



IPRJ  
Universidade do Estado  
do Rio de Janeiro

UERJ - Universidade do Estado do Rio de Janeiro

IPRJ - Instituto Politécnico do Rio de Janeiro

Graduação em Engenharia de Computação

# **Métodos Numéricos para Equações Diferenciais II Trabalho II Métodos TVD**

André Savioli Martins

Relatório apresentado ao curso de Engenharia  
de Computação, Universidade do Estado do  
Rio de Janeiro

**Professor:** Grazione de Souza Boy

**Nova Friburgo**

**12/08/2018**

# Lista de ilustrações

Figura 1 – Solução Analítica para o Problema do Traçador . . . . .	11
Figura 2 – Condição Inicial para o Problema do Traçador . . . . .	17
Figura 3 – Limitador Superbee - Primeiro Experimento em $t = 0,25$ . . . . .	19
Figura 4 – Limitador Superbee - Primeiro Experimento em $t = 0,5$ . . . . .	20
Figura 5 – Limitador Superbee - Primeiro Experimento em $t = 1,5$ . . . . .	20
Figura 6 – Limitador Superbee - Segundo Experimento em $t = 0,25$ . . . . .	21
Figura 7 – Limitador Superbee - Segundo Experimento em $t = 0,5$ . . . . .	22
Figura 8 – Limitador Superbee - Segundo Experimento em $t = 1,5$ . . . . .	22
Figura 9 – Limitador Superbee - Terceiro Experimento em $t = 0,25$ . . . . .	23
Figura 10 – Limitador Superbee - Terceiro Experimento em $t = 0,5$ . . . . .	24
Figura 11 – Limitador Superbee - Terceiro Experimento em $t = 1,5$ . . . . .	24
Figura 12 – Limitador Superbee - Quarto Experimento em $t = 0,25$ . . . . .	25
Figura 13 – Limitador Superbee - Quarto Experimento em $t = 0,5$ . . . . .	26
Figura 14 – Limitador Superbee - Quarto Experimento em $t = 1,5$ . . . . .	26
Figura 15 – Limitador MC - Primeiro Experimento em $t = 0,25$ . . . . .	28
Figura 16 – Limitador MC - Primeiro Experimento em $t = 0,5$ . . . . .	29
Figura 17 – Limitador MC - Primeiro Experimento em $t = 1,5$ . . . . .	29
Figura 18 – Limitador MC - Segundo Experimento em $t = 0,25$ . . . . .	31
Figura 19 – Limitador MC - Segundo Experimento em $t = 0,5$ . . . . .	32
Figura 20 – Limitador MC - Segundo Experimento em $t = 1,5$ . . . . .	32
Figura 21 – Limitador MC - Terceiro Experimento em $t = 0,25$ . . . . .	34
Figura 22 – Limitador MC - Terceiro Experimento em $t = 0,5$ . . . . .	35
Figura 23 – Limitador MC - Terceiro Experimento em $t = 1,5$ . . . . .	35
Figura 24 – Limitador MC - Quarto Experimento em $t = 0,25$ . . . . .	37
Figura 25 – Limitador MC - Quarto Experimento em $t = 0,5$ . . . . .	38
Figura 26 – Limitador MC - Quarto Experimento em $t = 1,5$ . . . . .	39
Figura 27 – Limitador van Leer - Primeiro Experimento em $t = 0,25$ . . . . .	42
Figura 28 – Limitador van Leer - Primeiro Experimento em $t = 0,5$ . . . . .	43
Figura 29 – Limitador van Leer - Primeiro Experimento em $t = 1,5$ . . . . .	43
Figura 30 – Limitador van Leer - Segundo Experimento em $t = 0,25$ . . . . .	45
Figura 31 – Limitador van Leer - Segundo Experimento em $t = 0,5$ . . . . .	46
Figura 32 – Limitador van Leer - Segundo Experimento em $t = 1,5$ . . . . .	46
Figura 33 – Limitador van Leer - Terceiro Experimento em $t = 0,25$ . . . . .	48
Figura 34 – Limitador van Leer - Terceiro Experimento em $t = 0,5$ . . . . .	49
Figura 35 – Limitador van Leer - Terceiro Experimento em $t = 1,5$ . . . . .	49
Figura 36 – Limitador van Leer - Quarto Experimento em $t = 0,25$ . . . . .	51

Figura 37 – Limitador van Leer - Quarto Experimento em $t = 0,5$ . . . . .	52
Figura 38 – Limitador van Leer - Quarto Experimento em $t = 1,5$ . . . . .	52
Figura 39 – Exemplo de Oscilação Espúria para o Problema do Traçador obtida através do Método não TVD Lax-Wendroff . . . . .	53
Figura 40 – Comparação dos Métodos TVD e da Solução Analítica com o Método UpWind para o Primeiro Experimento . . . . .	54
Figura 41 – Comparação dos Métodos TVD e da Solução Analítica com o Método UpWind para o Segundo Experimento . . . . .	55
Figura 42 – Comparação dos Métodos TVD e da Solução Analítica com o Método UpWind para o Terceiro Experimento . . . . .	55

# Lista de tabelas

Tabela 1	–	Definições de Variáveis para os Experimentos sobre cada Método Numérico	16
Tabela 2	–	Configuração do Computador Utilizado para a Execução dos Experimentos	16
Tabela 3	–	Tempo de Execução em Segundos para cada Método e Experimento . .	54

# Lista de abreviaturas e siglas

EDP	Equação Diferencial Parcial
CFL	Courant–Friedrichs–Lewy
MC	Monotonized Central-Difference

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>8</b>
<b>1.1</b>	<b>Objetivo</b>	<b>8</b>
<b>2</b>	<b>METODOLOGIA</b>	<b>10</b>
<b>2.1</b>	<b>Fundamentos Teóricos</b>	<b>10</b>
2.1.1	Solução Analítica	10
2.1.2	Solução Numérica	12
2.1.2.1	Métodos TVD	13
2.1.2.1.1	Método TVD com Limitador Superbee	14
2.1.2.1.2	Método TVD com Limitador MC	14
2.1.2.1.3	Método TVD com Limitador van Leer	15
<b>3</b>	<b>RESULTADOS</b>	<b>16</b>
<b>3.1</b>	<b>Resultados para o Método TVD com Limitador Superbee</b>	<b>18</b>
3.1.1	Primeiro Experimento do Método TVD com Limitador Superbee	19
3.1.2	Segundo Experimento do Método TVD com Limitador Superbee	21
3.1.3	Terceiro Experimento do Método TVD com Limitador Superbee	23
3.1.4	Quarto Experimento do Método TVD com Limitador Superbee	25
<b>3.2</b>	<b>Resultados para o Método TVD com Limitador MC</b>	<b>27</b>
3.2.1	Primeiro Experimento do Método TVD com Limitador MC	28
3.2.2	Segundo Experimento do Método TVD com Limitador MC	30
3.2.3	Terceiro Experimento do Método TVD com Limitador MC	33
3.2.4	Quarto Experimento do Método TVD com Limitador MC	36
<b>3.3</b>	<b>Resultados para o Método TVD com Limitador van Leer</b>	<b>40</b>
3.3.1	Primeiro Experimento do Método TVD com Limitador van Leer	41
3.3.2	Segundo Experimento do Método TVD com Limitador van Leer	44
3.3.3	Terceiro Experimento do Método TVD com Limitador van Leer	47
3.3.4	Quarto Experimento do Método TVD com Limitador van Leer	50
<b>4</b>	<b>DISCUSSÃO</b>	<b>53</b>
<b>5</b>	<b>CONCLUSÃO</b>	<b>56</b>
	<b>REFERÊNCIAS</b>	<b>57</b>

## **ANEXOS**

**58**

<b>ANEXO A – CÓDIGO PYTHON PARA OS MÉTODOS TVD COM LIMITADORES DE FLUXO SUPERBEE, MC E VAN LEER . . . . .</b>	<b>59</b>
---	-----------

# 1 Introdução

A Modelagem Computacional é a área da ciência que utiliza modelos matemáticos para a análise, solução e simulação de problemas científicos através de algoritmos e o emprego de computadores.

Quando se trata das leis naturais fundamentais, por exemplo, a Lei da Conservação de Massa, grande parte delas são governadas por Equações Diferenciais Parciais (EDPs), e para certos problemas, dependendo das suas condições iniciais, de contorno ou, de ambas é difícil encontrar uma solução analítica.

Tendo isto em vista é possível encontrar uma aproximação do problema usando técnicas para aproximação de equações diferenciais parciais, onde nos dias atuais, encontraremos uma variedade de métodos à disposição, cada qual com seus pontos positivos e negativos levando em consideração o problema ao qual serão aplicados, tornando-se imprescindível a avaliação de cada um.

Em se tratando de métodos para resolução de problemas que envolvem equações diferenciais parciais podemos inicialmente lançar mão de discretizações a partir do método de elementos finitos usando Diferenças Finitas que é guiado por aproximação de derivadas ou do método de Volumes Finitos, que contempla as aproximações de integrais.

## 1.1 Objetivo

Este trabalho tem como objetivo comparar a aplicação de métodos *Total Variation Diminishing* (TVD) na obtenção de aproximações numéricas de problemas que envolvem EDPs hiperbólicas.

Sendo assim trataremos do problema que envolve a determinação da variação espacial e temporal da concentração de um traçador escoando em um meio físico, que é modelado matematicamente pela EDP de advecção unidimensional, que é dada por

$$\frac{\partial c}{\partial t} + \bar{u} \frac{\partial c}{\partial x} = 0 \quad (1.1)$$

Onde  $c$  indica a variável dependente e  $\bar{u}$  a velocidade de advecção, que para este trabalho é constante e positiva.

A condição inicial para este problema é dada por

$$c(x, 0) = c_0 \quad (1.2)$$



onde  $c_0$  é uma constante entre  $0 \leq c_0 < 1,0$  que determina a concentração inicial no domínio de comprimento  $L$ .

Além da condição inicial, este problema envolve condições de contorno, sendo a primeira dada por

$$c(0, t) = C \tag{1.3}$$

com  $C$  constante e maior que  $c_0$ , e a segunda por

$$\frac{\partial c}{\partial x} = 0 \tag{1.4}$$

em  $x = L$ .

Por simplicidade a referência à este problema será feita apenas como *Problema do Traçador*, e sobre ele aplicaremos três métodos TVD, o primeiro com limitador de fluxo *Superbee*, o segundo, um método com limitador *Monotonized Central-Difference* (MC) e o terceiro e último, um método com limitador *van Leer*.

## 2 Metodologia

Neste capítulo serão descritos os procedimentos realizados para a obtenção dos resultados do [Capítulo 3](#). Serão descritos os cuidados que foram tomados na implementação de uma formulação geral para métodos com limitadores de fluxo apresentada na [subseção 2.1.2.1](#) e os limitadores utilizados neste trabalho, além disso na [subseção 2.1.1](#) é discutida a necessidade de se obter a solução analítica do Problema do Traçador, que existe, para compará-la com as soluções numéricas obtidas.

Na [subseção 2.1.2.1](#) são abordados os fundamentos teóricos e as características que fazem com que um método seja ou não TVD através de uma noção formal e outra prática sobre essa classificação. Nas subseções seguintes abordamos a formulação e o comportamento dos limitadores de fluxo Superbee ([subseção 2.1.2.1.1](#)), MC ([subseção 2.1.2.1.2](#)) e van Leer ([subseção 2.1.2.1.3](#)) quando estes tendem ao infinito.

### 2.1 Fundamentos Teóricos

Nesta seção serão abordados os fundamentos teóricos necessários para avaliação dos métodos TVD com limitadores de fluxo Superbee, MC e van Leer em relação a solução analítica e as soluções numéricas a serem obtidas.

#### 2.1.1 Solução Analítica

Obter uma solução exata é algo muito importante para a avaliação de métodos numéricos, pois através de problemas onde a mesma exista, trará a segurança adicional quando o método for aplicado para a obtenção de casos em que não exista uma solução analítica.

O Problema do Traçador apresentado na [seção 1.1](#) pode ser reduzido a um problema de Riemann ([MAZZIA, 2012](#)) onde temos a seguinte solução

$$c(x, t) = \begin{cases} C, & \text{se } x < \bar{u}t \\ c_0, & \text{se } x > \bar{u}t \end{cases} \quad (2.1)$$

Através da [Figura 1](#) podemos observar o aspecto da solução analítica que consiste em uma onda sendo advectada com velocidade  $\bar{u}$  ao longo do tempo.

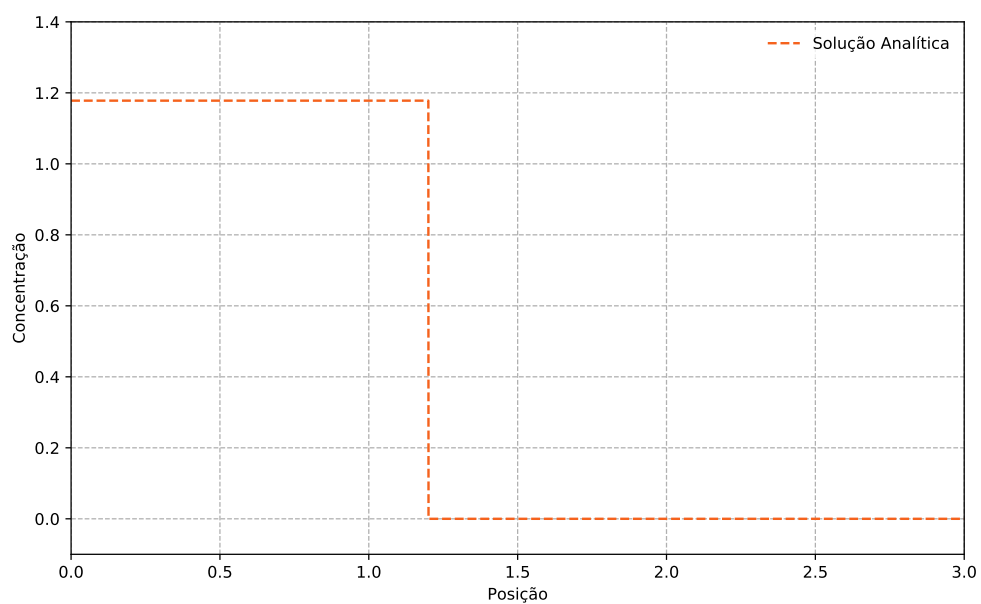


Figura 1 – Solução Analítica para o Problema do Traçador

A solução analítica dada pela [Equação 2.1](#) será utilizada a fim de ser uma referência precisa para as comparações feitas no [Capítulo 3](#).

### 2.1.2 Solução Numérica

Para a implementação dos métodos de solução numérica foram utilizadas: a linguagem de programação *Python*<sup>1</sup> versão 3.5, a biblioteca de computação científica *NumPy*<sup>2</sup>, e, para a geração de gráficos, a biblioteca *Matplotlib*<sup>3</sup>..

De imediato definiremos como  $C$  a seguinte expressão

$$C = \bar{u} \frac{\Delta t}{\Delta x} \quad (2.2)$$

onde  $C$  é o número de *Courant* ou *Courant–Friedrichs–Lewy* (CFL), que será encontrada na subseção 2.1.2.1 e será importante para a discussão e a conclusão, no Capítulo 4 e Capítulo 5.

Para os métodos empregados neste trabalho deve ser atendida a condição de estabilidade

$$C \leq 1 \quad (2.3)$$

sendo assim, para que isso seja demonstrado de forma prática, o quarto experimento de cada método mostrará o comportamento das aproximações quando esta condição não for respeitada.

A implementação do algoritmo de solução numérica TVD que utiliza os limitadores Superbee, MC e van Leer encontra-se em anexo.

---

<sup>1</sup> <<https://www.python.org>>

<sup>2</sup> <<http://www.numpy.org>>

<sup>3</sup> <<https://matplotlib.org>>

### 2.1.2.1 Métodos TVD

Métodos TVD são métodos que levam a variação total em consideração para a sua construção, que além de ser uma forma mais fácil de se verificar a convergência de um método numérico não-linear, elimina oscilações indesejadas. Na prática os métodos TVD tentam combinar as vantagens de métodos UpWind com métodos de segunda ordem (SOUTO, 2017).

Formalmente para que um método seja considerado TVD, para qualquer conjunto de dados  $Q^n$ , os valores  $Q^{n+1}$  calculados devem satisfazer

$$TV(Q^{n+1}) \leq TV(Q^n) \quad (2.4)$$

Isso nos leva a uma característica importante dos métodos TVD é a preservação da monotonicidade, ou seja

$$Q_i^n \geq Q_{i+1}^n \implies Q_i^{n+1} \geq Q_{i+1}^{n+1}, \forall i \quad (2.5)$$

Os métodos TVD determinam os fluxos nas faces dos volumes finitos através da introdução de um termo anti-difusivo e de uma função limitadora de fluxo

$$F_{i-1/2}^n = \bar{u}^- Q_i^n + \bar{u}^+ Q_{i-1}^n + \frac{1}{2} |\bar{u}| \left( 1 - \left| \frac{\bar{u} \Delta t}{\Delta x} \right| \right) \delta_{i-1/2}^n \quad (2.6)$$

$$F_{i+1/2}^n = \bar{u}^- Q_{i+1}^n + \bar{u}^+ Q_i^n + \frac{1}{2} |\bar{u}| \left( 1 - \left| \frac{\bar{u} \Delta t}{\Delta x} \right| \right) \delta_{i+1/2}^n \quad (2.7)$$

válidos para  $\bar{u} > 0$  e  $\bar{u} < 0$ . No entanto, como o Problema do Traçador tem  $\bar{u} > 0$ , consideraremos apenas o caso  $\bar{u} > 0$  para este desenvolvimento. Dessa forma os limitadores de fluxo serão obtidos por

$$\delta_{i\pm 1/2}^n = \psi(\theta_{i\pm 1/2}^n) \Delta Q_{i\pm 1/2}^n \quad (2.8)$$

onde

$$\theta_{i\pm 1/2}^n = \frac{\Delta Q_{I\pm 1/2}^n}{\Delta Q_{i\pm 1/2}^n} \quad (2.9)$$

com  $I$  utilizado para indicar a interface  $x_{i-1/2}$ , sendo  $I = i - 1$  para o caso  $\bar{u} > 0$ .

Como será visto em seguida, forneceremos diferentes valores de  $\theta$  para a função limitadora de fluxo, e unindo a utilização dos fluxos e as expressões  $\delta_{i-1/2}^n$  e  $\delta_{i+1/2}^n$  chegaremos a conveniente forma geral para os métodos numéricos com limitadores de fluxo, dada por

$$Q_i^{n+1} = Q_i^n - C(Q_i^n - Q_{i-1}^n) - \frac{1}{2}C(1 - C)[\psi(\theta_{i+1/2}^n)(Q_{i+1}^n - Q_i^n) - \psi(\theta_{i-1/2}^n)(Q_i^n - Q_{i-1}^n)] \quad (2.10)$$

No contexto desta formulação  $C$  representa o número de *Courant* definido pela Equação 2.5, e não deve ser confundido com a constante  $C$  da condição de contorno apresentada na Equação 1.3.

Esta formulação geral tem  $\theta_{i-1/2}^n$  e  $\theta_{i+1/2}^n$  calculados conforme a Equação 2.11 e Equação 2.12 abaixo.

$$\theta_{i-1/2}^n = \frac{Q_{i-1}^n - Q_{i-2}^n}{Q_i^n - Q_{i-1}^n} \quad (2.11)$$

$$\theta_{i+1/2}^n = \frac{Q_i^n - Q_{i-1}^n}{Q_{i+1}^n - Q_i^n} \quad (2.12)$$

Como dito anteriormente a implementação computacional deve levar em conta os *ghost values* existentes nas extremidades do domínio computacional. Tendo isto em vista, nos casos em que foi não possível avaliar os valores de  $\theta_{i-1/2}^n$  e  $\theta_{i+1/2}^n$ , foi utilizado o valor  $\psi(\theta_{i\pm 1/2}^n) = 1$  que recupera o método Lax-Wendroff, que é um método de segunda ordem não TVD, em detrimento do método TVD de primeira ordem UpWind.

#### 2.1.2.1.1 Método TVD com Limitador Superbee

O limitador Superbee é o primeiro limitador de fluxo utilizado na formulação dada pela Equação 2.10, e é definido como:

$$\psi(\theta) = \frac{\theta + |\theta|}{1 + |\theta|} \quad (2.13)$$

Para este limitador quando  $\theta \rightarrow \infty$  tem-se que  $\psi(\theta) = 2$ .

#### 2.1.2.1.2 Método TVD com Limitador MC

O limitador MC é o segundo limitador de fluxo utilizado na formulação dada pela Equação 2.10, e é definido como:

$$\psi(\theta) = \max(0, \min((1 + \theta)/2, 2, 2\theta)) \quad (2.14)$$

Para este limitador quando  $\theta \rightarrow \infty$  tem-se que  $\psi(\theta) = 2$ .

### 2.1.2.1.3 Método TVD com Limitador van Leer

O limitador van Leer é o terceiro limitador de fluxo utilizado na formulação dada pela [Equação 2.10](#), e é definido como:

$$\psi(\theta) = \max(0, \min(1, 2\theta), \min(2, \theta)) \quad (2.15)$$

Para este limitador quando  $\theta \rightarrow \infty$  tem-se que  $\psi(\theta) = 2$ .

### 3 Resultados

Neste Capítulo apresentamos graficamente os resultados obtidos através da execução dos métodos implementados computacionalmente. Ao todo foram realizados 12 experimentos, 4 para cada método variando os parâmetros  $\Delta t$ ,  $\Delta x$ ,  $\bar{u}$  e consequentemente o número de *Courant*, conforme a [Tabela 1](#).

Núm. do Experimento	L	C	$c_0$	$\Delta t$	$\Delta x$	$\bar{u}$	Núm. de Courant
1	3	1,178	0	0,01	0,02	0,4	0,2
2	3	1,178	0	0,001	0,001	0,6	0,6
3	3	1,178	0	0,0085	0,0085	0,9	0,8
4	3	1,178	0	0,01	0,01	1,01	1,01

Tabela 1 – Definições de Variáveis para os Experimentos sobre cada Método Numérico

Todos os experimentos foram realizados utilizando um computador com a seguinte configuração apresentada na [Tabela 2](#). Como o processo de obtenção das soluções numéricas é *CPU bound*, ou seja, o processador é altamente exigido, apresentamos as configurações referentes ao processador de forma mais detalhada.

Componente	Configuração	Valor
Processador	Modelo	Intel Core i7 - i7-7500U
Processador	Geração	7ª geração (Kaby Lake)
Processador	Nº de Núcleos	2
Processador	Nº de Threads	4
Processador	Cache	4 MB SmartCache
Processador	Frequência Máxima	3,5 GHz
Processador	Frequência Baseada em Processador	2,7 GHz
Memória	Memória RAM	16GB (2x8GB)
Memória	Tipo de Memória RAM	DDR4
Memória	Frequência de Memória RAM	2400MHz

Fonte: [https://ark.intel.com/PT-BR/products/95451/Intel-Core-i7-7500U-Processor-4M-Cache-up-to-3\\_50-GHz-](https://ark.intel.com/PT-BR/products/95451/Intel-Core-i7-7500U-Processor-4M-Cache-up-to-3_50-GHz-)

Fonte: [https://topics-cdn.dell.com/pdf/inspiron-15-7560-laptop\\_specifications\\_pt-br.pdf](https://topics-cdn.dell.com/pdf/inspiron-15-7560-laptop_specifications_pt-br.pdf)

Tabela 2 – Configuração do Computador Utilizado para a Execução dos Experimentos

Nos gráficos deste capítulo a solução analítica é apresentada em linha tracejada vermelha e a solução numérica é apresentada em linha contínua preta.



Abaixo na [Figura 2](#) temos o gráfico da condição inicial em  $t = 0$ , que é igual para todos os experimentos.

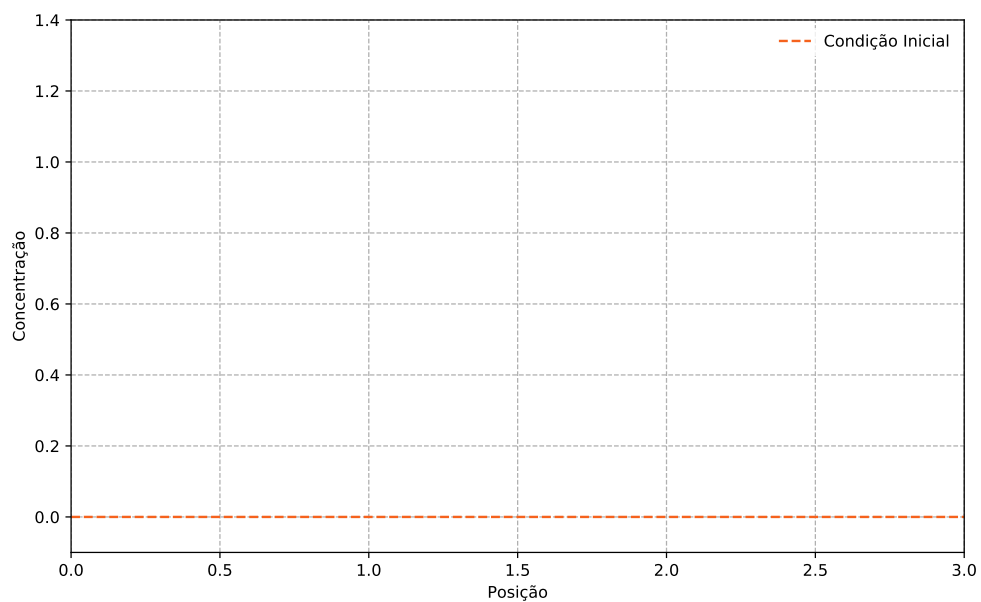


Figura 2 – Condição Inicial para o Problema do Traçador

## 3.1 Resultados para o Método TVD com Limitador Superbee

Na [subseção 3.1.1](#), [subseção 3.1.2](#), [subseção 3.1.3](#) e [subseção 3.1.4](#) são apresentados os resultados para os 4 experimentos realizados para o método TVD com limitador Superbee tendo em vista os dados da [Tabela 1](#).

### 3.1.1 Primeiro Experimento do Método TVD com Limitador Superbee

Abaixo apresentamos os resultados do primeiro experimento do Método TVD com limitador Superbee, com tempo total de execução de 4,241 segundos.

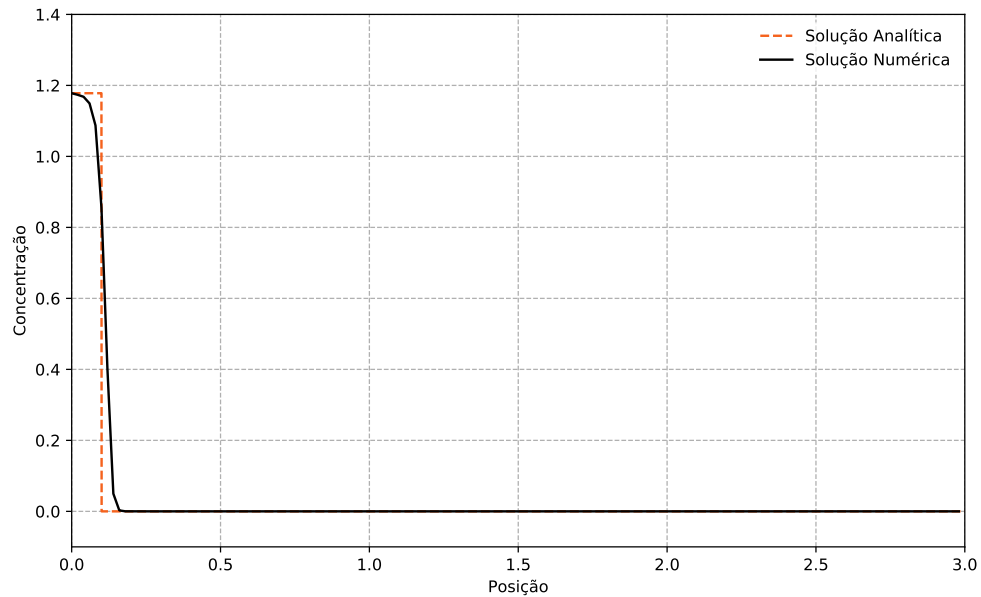
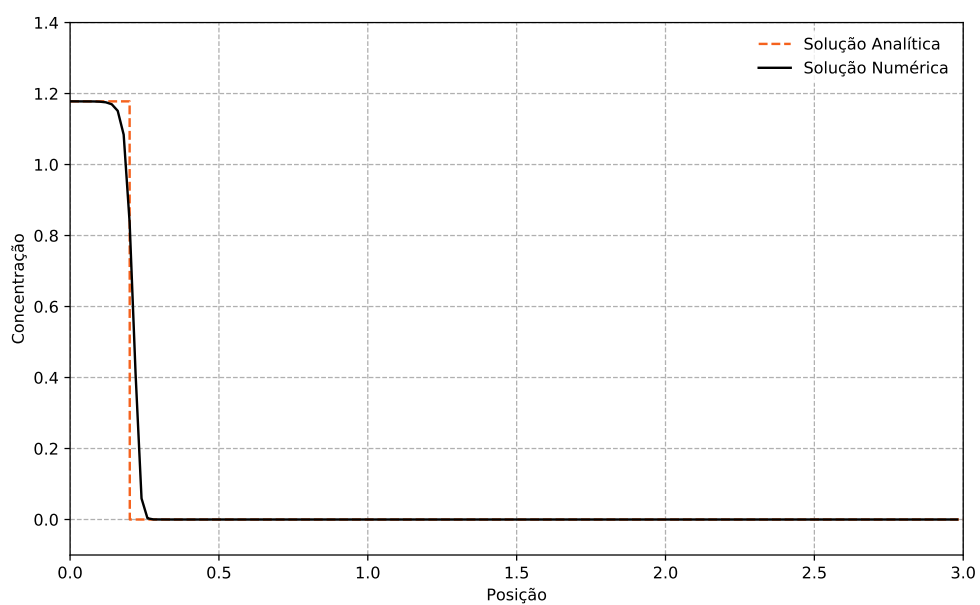
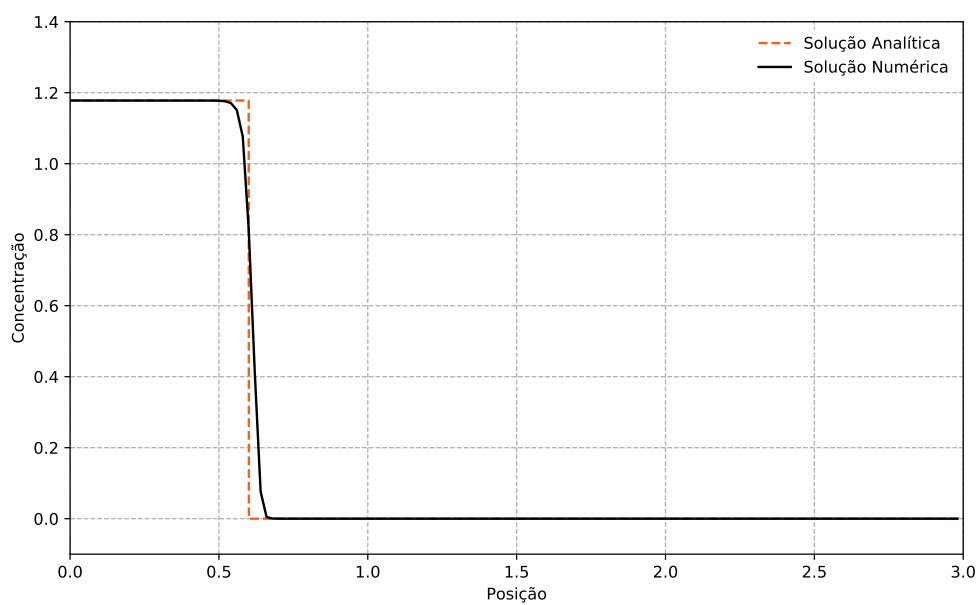


Figura 3 – Limitador Superbee - Primeiro Experimento em  $t = 0,25$

Figura 4 – Limitador Superbee - Primeiro Experimento em  $t = 0,5$ Figura 5 – Limitador Superbee - Primeiro Experimento em  $t = 1,5$

### 3.1.2 Segundo Experimento do Método TVD com Limitador Superbee

Abaixo apresentamos os resultados do segundo experimento do Método TVD com limitador Superbee, com tempo total de execução de 910,025 segundos.

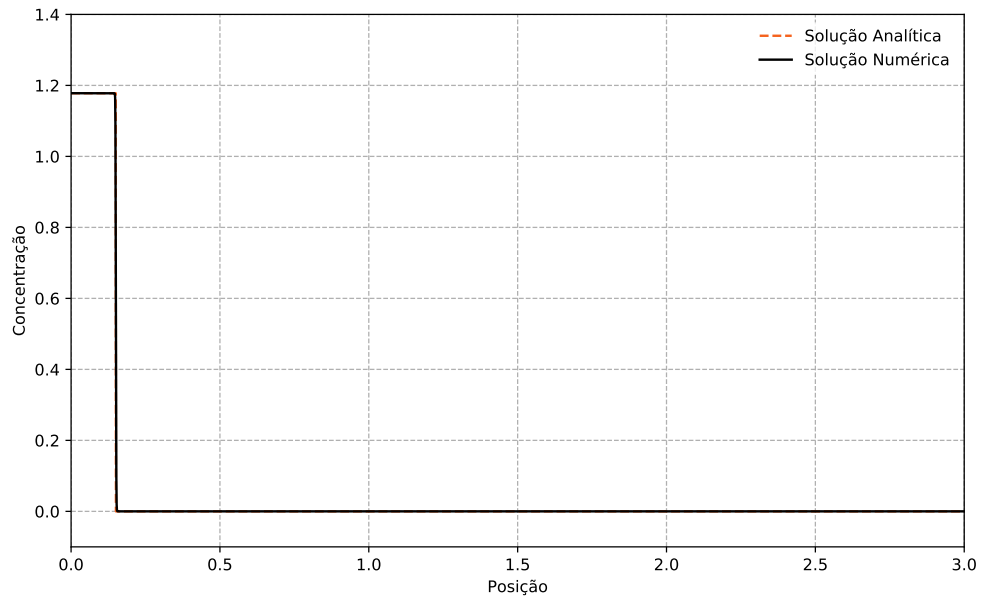
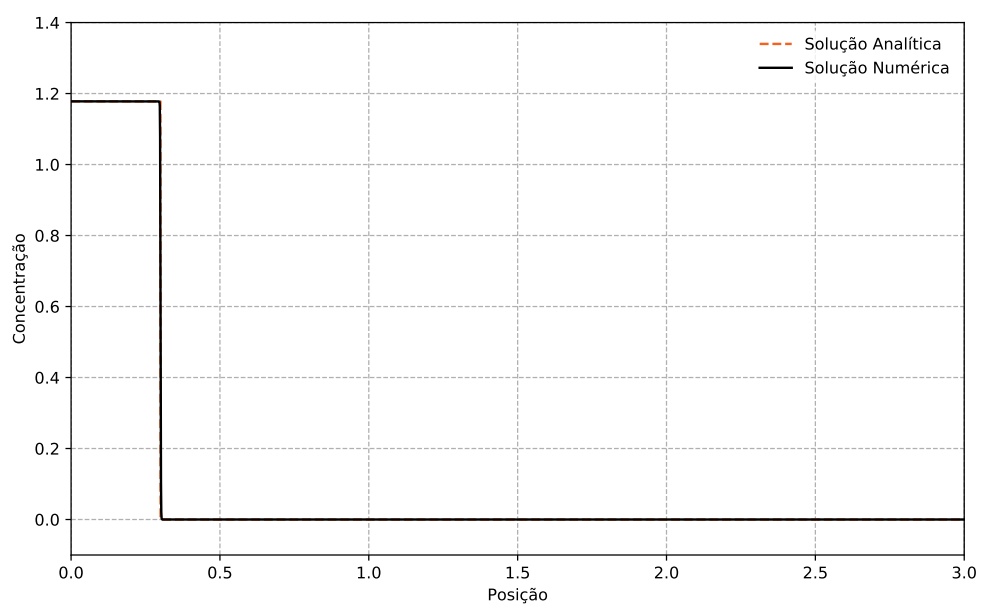
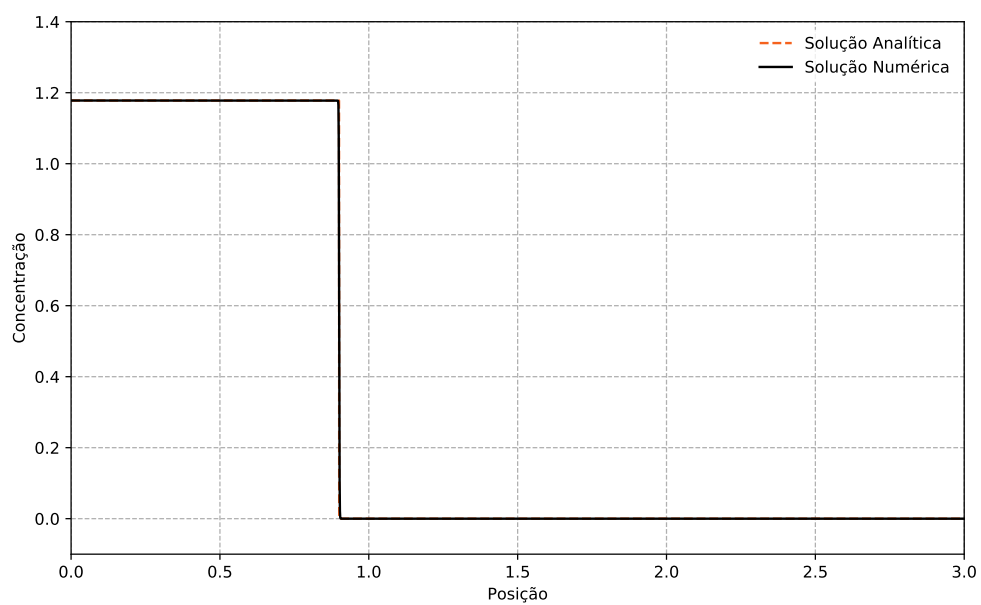


Figura 6 – Limitador Superbee - Segundo Experimento em  $t = 0,25$

Figura 7 – Limitador Superbee - Segundo Experimento em  $t = 0,5$ Figura 8 – Limitador Superbee - Segundo Experimento em  $t = 1,5$

### 3.1.3 Terceiro Experimento do Método TVD com Limitador Superbee

Abaixo apresentamos os resultados do terceiro experimento do Método TVD com limitador Superbee, com tempo total de execução de 12,447 segundos.

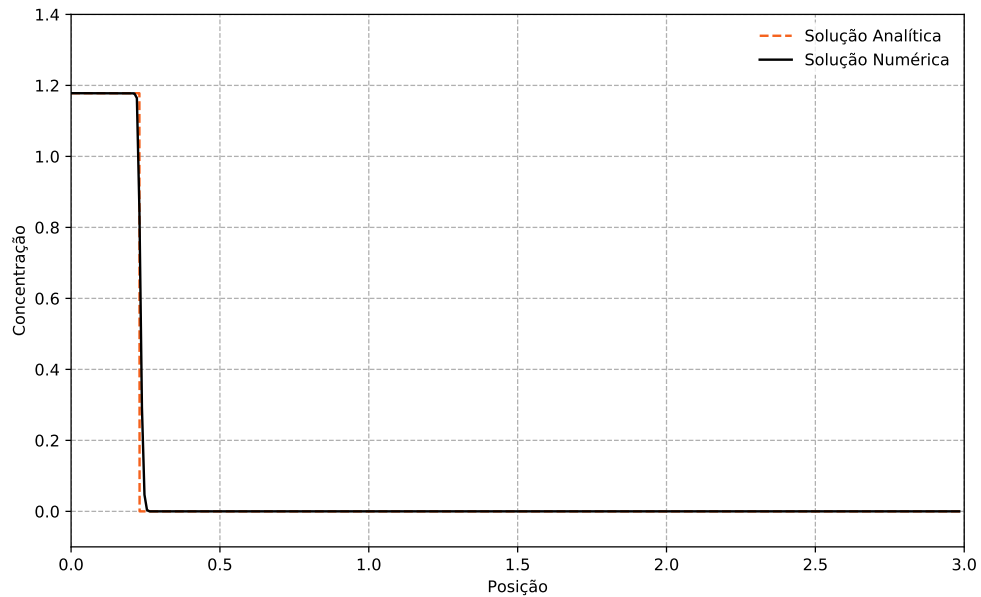
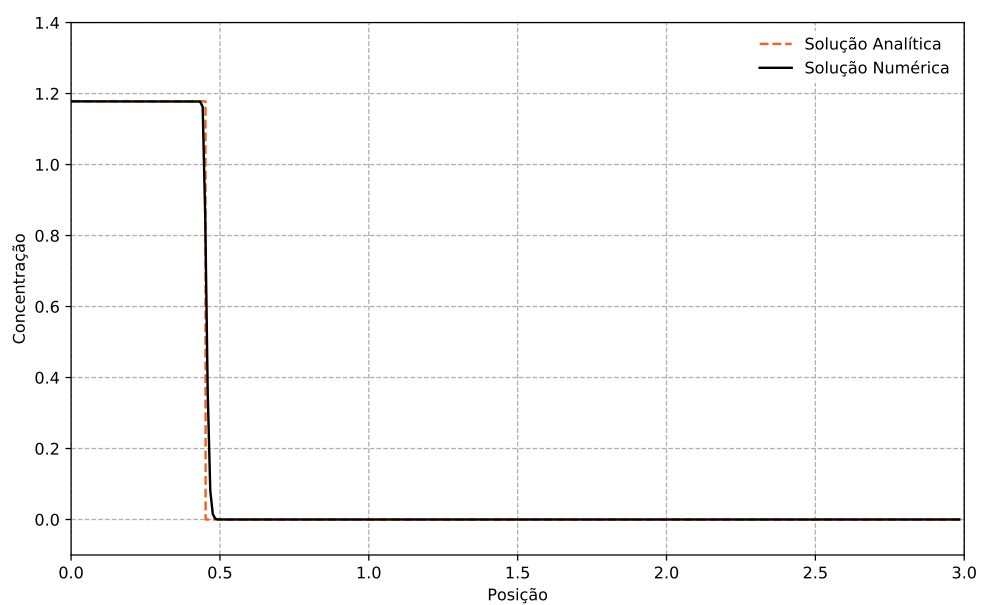
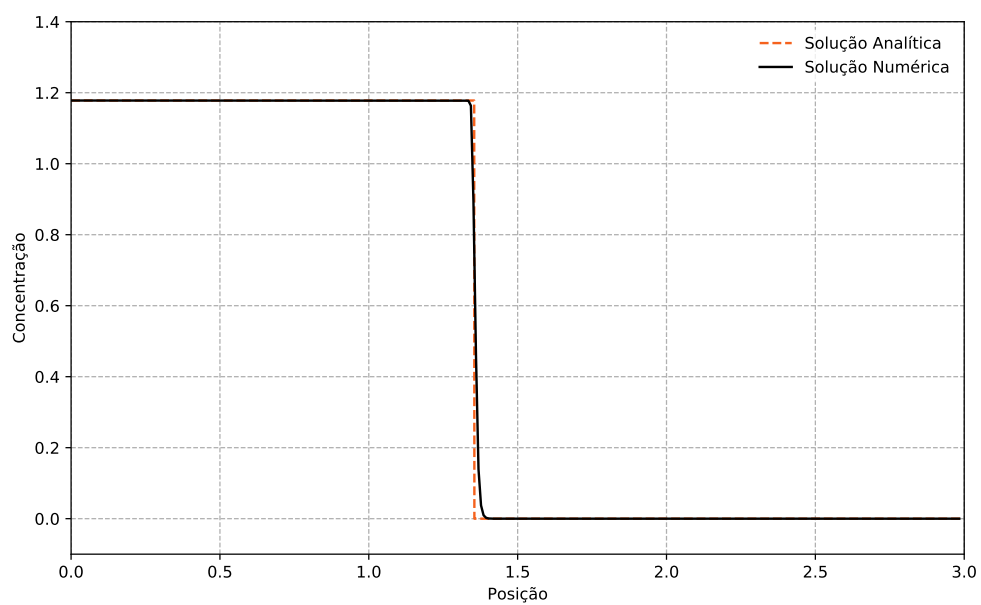


Figura 9 – Limitador Superbee - Terceiro Experimento em  $t = 0,25$

Figura 10 – Limitador Superbee - Terceiro Experimento em  $t = 0,5$ Figura 11 – Limitador Superbee - Terceiro Experimento em  $t = 1,5$



### 3.1.4 Quarto Experimento do Método TVD com Limitador Superbee

Abaixo apresentamos os resultados do quarto experimento do Método TVD com limitador Superbee, com tempo total de execução de 8,454 segundos.

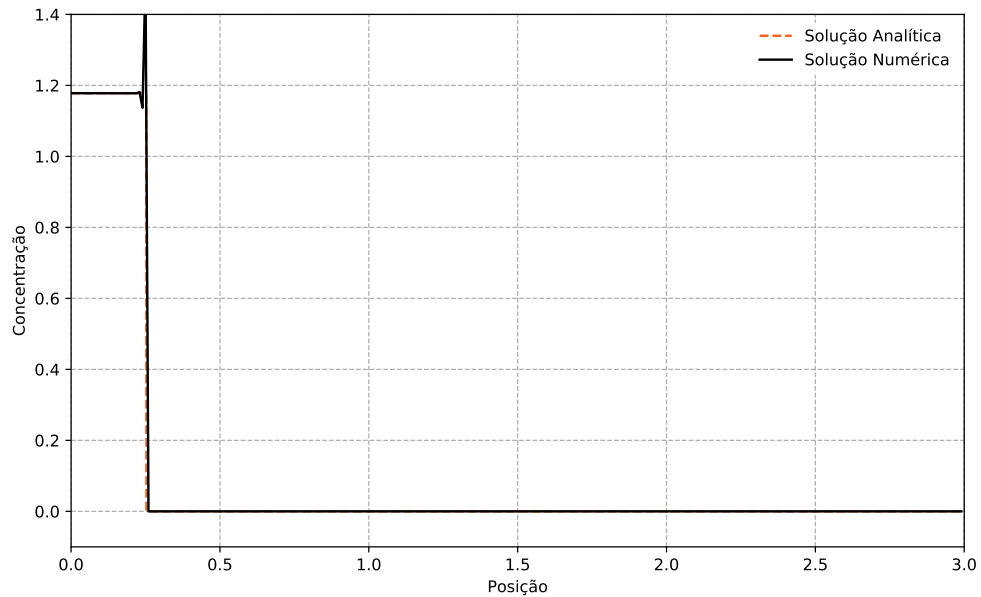
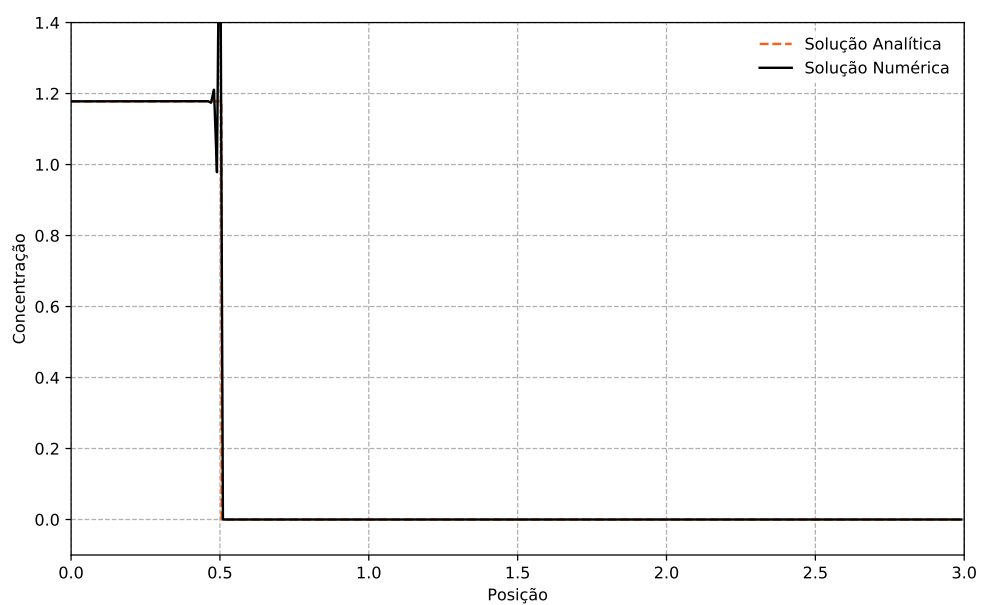
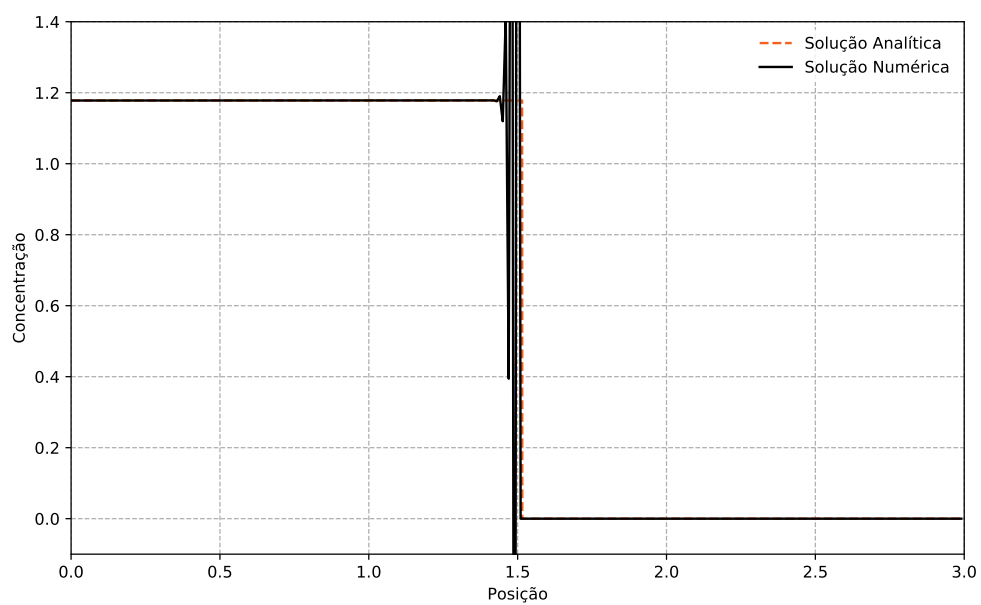


Figura 12 – Limitador Superbee - Quarto Experimento em  $t = 0,25$

Figura 13 – Limitador Superbee - Quarto Experimento em  $t = 0,5$ Figura 14 – Limitador Superbee - Quarto Experimento em  $t = 1,5$

## 3.2 Resultados para o Método TVD com Limitador MC

Na [subseção 3.2.1](#), [subseção 3.1.2](#), [subseção 3.2.3](#) e [subseção 3.2.4](#) são apresentados os resultados para os 4 experimentos realizados para o método TVD com limitador MC tendo em vista os dados da [Tabela 1](#).

### 3.2.1 Primeiro Experimento do Método TVD com Limitador MC

Abaixo apresentamos os resultados do primeiro experimento do Método TVD com limitador MC, com tempo total de execução de 3,177 segundos.

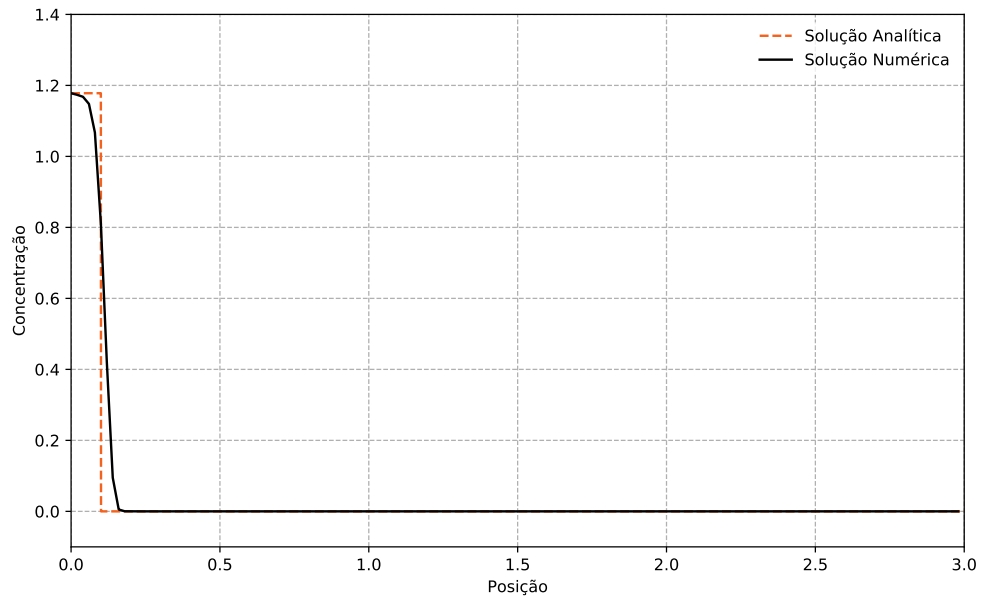
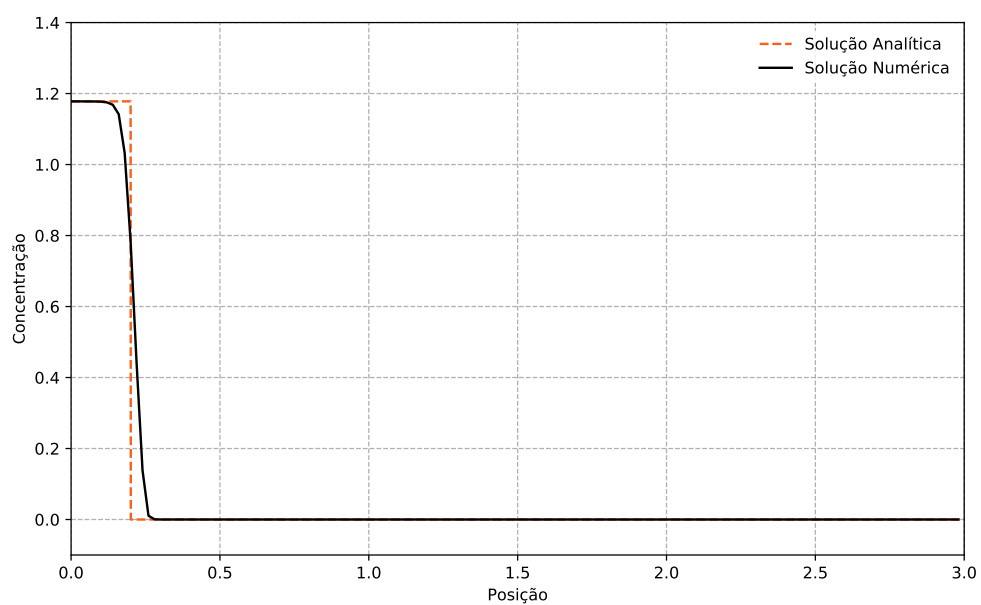
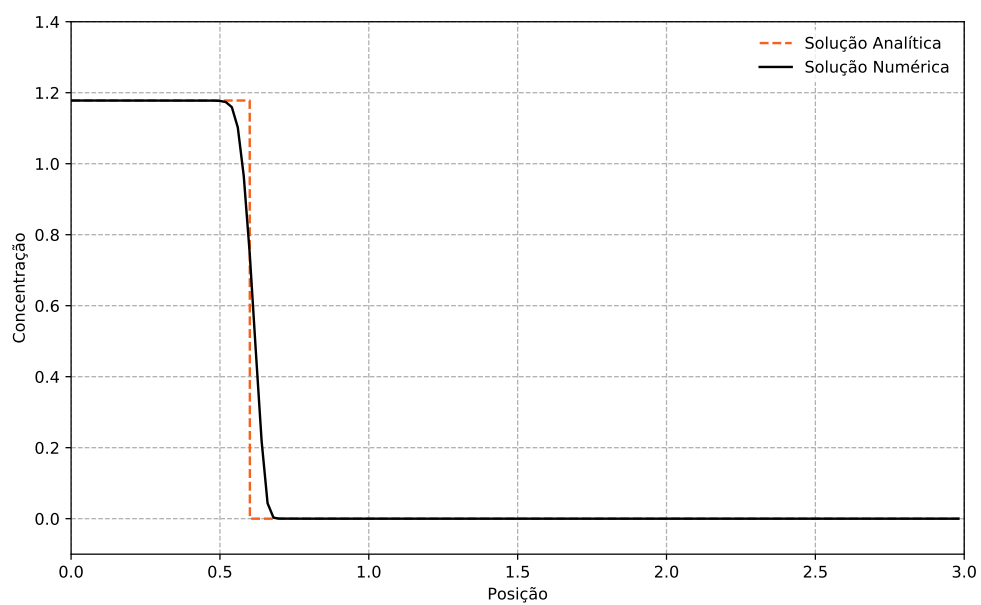
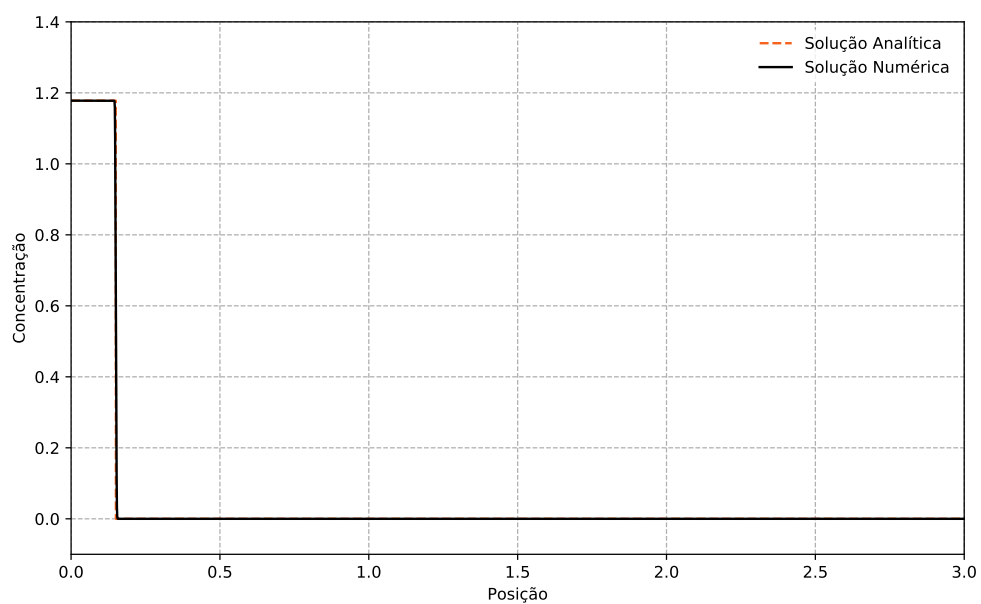


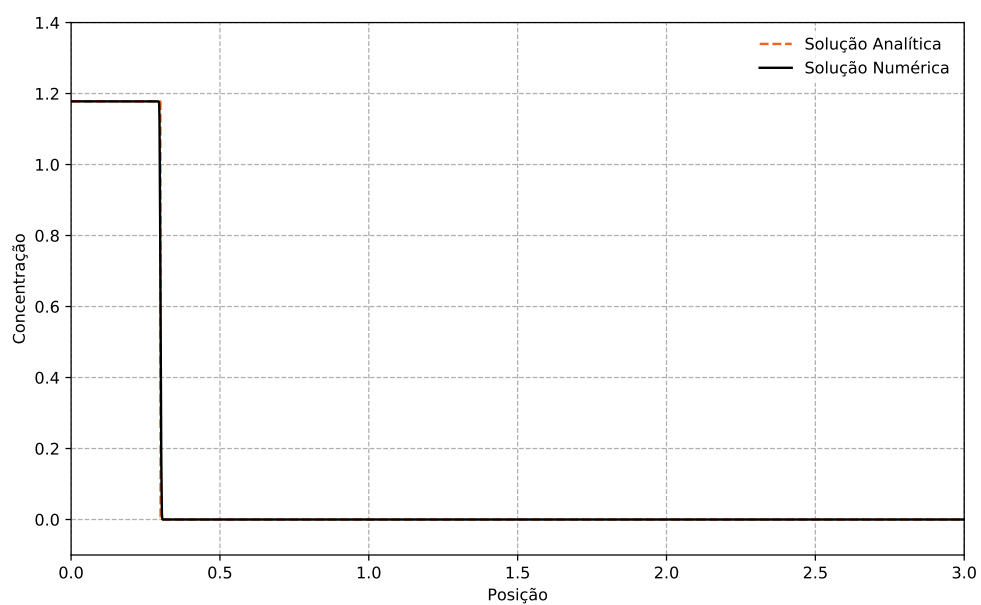
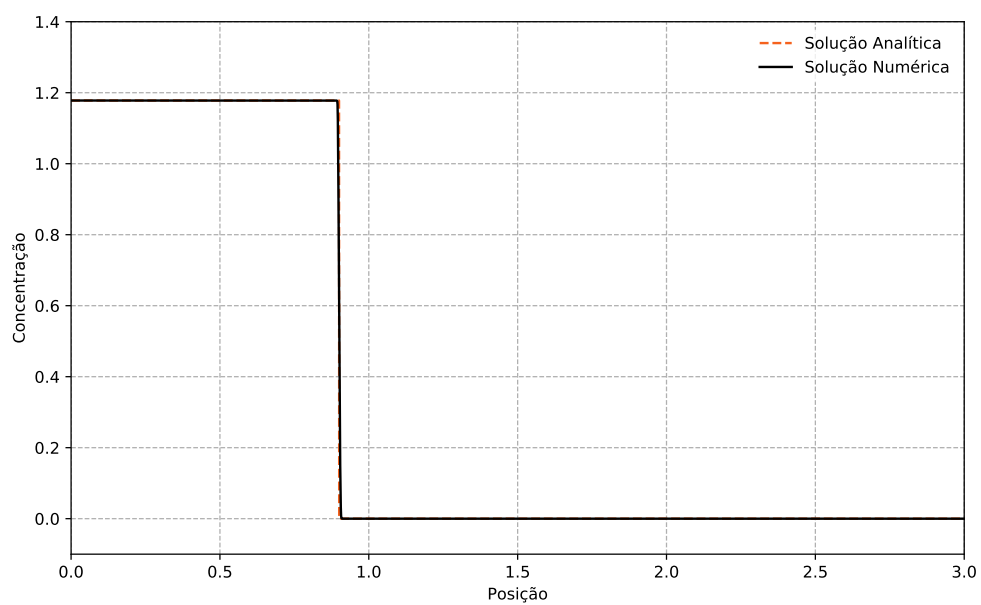
Figura 15 – Limitador MC - Primeiro Experimento em  $t = 0,25$

Figura 16 – Limitador MC - Primeiro Experimento em  $t = 0,5$ Figura 17 – Limitador MC - Primeiro Experimento em  $t = 1,5$

### 3.2.2 Segundo Experimento do Método TVD com Limitador MC

Abaixo apresentamos os resultados do segundo experimento do Método TVD com limitador MC, com tempo total de execução de 659,023 segundos.

Figura 18 – Limitador MC - Segundo Experimento em  $t = 0,25$

Figura 19 – Limitador MC - Segundo Experimento em  $t = 0,5$ Figura 20 – Limitador MC - Segundo Experimento em  $t = 1,5$



### 3.2.3 Terceiro Experimento do Método TVD com Limitador MC

Abaixo apresentamos os resultados do terceiro experimento do Método TVD com limitador MC, com tempo total de execução de 8,281 segundos.

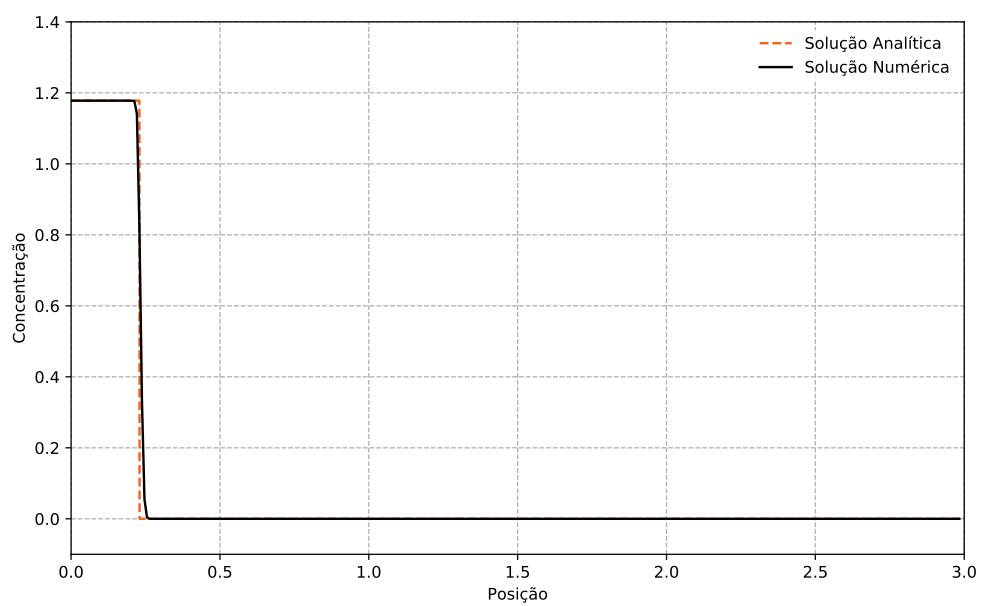
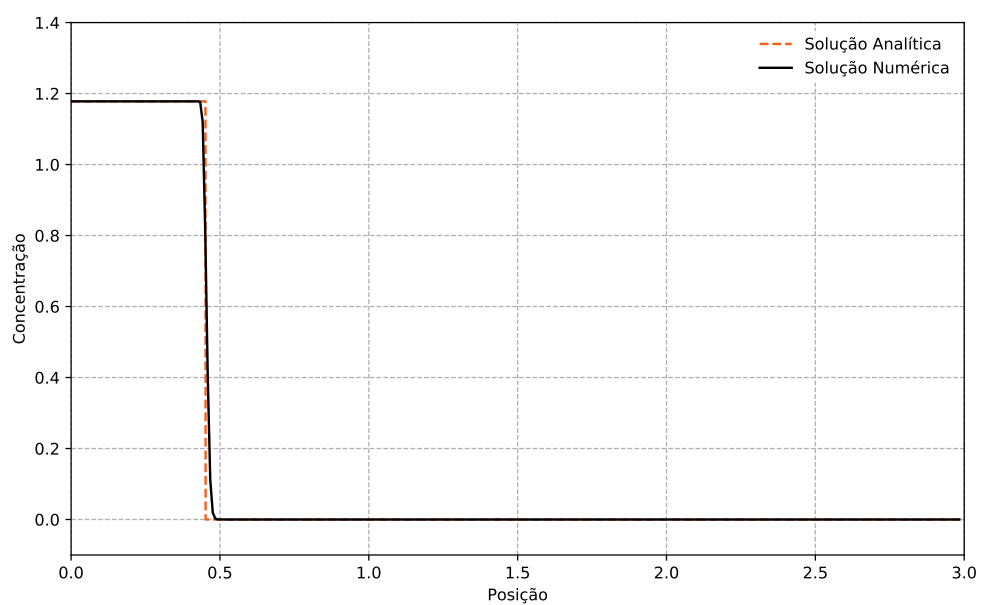
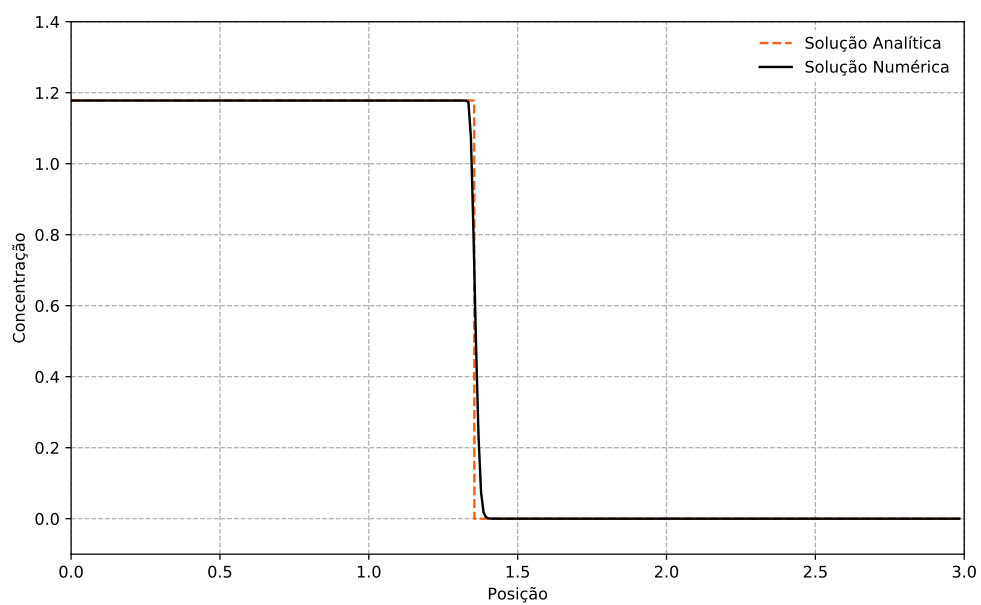
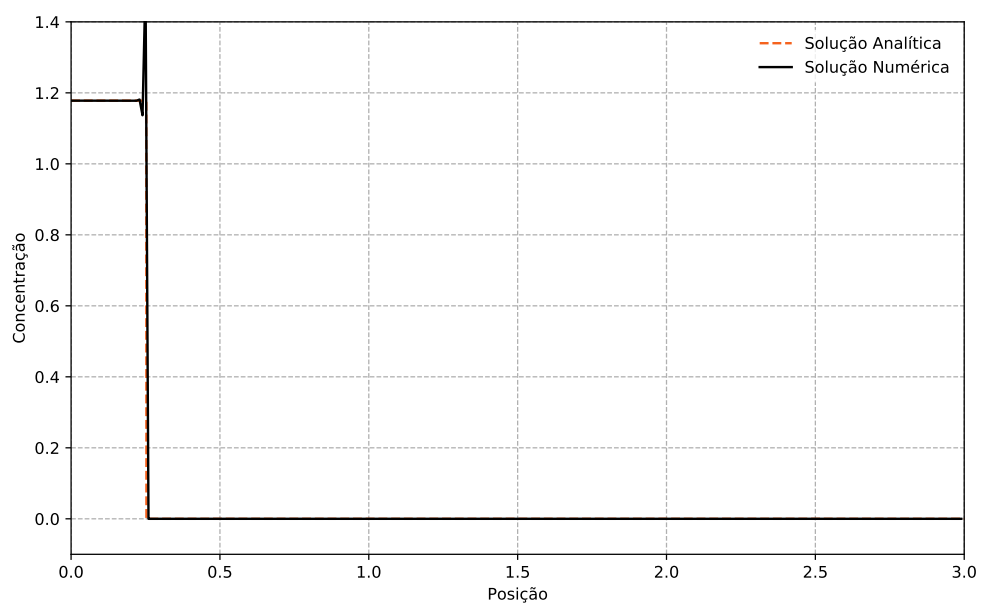


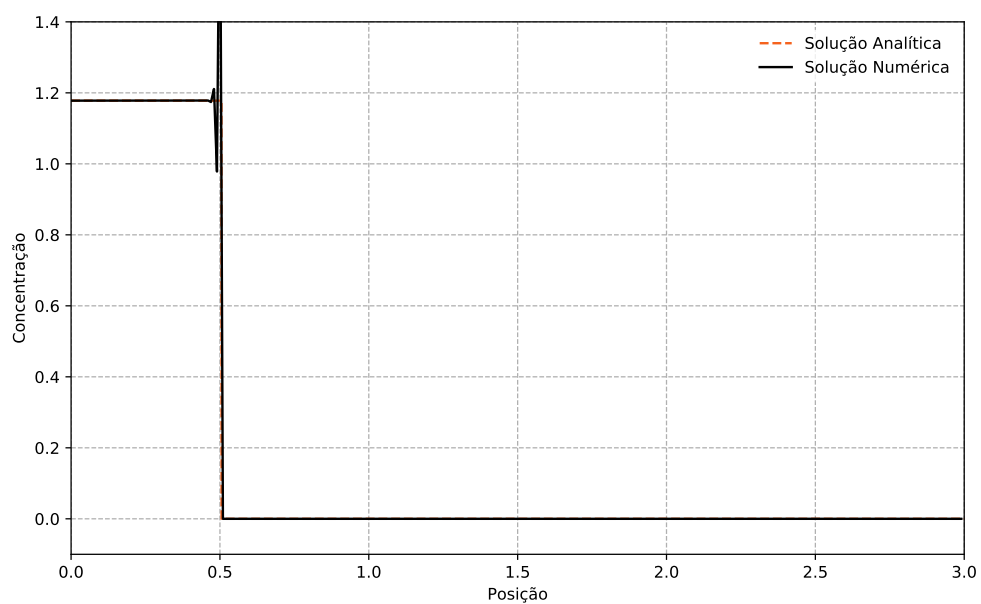
Figura 21 – Limitador MC - Terceiro Experimento em  $t = 0,25$

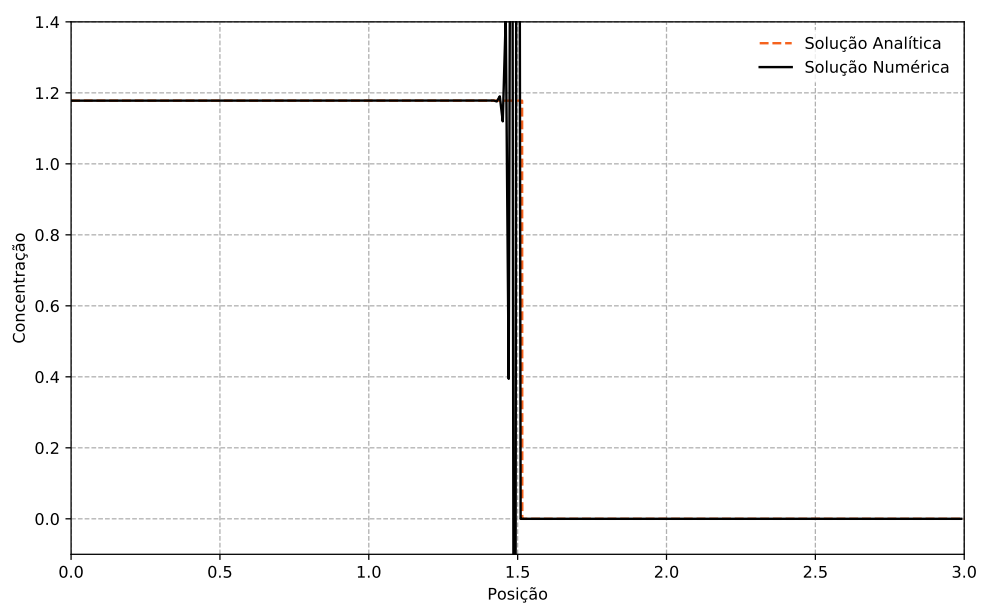
Figura 22 – Limitador MC - Terceiro Experimento em  $t = 0,5$ Figura 23 – Limitador MC - Terceiro Experimento em  $t = 1,5$

### 3.2.4 Quarto Experimento do Método TVD com Limitador MC

Abaixo apresentamos os resultados do quarto experimento do Método TVD com limitador MC, com tempo total de execução de 6,852 segundos.

Figura 24 – Limitador MC - Quarto Experimento em  $t = 0,25$

Figura 25 – Limitador MC - Quarto Experimento em  $t = 0,5$

Figura 26 – Limitador MC - Quarto Experimento em  $t = 1,5$

### 3.3 Resultados para o Método TVD com Limitador van Leer

Na [subseção 3.3.1](#), [subseção 3.3.2](#), [subseção 3.3.3](#) e [subseção 3.3.4](#) são apresentados os resultados para os 4 experimentos realizados para o método TVD com limitador van Leer tendo em vista os dados da [Tabela 1](#).



### 3.3.1 Primeiro Experimento do Método TVD com Limitador van Leer

Abaixo apresentamos os resultados do primeiro experimento do Método TVD com limitador van Leer, com tempo total de execução de 1,385 segundos.

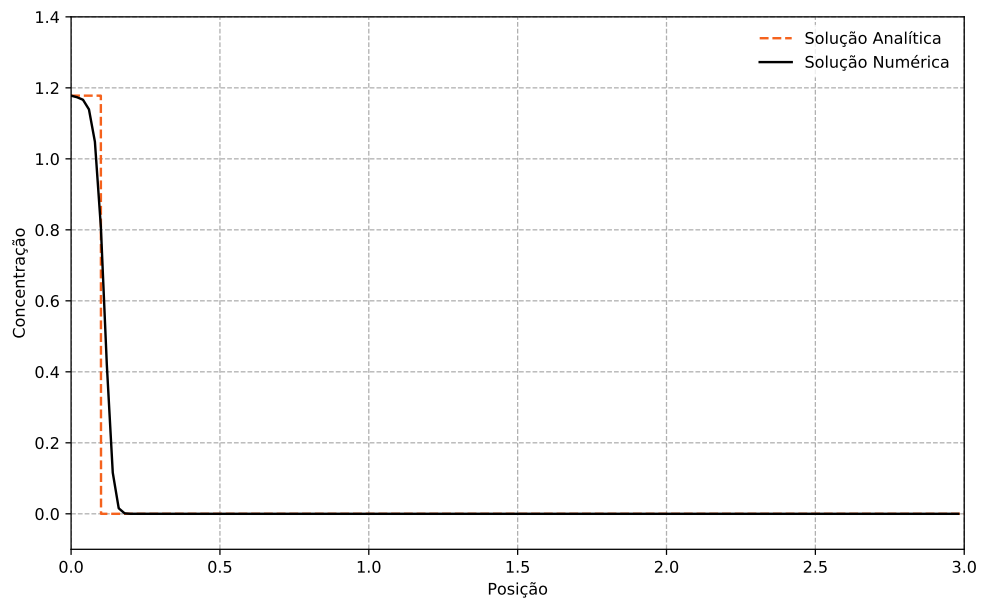
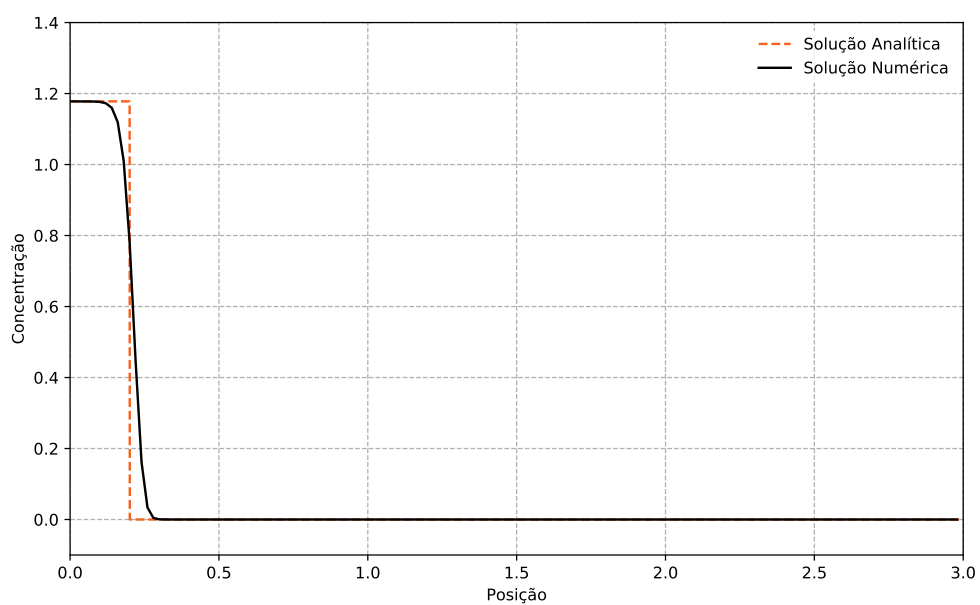
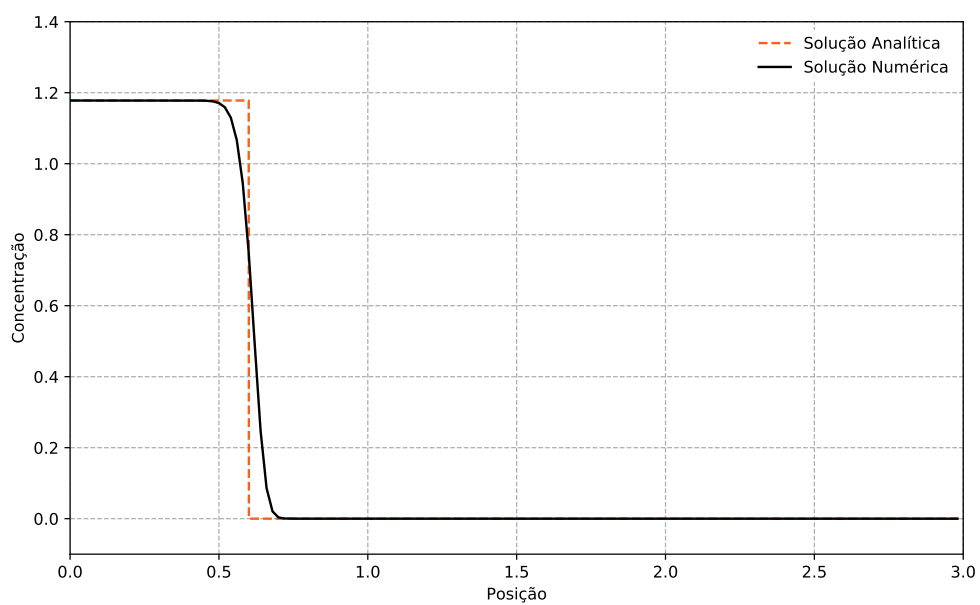


Figura 27 – Limitador van Leer - Primeiro Experimento em  $t = 0,25$

Figura 28 – Limitador van Leer - Primeiro Experimento em  $t = 0,5$ Figura 29 – Limitador van Leer - Primeiro Experimento em  $t = 1,5$

### 3.3.2 Segundo Experimento do Método TVD com Limitador van Leer

Abaixo apresentamos os resultados do segundo experimento do Método TVD com limitador van Leer, com tempo total de execução de 253,646 segundos.

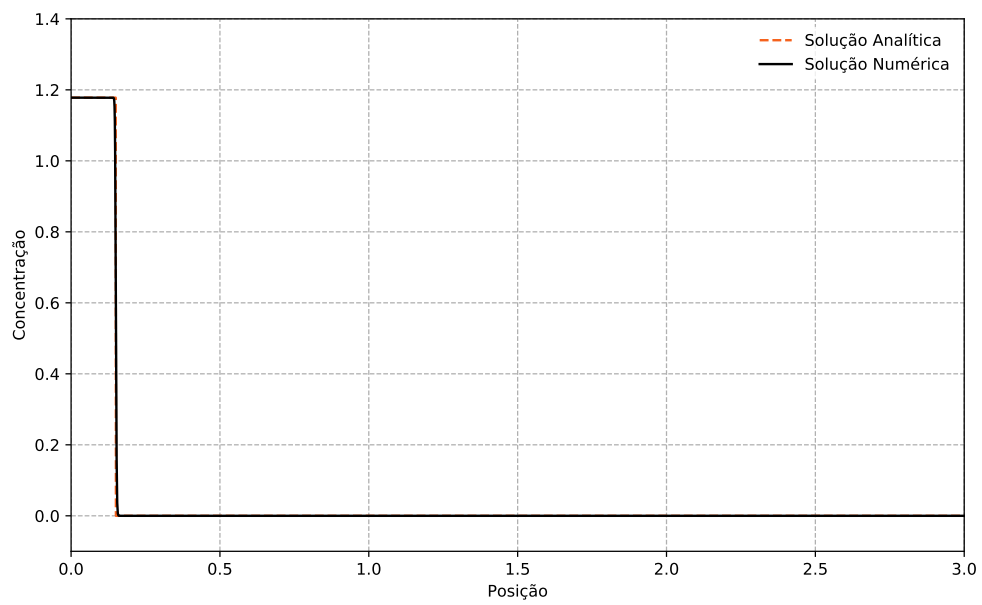
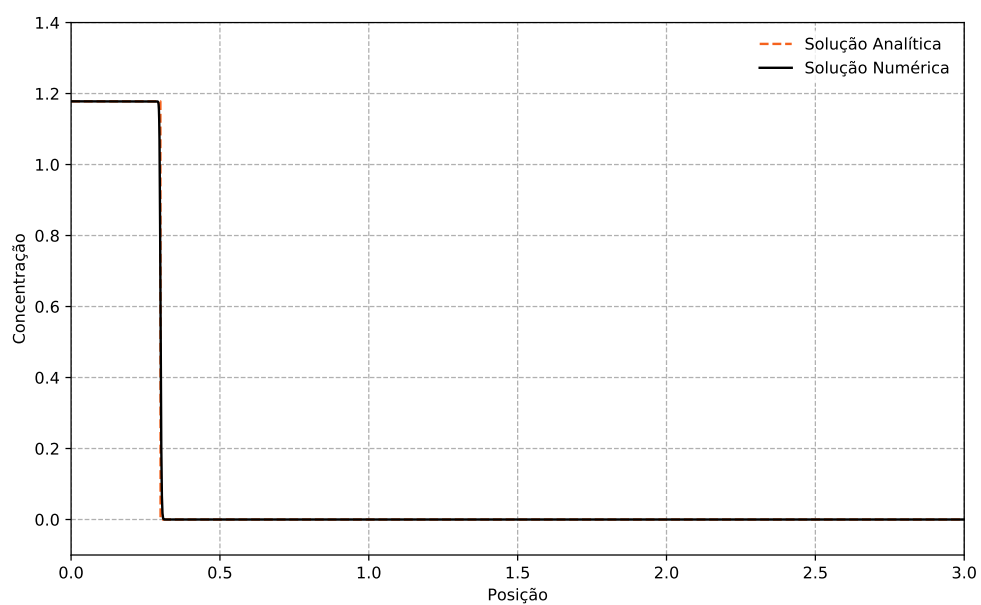
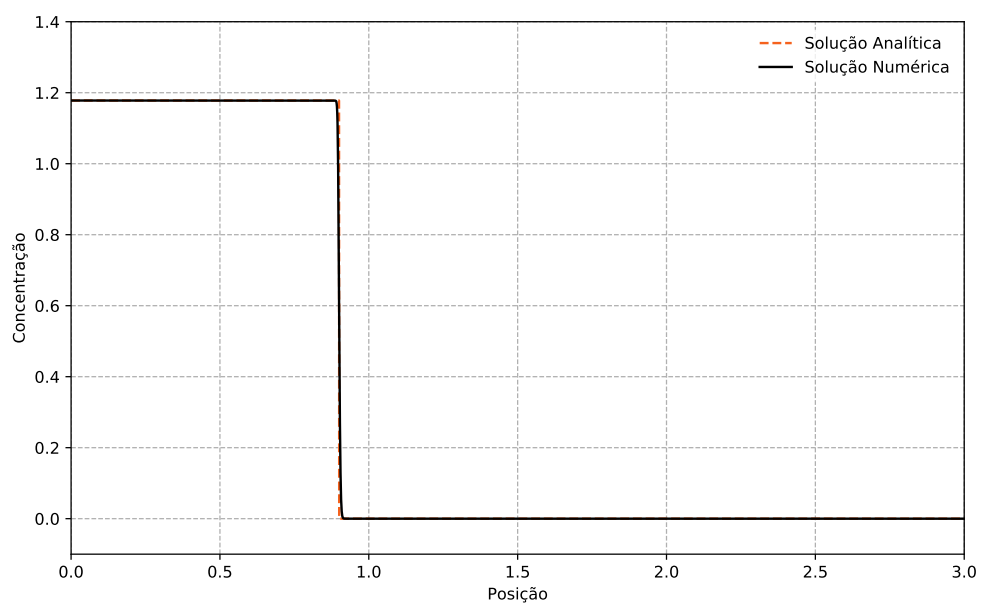
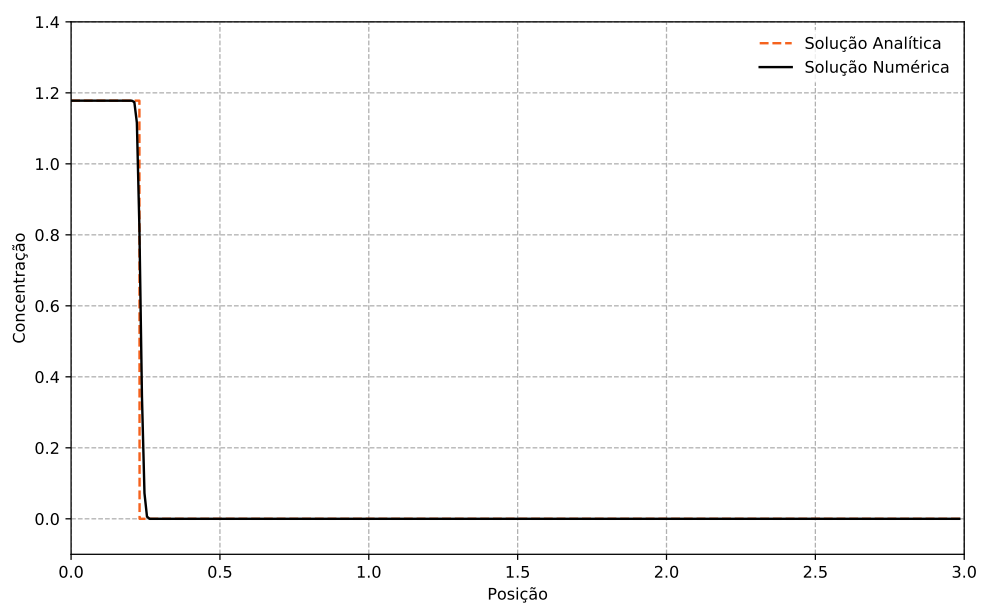


Figura 30 – Limitador van Leer - Segundo Experimento em  $t = 0,25$

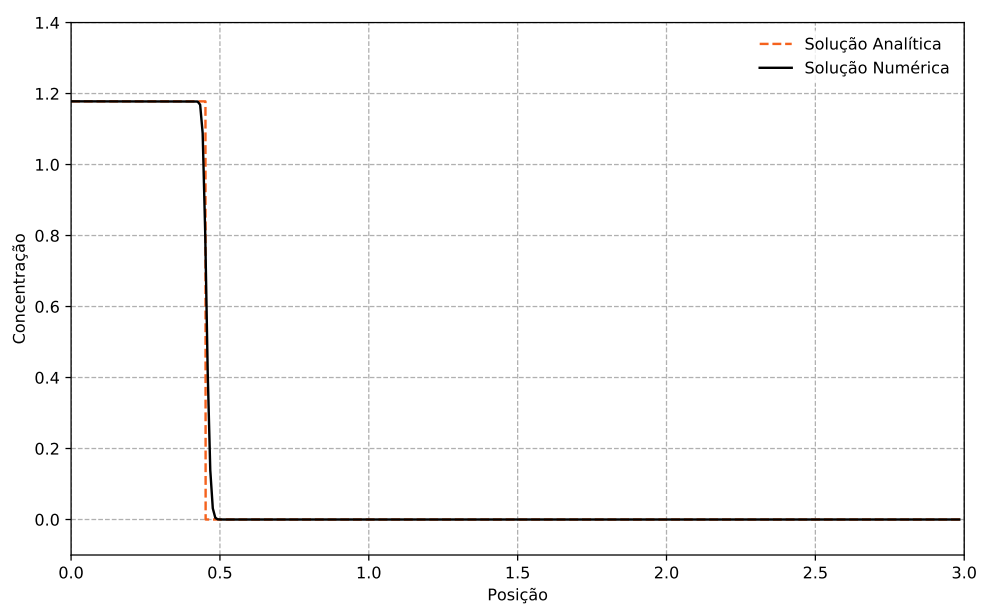
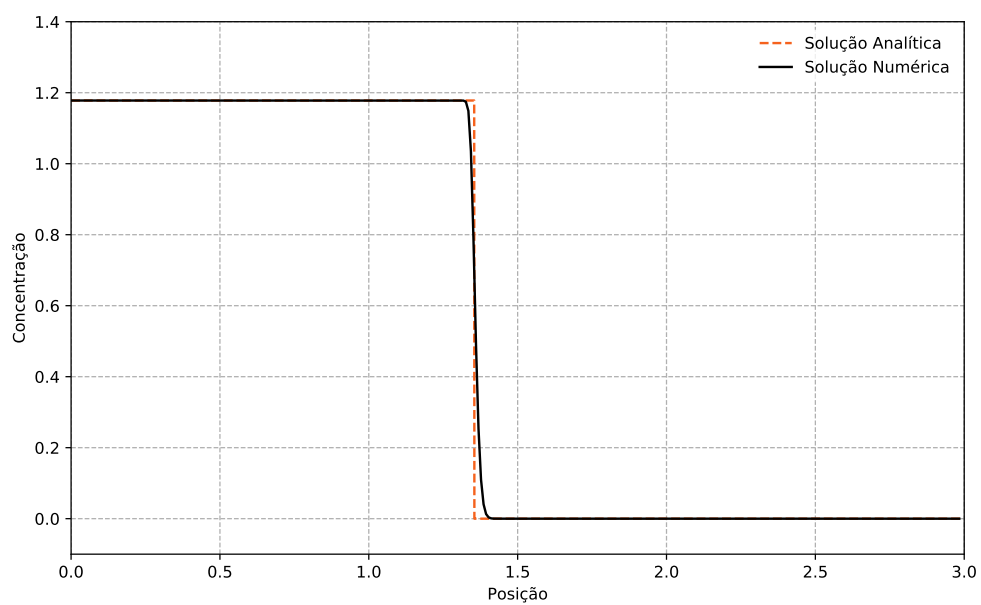
Figura 31 – Limitador van Leer - Segundo Experimento em  $t = 0,5$ Figura 32 – Limitador van Leer - Segundo Experimento em  $t = 1,5$

### 3.3.3 Terceiro Experimento do Método TVD com Limitador van Leer

Abaixo apresentamos os resultados do terceiro experimento do Método TVD com limitador van Leer, com tempo total de execução de 3,203 segundos.

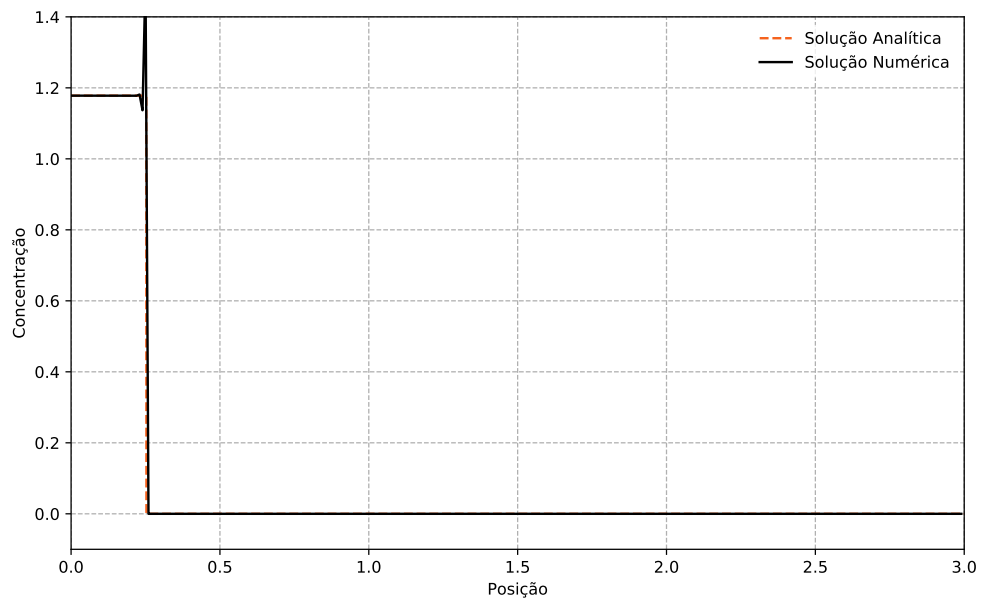
Figura 33 – Limitador van Leer - Terceiro Experimento em  $t = 0,25$

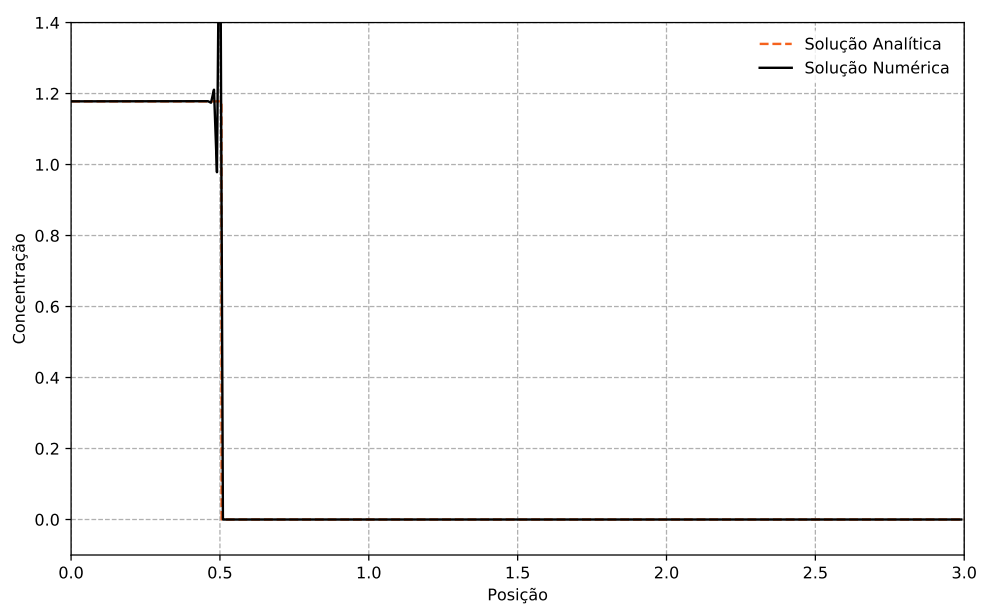
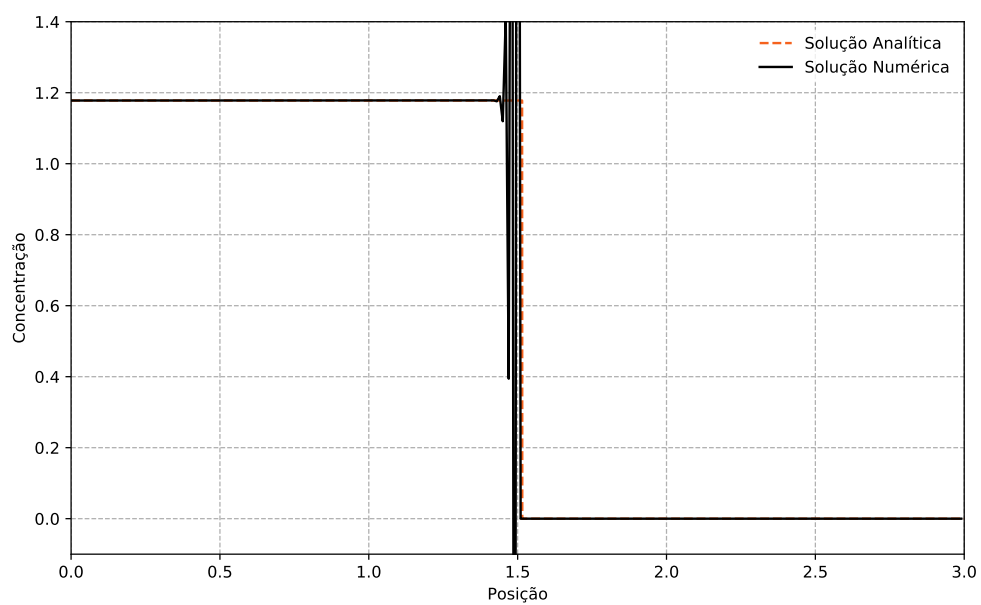


Figura 34 – Limitador van Leer - Terceiro Experimento em  $t = 0,5$ Figura 35 – Limitador van Leer - Terceiro Experimento em  $t = 1,5$

### 3.3.4 Quarto Experimento do Método TVD com Limitador van Leer

Abaixo apresentamos os resultados do quarto experimento do Método TVD com limitador van Leer, com tempo total de execução de 2,469 segundos.

Figura 36 – Limitador van Leer - Quarto Experimento em  $t = 0,25$

Figura 37 – Limitador van Leer - Quarto Experimento em  $t = 0,5$ Figura 38 – Limitador van Leer - Quarto Experimento em  $t = 1,5$

## 4 Discussão

Conforme podemos observar através dos resultados do [Capítulo 3](#) os métodos TVD obtiveram uma aproximação muito acurada em todas as variações da malha espacial e de passos de tempo, mesmo que intencionalmente o método Lax-Wendroff tenha sido utilizado quando os valores para  $\psi(\theta)$  de cada limitador não puderam ser avaliados.

Além disso não ocorreram oscilações espúrias, mesmo na presença da descontinuidade que caracteriza o Problema do Traçador, onde a mesma avança como uma onda, e que seria um ponto crítico para o surgimento dessas oscilações como nos casos da utilização por exemplo do método Lax-Wendroff que pode ser visto na [Figura 39](#).

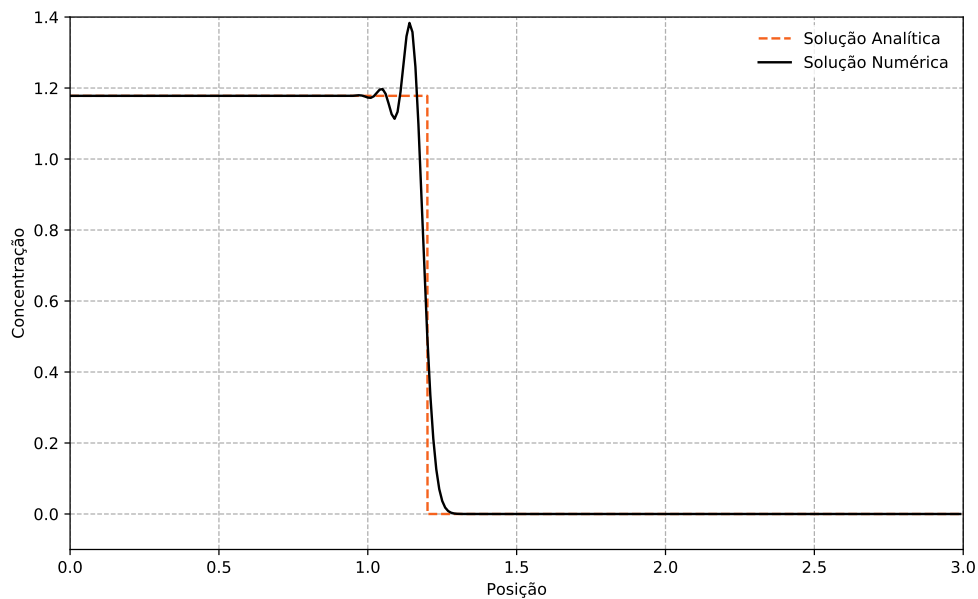


Figura 39 – Exemplo de Oscilação Espúria para o Problema do Traçador obtida através do Método não TVD Lax-Wendroff

Entretanto quando avaliamos cada método em relação ao tempo de computação usando os dados da [Tabela 3](#) vemos que em todos os 4 experimentos o método TVD que utiliza o limitador de fluxo van Leer teve um custo computacional menor, levando menos tempo para concluir o processamento.

Experimento	$\Delta t$	$\Delta x$	Superbee	MC	van Leer
1	0,01	0,02	4,241	3,177	1,385
2	0,001	0,001	910,025	659,023	253,646
3	0,0085	0,0085	12,447	8,281	3,203
4	0,01	0,01	8,454	6,852	2,469

Tabela 3 – Tempo de Execução em Segundos para cada Método e Experimento

Além da análise realizada levando em conta o custo computacional e o refinamento das malhas, podemos avaliar as curvas numéricas de cada método TVD em estudo, para cada experimento em um determinado tempo, sobrepostas e acompanhadas da solução analítica e da curva numérica de um método UpWind de primeira ordem obtida com as mesmas configurações.

Essa comparação pode ser vista nas [Figura 40](#), [Figura 41](#) e [Figura 42](#).

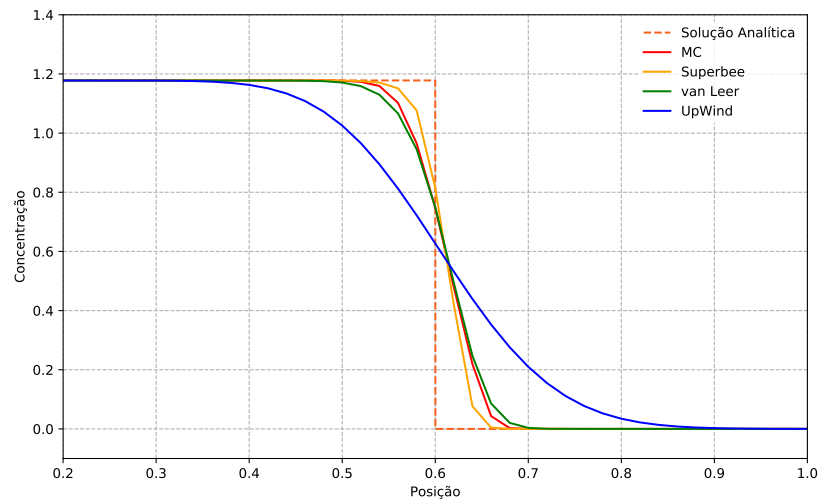


Figura 40 – Comparação dos Métodos TVD e da Solução Analítica com o Método UpWind para o Primeiro Experimento

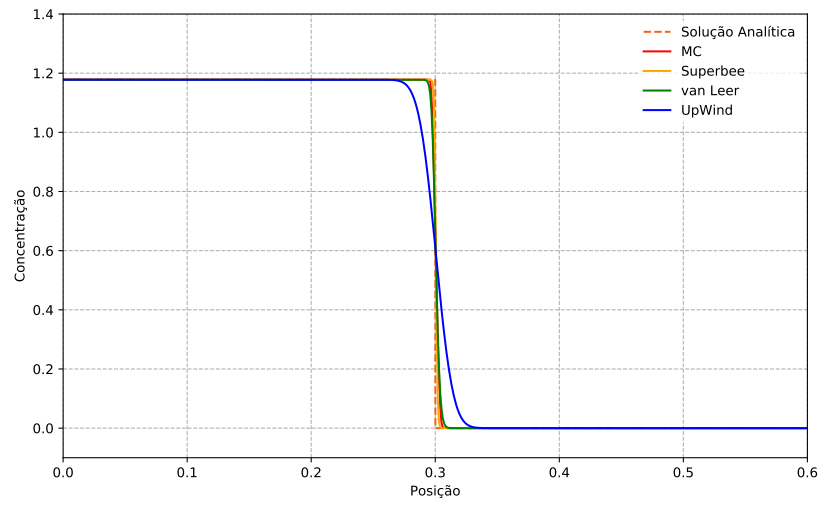


Figura 41 – Comparação dos Métodos TVD e da Solução Analítica com o Método UpWind para o Segundo Experimento

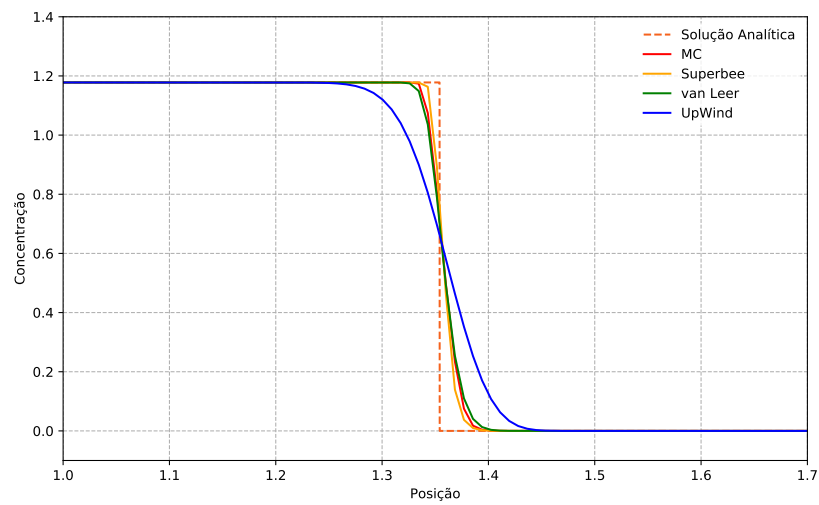


Figura 42 – Comparação dos Métodos TVD e da Solução Analítica com o Método UpWind para o Terceiro Experimento

## 5 Conclusão

De acordo com os resultados obtidos no [Capítulo 3](#) e através da comparação dos mesmos com a curva analítica e com o método TVD UpWind feita no [Capítulo 4](#), pôde-se confirmar a característica dos métodos TVD de eliminar oscilações espúrias e de obter sempre que possível uma aproximação de segunda ordem.

Ainda através da comparação pôde-se ver a influência da difusão numérica de um método de primeira ordem sobre a solução mesmo que este seja um método TVD, como pode ser observado na [Figura 40](#), [Figura 41](#) e [Figura 42](#), onde houve uma suavização que afasta o método UpWind da solução analítica e das demais soluções que utilizam os limitadores Superbee, MC e van Leer.

Além disso podemos concluir que a utilização dos métodos TVD estudados não necessitam de um alto refinamento da malha para a obtenção de boas soluções, e que se, ainda assim um alto refinamento seja necessário, limitadores como o Superbee terão um alto custo computacional, podendo levar cerca de 72.12% a mais de tempo em comparação com o limitador van Leer e de 27.58% mais tempo, se comparado ao limitador MC.



## Referências

MAZZIA, A. *Numerical Methods for the solution of Hyperbolic Conservation Laws*. 2012. Disponível em: <[http://www.math.unipd.it/~ancona/pdf/others-papers/numerical\\_methods\\_for\\_conlaws.pdf](http://www.math.unipd.it/~ancona/pdf/others-papers/numerical_methods_for_conlaws.pdf)>. Acesso em: 12 ago 2018. Citado na página 10.

SOUTO, H. P. A. *Métodos Numéricos para Equações Diferenciais II*. Nova Friburgo - Rio de Janeiro: Helio Pedro Amaral Souto, 2017. Citado na página 13.

## Anexos

# ANEXO A – Código Python para os Métodos TVD com Limitadores de Fluxo Superbee, MC e van Leer

```
#!/usr/bin/python

"""
Author: André Savioli
Email: andre_savioli@hotmail.com
"""

import time;
import math;
import numpy as np;
import matplotlib.pyplot as plt
import pylab

# Constants

L = 3

DX = 0.0085;

DT = 0.0085;

U = 0.9;

analyticSolutionResolutionToPlot = 1000

# Select Limiter

limiter = ('van-leer', 'van Leer');
# limiter = ('superbee', 'Superbee');
# limiter = ('mc', 'Monitonized Central Difference' );
# limiter = ( 'upwind', 'UpWind' );
```

```
# limiter = ( 'lax-wendroff', 'Lax-Wendroff' );
# limiter = ( 'beam-warming', 'Beam-Warming' );
# limiter = ( 'fromm', 'Fromm' );

# Select Phi In Case of NaN

# phiInCaseOfNaN = ( 'upwind', 'UpWind' );
phiInCaseOfNaN = ( 'lax-wendroff', 'Lax-Wendroff' );
# phiInCaseOfNaN = ( 'upwindw', 'UpWind' );

# Image Format Output
# ( extension, matplotlib format )
imageFormat = ('pdf','pdf')
# imageFormat = ('png','png')

# Courant Number

CFL = U * ( DT / DX )

shots = [ 0.25, 0.5, 1.5 ];

maxTime = 5;

# Problem

# Values
C = 1.178

cZero = 0

# Index from 0 to totalSizeParts - 1

totalSizeParts = int( L / DX );

# Index from 0 to totalTimeParts - 1

totalTimeParts = int( maxTime / DT );

totalTime = totalTimeParts * DT
```

```
print( "Total Time: {0:.3f}".format( totalTime ) )
print( "Total of Volumes: {0}".format( totalSizeParts ) )

# Matrix to store the numeric solution

Qs = np.zeros( shape = ( totalTimeParts, totalSizeParts ), dtype = "float64" );

# Analytic Solution

def initialCondition( x ):

    result = cZero;

    return result;

# Analytic Solution

def analyticSolution( x, t ):

    if( x < U*t ):

        result = C

    elif( x > U*t and L < U*t ):

        result = 0

    else:

        result = cZero

    return result;

# Numeric Solution

def thetaL( i, x ):

    if( x > 2 ):
```

---

```
num = Qs[ i ] [ x-1 ] - Qs[ i ] [ x-2 ]
den = Qs[ i ] [ x ] - Qs[ i ] [ x-1 ]

if( den > 0. or den < 0. ):

    return ( num ) / ( den );

else:

    return 0;

else:

    return 0;

def thetaR( i, x ):

    if( x > 1 and ( x < totalSizeParts -1 ) ):

        num = Qs[ i ] [ x ] - Qs[ i ] [ x - 1 ]
        den = Qs[ i ] [ x + 1 ] - Qs[ i ] [ x ]

        if( den > 0. or den < 0. ):

            return ( num ) / ( den );

        else:

            return 0;

    else:

        return 0;

def phi( theta ):

    # High-Resolution Limiters

    # van Leer Limiter
```

```
if( limiter[0] == 'van-leer' ):

    phi = ( theta + np.abs( theta ) ) \
           / ( 1 + np.abs( theta ) )

# Superbee Limiter

elif( limiter[0] == 'superbee' ):

    phi = np.max( [ 0, np.min( [ 1, 2 * theta ] ), np.min( [ 2, theta ] ) ] )

# MC - Monitonzed Central Difference

elif( limiter[0] == 'mc' ):

    phi = np.max( [ 0, np.min( [ ( 1 + theta ) / 2, 2, 2 * theta ] ) ] )

# Linear Limiters

# UpWind Limiter

elif( limiter[0] == 'upwind' ):

    phi = 0

# Lax-Wendroff Limiter

elif( limiter[0] == 'lax-wendroff' ):

    phi = 1

# Beam-Warming Limiter

elif( limiter[0] == 'beam-warming' ):

    phi = theta

# Fromm Limiter
```

```
elif( limiter[0] == 'fromm' ):

    phi = 0.5 * ( 1 + theta )

    # Check if phi is NaN, if True we use
    # Linear UpWind Limiter or Linear Lax-Wendroff Limiter

    if( np.isnan( phi ) ):

        if( phiInCaseOfNaN[0] == 'upwind' ):

            return 0;

        elif( phiInCaseOfNaN[0] == 'lax-wendroff' ):

            return 1;

    else:

        return phi;

def boundaryCondition( i, x ):

    Qs[ i ] [ 0 ] = C
    Qs[ i ] [ totalSizeParts -1 ] = 0

def TVD( i, x ):

    # Limits Verification

    # For Previous x

    if( x - 1 < 0 ):

        xPrevious = 0

    else:
```



---

```
xPrevious = x - 1

    if( x >= totalSizeParts -1 ):

        xNext = totalSizeParts -1

    else:

        xNext = x + 1

    phiR = phi( thetaR( i,x ) );
    phiL = phi( thetaL( i,x ) );

    Q = Qs[ i ] [ x ] - ( CFL * ( Qs[ i ] [ x ] - Qs[ i ] [ xPrevious ] ) ) \
        -0.5 * CFL * ( 1 - CFL ) * \
        ( phiR * ( Qs[ i ] [ xNext ] - Qs[ i ] [ x ] ) \
        - phiL * ( Qs[ i ] [ x ] - Qs[ i ] [ xPrevious ] ) );

    return Q;

startTime = time.time()

# Initial Condition

for x in range( 0, totalSizeParts ):

    Qs[ 0 ] [ x ] = initialCondition( x * DX );

boundaryCondition( 0, None );

# For each time
for i in range( 0, totalTimeParts ):

    if( i == ( totalTimeParts - 1 ) ):
        break;

    # For each position
    for x in range( 0, totalSizeParts ):
```

```
        boundaryCondition( i + 1, None );

        Qs[ i + 1 ][ x ] = TVD( i, x );

xArray = [ DX*i for i in range( 0, totalSizeParts ) ]
xnArray = [ DX/analyticSolutionResolutionToPlot*i for i in range( 0, totalSizeParts ) ]

endTime = time.time();

executionTime = endTime - startTime;

print('Total Execution Time: {0:.3f}'.format( executionTime ), 's' );

for p in shots:

    p = int(math.ceil(p/DT));

    if( p > totalTimeParts ):
        p = totalTimeParts-1;

    analyticSolutionData = [ analyticSolution( i * DX/analyticSolutionResolutionToPlot for i in range( 0, totalSizeParts ) ) ]

    numericalData = Qs[ p ];

    fig, ax1 = plt.subplots()

    plt.axis([0, L, -0.1, 1.4])

    color = (245/255, 98/255, 29/255, 1)
    ax1.set_xlabel('Posição')
    ax1.set_ylabel('exp', color=color)
    ax1.plot(xnArray, analyticSolutionData, linestyle='dashed', color=color, label='Solução Analítica')
    ax1.tick_params(axis='y', labelcolor=color)

    color = (0,0,0,1)
    ax1.set_ylabel('Concentração', color=color)
    ax1.plot(xArray, numericalData, linestyle='solid', color=color, label = 'Solução Numérica')
    ax1.tick_params(axis='y', labelcolor=color)
```

```
ax1.legend( [ 'Solução Analítica', 'Solução Numérica' ], loc='upper right', fra
ax1.grid( linestyle = 'dashed' )
ax1.legend().get_frame().set_linewidth(00.)

F = pylab.gcf()

F.set_size_inches( ( 10, 6 ) )

plt.savefig( "{0} - SHOT {1} - Courant Number = {2} - dT = {3} - dX = {4} - U =
            .format( limiter[1],
                    p, CFL, DT, DX, U, p * DT, p * DX, imageFormat[0]),
            format = imageFormat[1],
            dpi = 300,
            transparent = False )

plt.savefig( "{0} - SHOT {1} - Courant Number = {2} - dT = {3} - dX = {4} - U =
            .format( limiter[1],
                    p, CFL, DT, DX, U, p * DT, p * DX, imageFormat[0]),
            format = 'png',
            dpi = 300,
            transparent = False )

plt.show()

np.savetxt( "{0} - SHOT {1} - Courant Number = {2} - dT = {3} - dX = {4} - U = {5}
            .format( limiter[1],
                    p, CFL, DT, DX, U, p * DT, p * DX, executionTime ),
            Qs, delimiter=",")
```