48
49     Event{start = 1950, action = function()
50         consumption of energy model.energy = model.energy * (1 + model.increaseEnergy) -- kWh/month inhabitant, with the 5% increase in
51     end}, -- Event
52
53
54     Event{start = 1950, action = model.chart} -- one graph for each scenario
55 } -- Timer
56 end -- init
57 } -- Dam
58
59 env = Environment{
60     scenario1 = Dam(),
61     scenario2 = Dam{waterkWh = 80},
62     scenario3 = Dam{increaseEnergy = 0.05/2},
63     scenario4 = Dam{after1970 = true},
64     scenario5 = Dam{waterkWh = 80, increaseEnergy = 0.05/2, after1970 = true}
65 } -- Environment
66
67 chart = Chart{
68     target = env,
69     select = "waterDam"
70 } -- Chart
71
72 env:add(Event{start = 1950, action = chart}) -- all scenarios on a single graph
73 env:run()
74
    
```

flag para condição de chuva nos cenários 4 e 5

Equação 5

Equação 6

Gráficos para cada cenário

inputs específicos para os cenários

Gráfico para todos os cenários


waterdam.lua

Pode-se observar, por meio das unidades das equações desenvolvidas, que as mesmas foram parametrizadas para o passo de tempo de 1 ano ($dt = 1 \text{ year}$), com o qual obteve-se resultados coerentes, embora algumas limitações para o modelo foram reconhecidas e são declaradas no tópico 4 do presente relatório.

3. Respostas para os Cenários

Gráficos para cada cenário proposto, que indicam a resposta de quanto tempo (em anos) levará até que a represa não seja mais capaz de fornecer toda a energia necessária à cidade, são apresentados na Figura 2. A Figura 3 traz o mesmo resultado compilando todos os cenários em um único gráfico.

O instante de tempo em que o valor zero (0, no eixo das ordenadas) é atingido pelas curvas representa o ano (no eixo das abscissas) a partir do qual a represa não é mais capaz de suprir com a demanda de energia da população. Dito isso, e por meio dos gráficos de resposta para os cenários, é perceptível que o cenário 3 é o mais otimista entre todos os cenários simulados (represa capaz de fornecer energia durante o maior período), enquanto o cenário 4 é o mais crítico dentre eles.

Figura 2: Gráficos de resposta para cada cenário futuro proposto.

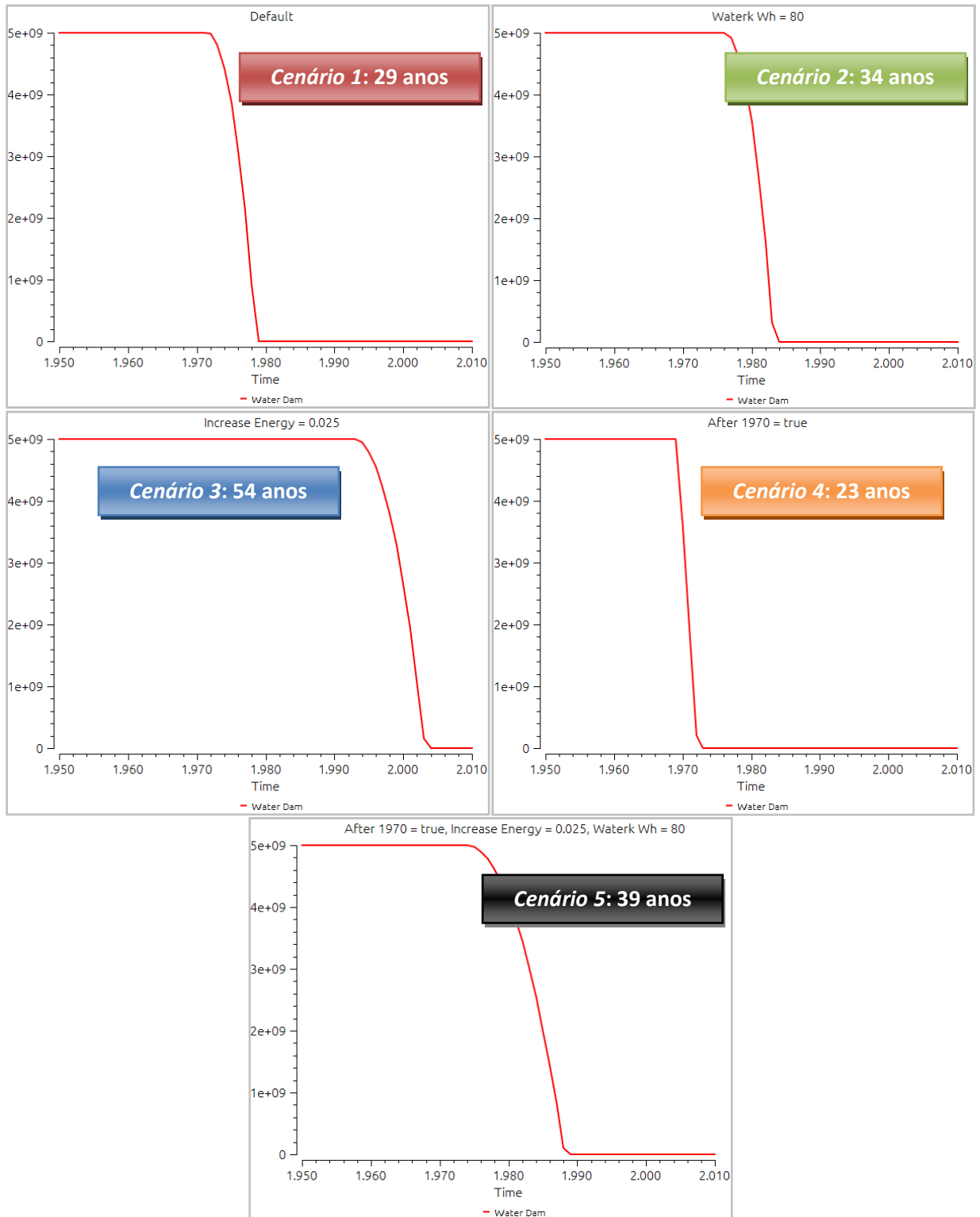
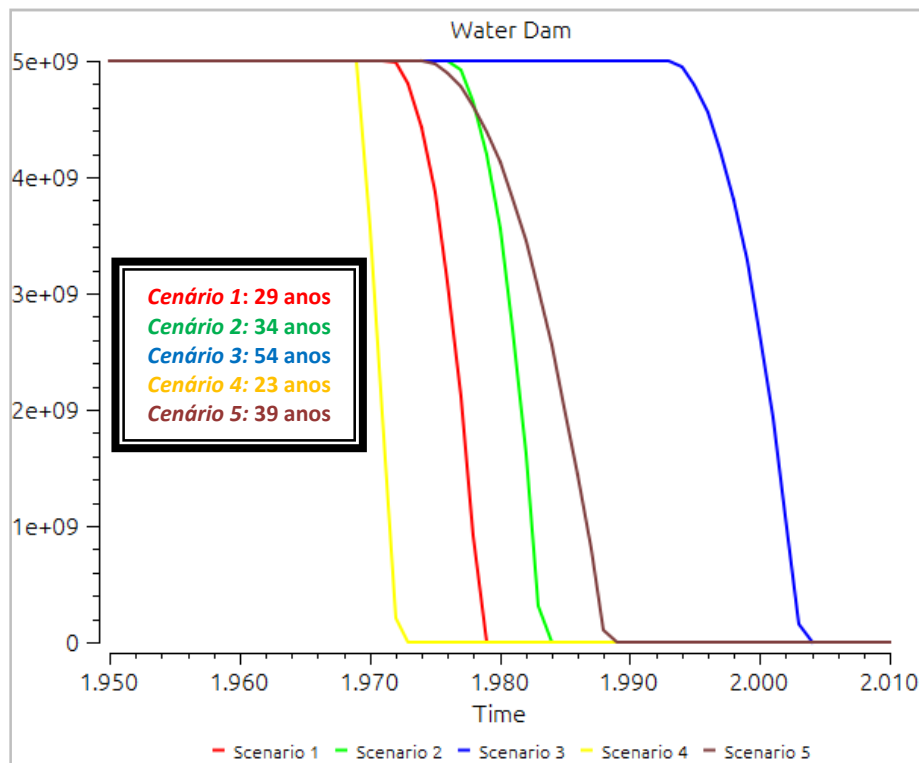


Figura 3: Gráfico de resposta para todos os cenários futuros propostos.



4. Limitações do Modelo e Conclusões

O programa desenvolvido assume o passo de tempo de 1 ano e incorpora as seguintes premissas no algoritmo: as chuvas das estações 1 e 2 são adicionadas de uma só vez à represa (Equação 4), e essa adição se dá após o consumo anual de água (Equação 3), em forma de energia, por parte da população. Isso explica o comportamento das curvas, as quais se mantêm constantes no valor limite de água na represa durante um determinado período seguido de uma queda abrupta na quantidade de água armazenada. Entende-se que um resultado mais realista seriam curvas “dentes de serra” com decaimento suave – a subida do “dente de serra” representaria o depósito de chuva ao longo das estações e a descida, o consumo de energia ao longo do ano; já o decaimento da curva se dá em função do progressivo consumo e volume insuficiente de chuva para compensar o aumento de consumo ano após ano. Os gráficos de resultado com representação pouco realista é uma das limitações do modelo.

As premissas adotadas também explicam a necessidade de inserir o condicional *if/elseif*, para desconsiderar quantidades negativas de água e quantidades superiores à capacidade da represa, após as equações 3 e 4 mencionadas, conforme destacado na Figura 4. Essa estratégia foi adotada para que as curvas atingissem o valor zero, que representa o momento a partir do qual a represa não

mais fornece a totalidade da energia para a cidade (vide Figura 2 e Figura 3). De forma a clarificar, a Figura 5 traz um código teste, definido como ‘programa teste (1)’, com um condicional *if* após a Equação 3 e um condicional *if* após a Equação 4. O resultado para o ‘programa teste (1)’ comparado ao ‘programa do modelo’, na Figura 6, ilustra essa outra limitação do modelo, também em consequência das premissas de que o consumo é primeiro contabilizado e depois toda a chuva é então adicionada à represa, levando à estabilização no valor da soma de chuvas (cenários 1, 2 e 3: 3.5E9 m³; cenários 4 e 5: 1.75E9 m³) ao invés do valor zero.

Figura 4: Programa do modelo: *dt = 1 year* e *if/elseif* após equações 3 e 4.

```

17 dt = 1, -- year(s)
18 finalTime= 2010, -- years
19
20 init = function(model)
21   model.chart = Chart{
22     target = model,
23     select = "waterDam"
24   } -- Chart
25
26 model.timer = Timer{
27   Event{start = 1950, action = function()
28     local consumeEnergy = model.population * model.energy * 12 -- kWh/year
29     local consumeWater = consumeEnergy * model.waterkWh -- m3/year
30
31     model.waterDam = model.waterDam - consumeWater -- m3/year
32     model.waterDam = model.waterDam + (model.rain1 + model.rain2) -- m3/year
33
34     if model.waterDam < 0 then
35       model.waterDam = 0 -- disregarding negative amounts of water
36     elseif model.waterDam > 5E9 then
37       model.waterDam = 5E9 -- disregarding amounts of water greater than the dam capacity
38     end -- if
39   end}, -- Event
40
41   Event{start = 1970, action = function()
42     if model.after1970 == true then
43       model.rain1 = model.rain1 / 2 -- m3/year
44       model.rain2 = model.rain2 / 2 -- m3/year
45       return false
46     end -- if
47   end}, -- Event
48
49   Event{start = 1950, action = function()
50     model.energy = model.energy * (1 + model.increaseEnergy) -- kWh/month.inhabitant, with the 5% increase in
51     consumption of energy
52   end}, -- Event
53

```

Programa do modelo
waterdam.lua

Figura 5: Programa teste (1): *dt = 1 year*, *if* após equação 3 e *if* após equação 4.

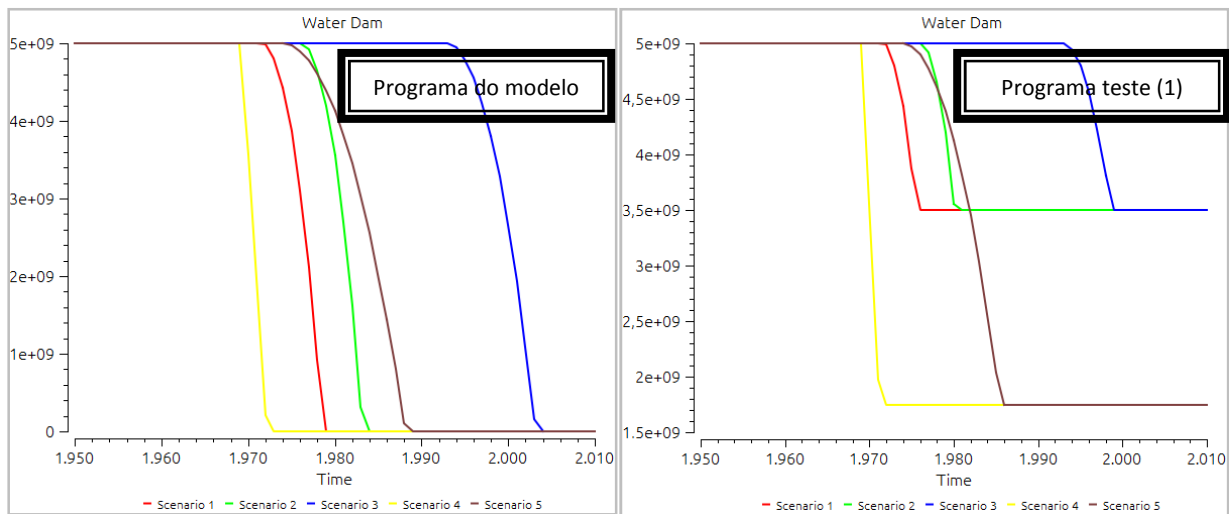
```

17 dt = 1, -- year(s)
18 finalTime= 2010, -- years
19
20 init = function(model)
21   --[[model.chart = Chart{
22     target = model,
23     select = "waterDam"
24   } -- Chart]]
25
26 model.timer = Timer{
27   Event{start = 1950, action = function()
28     local consumeEnergy = model.population * model.energy * 12 -- kWh/year
29     local consumeWater = consumeEnergy * model.waterkWh -- m3/year
30
31     model.waterDam = model.waterDam - consumeWater -- m3/year
32     if model.waterDam < 0 then
33       model.waterDam = 0 -- disregarding negative amounts of water
34     end -- if
35
36     model.waterDam = model.waterDam + (model.rain1 + model.rain2) -- m3/year
37     if model.waterDam > 5E9 then
38       model.waterDam = 5E9 -- disregarding amounts of water greater than the dam capacity
39     end -- if
40   end}, -- Event
41
42   Event{start = 1970, action = function()
43     if model.after1970 == true then
44       model.rain1 = model.rain1 / 2 -- m3/year
45       model.rain2 = model.rain2 / 2 -- m3/year
46       return false
47     end -- if
48   end}, -- Event
49
50   Event{start = 1950, action = function()
51     model.energy = model.energy * (1 + model.increaseEnergy) -- kWh/month.inhabitant, with the 5% increase in
52     consumption of energy
53   end}, -- Event

```

Programa teste (1)

Figura 6: Resultado para o programa do modelo e programa teste (1).



O passo de tempo alterado para 1 mês em um segundo código teste, denominado ‘programa teste (2)’ e mostrado na Figura 7, foi executado de forma a mostrar que um refinamento do passo de tempo do modelo minimizaria a discrepância descrita anteriormente. A Figura 8 é uma evidência ao trazer os resultados comparativos entre o ‘programa teste (1)’ e o ‘programa teste (2)’. Ainda assim, o algoritmo careceria de um maior refinamento no que tange uma chuva distribuída em estações. De toda forma, os resultados para passo de tempo em mês e em ano foram bem próximos (diferença relativa não superior a 5%, conforme Tabela 1) e o modelo desenvolvido com $dt = 1$ ano permaneceu como o escolhido, sob o julgamento de que as limitações não comprometeram sua robustez na questão a que se propõe responder.

Figura 7: Programa teste (2): $dt = 1$ month, if após equação 3 e if após equação 4.

```

17 dt = 1, -- month(s)
18 finalTime= 800, -- months
19
20 init = function(model)
21   --[[model.chart = Chart{
22     target = model,
23     select = "waterDam"
24   } -- Chart]]
25
26 model.timer = Timer{
27   Event{action = function()
28     local consumeEnergy = model.population * model.energy -- kWh/month
29     local consumeWater = consumeEnergy * model.waterkWh * model.dt -- m3/month
30
31     model.waterDam = model.waterDam - consumeWater -- m3/month
32     if model.waterDam < 0 then
33       model.waterDam = 0 -- disregarding negative amounts of water
34     end -- if
35
36     model.waterDam = model.waterDam + ((model.rain1 + model.rain2) / 12) * model.dt -- m3/month
37     if model.waterDam > 5E9 then
38       model.waterDam = 5E9 -- disregarding amounts of water greater than the dam capacity
39     end -- if
40   end}, -- Event
41
42   Event{start = 240, action = function()
43     if model.after1970 == true then
44       model.rain1 = model.rain1 / 2 -- m3/year
45       model.rain2 = model.rain2 / 2 -- m3/year
46       return false
47     end -- if
48   end}, -- Event
49
50   Event{action = function()
51     model.energy = model.energy * (1 + (model.increaseEnergy / 12) * model.dt) -- kWh/month.inhabitant, with the 5%
52     increase in consumption of energy
53   end}, -- Event

```

Figura 8: Resultado para o programa teste (1) e programa teste (2).

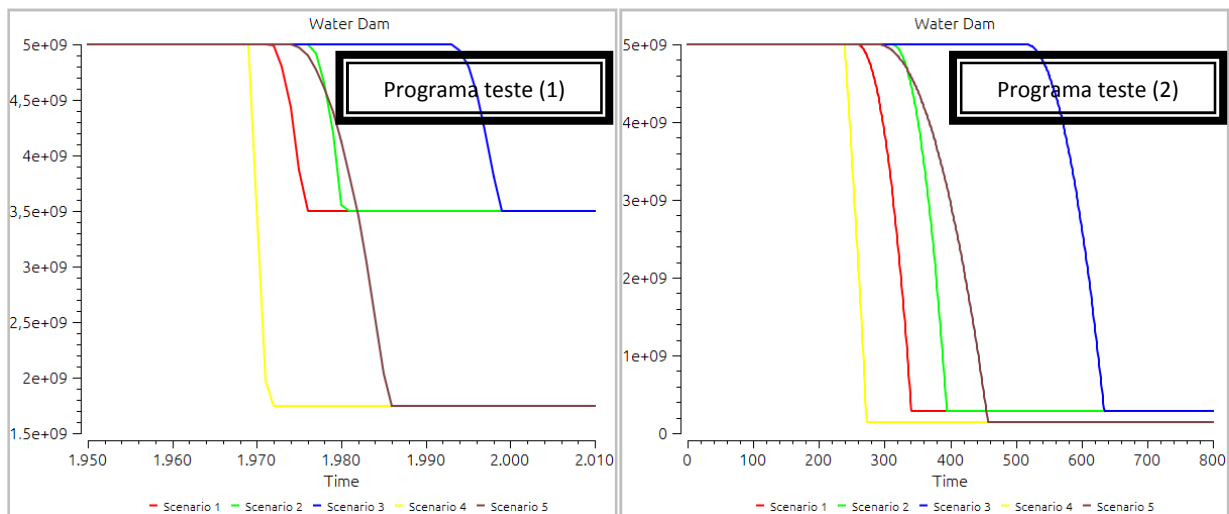


Tabela 1: Diferença relativa entre resultados para $dt = 1$ mês – ‘programa teste (2)’, e para $dt = 1$ ano – ‘programa do modelo’.

	Programa do modelo [anos]	Programa teste (2) [meses]	Programa teste (2) [anos]	Diferença relativa entre resultados [%]
Cenário 1:	29	270	28,33	2,36
Cenário 2:	34	390	32,5	4,61
Cenário 3:	54	630	52,5	2,86
Cenário 4:	23	270	22,5	2,22
Cenário 5:	39	460	38,33	1,75

Em modelo de Dinâmica de Sistemas, o processo de *feedback* inerente a certos sistemas pode não permitir uma previsibilidade do que vai acontecer, uma vez que o efeito da interferência nos chamados fluxos de entrada e saída de um estoque não acontece no tempo presente, mas se torna evidente na dinâmica ao longo do tempo. O modelo desenvolvido para resolver o problema apresentado no exercício *Water in the Dam* considerou a dinâmica da cidade (estoque), que eleva o consumo de energia (fluxo) anualmente, bem como a dinâmica da represa (estoque), a qual possui dependência de estações chuvosas (fluxo) para suprir com a demanda de energia da população. Por meio dos resultados do modelo, pode-se concluir que os cenários futuros propostos levaram, em tempos variados (mínimo de 23 anos no cenário 4 e máximo de 54 anos no cenário 3), ao colapso da represa, caracterizando um *feedback* do tipo positivo para o sistema de estudo.