

**UNIVERSIDADE DE SÃO PAULO ESCOLA POLITÉCNICA**

**DEPARTAMENTO DE ENGENHARIA MECÂNICA**

**Simulação de um tráfego veicular através de um modelo de duas equações  
embasado na metodologia dos *Bond Graphs***

Tiago de Freitas Silveira

São Paulo

2009

**UNIVERSIDADE DE SÃO PAULO ESCOLA POLITÉCNICA**

**DEPARTAMENTO DE ENGENHARIA MECÂNICA**

**Simulação de um tráfego veicular através de um modelo de duas equações  
embasado na metodologia dos *Bond Graphs***

Trabalho de formatura apresentado à Escola  
Politécnica da Universidade de São Paulo para  
obtenção do título de Graduação em Engenharia

Tiago de Freitas Silveira

Orientador: Jorge Luis Baliño

Área de Concentração:  
Engenharia Mecânica

São Paulo

2009

## FICHA CATALOGRÁFICA

**Silveira, Tiago de Freitas**

**Simulação de um tráfico veicular através de um modelo de duas equações embasado na metodologia dos Bond Graphs / T.F. Silveira. – São Paulo, 2009.**

**83 p.**

**Trabalho de Formatura - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Mecânica.**

**1. Tráfego rodoviário 2. Método dos elementos finitos  
3. Engenharia mecânica I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia Mecânica II. t.**

## Resumo

O presente trabalho de formatura tem como objetivo apresentar o uso dos grafos de ligação na modelagem de um problema de tráfego veicular, através da analogia com um escoamento compressível. Estudou-se o tráfego em uma via de mão única, cujas variáveis independentes são a velocidade e a densidade. Adotou-se neste trabalho a metodologia dos grafos de ligação, pelas vantagens que esta metodologia apresenta na modelagem de sistemas físicos, conduzindo a solução sempre sob a óptica da conservação da energia. O modelo utilizado neste caso foi um modelo de duas equações de estado, desacoplando as variáveis densidade e velocidade, fazendo com que a gama de possibilidades a serem estudadas pelo modelo aumentasse consideravelmente. Utilizou-se fundamentos do método de elementos finitos, discretizando a via em um número finito de elementos e usadas funções de forma lineares para a velocidade e a densidade, e função linear para a função de peso. O trabalho apresenta alguns resultados da simulação de um tráfego veicular, como propagações de distribuições de carros finitas, e também inicia uma abordagem em problemas de *stop-and-go*.

## Abstract

This report presents the application of the bond graph methodology to model a traffic flow model, by means of an analogy with a compressible fluid flow. It was studied a traffic flow in a single way road, with car density and velocity as independent variables. The bond graph methodology was adopted because of the advantages it presents, related to the solution of the problem always from an energy point of view. In this work the two equation model was used, with different equations for density and velocity, providing many possibilities for simulations and analyses. The finite element method was used, with the description of the road in a finite number of elements. Linear shape functions for the density and velocity and linear function for the weight function were used. Some simulation results of a traffic flow are presented, related to propagation of finites distributions of vehicles. The approach to deal with stop-and-go problems is presented.

# Índice

1.	INTRODUÇÃO .....	4
2.	O FORMALISMO DOS BOND GRAPHS .....	6
2.1.	Definições iniciais .....	6
2.2.	Elementos básicos .....	7
2.3.	Elementos de junção .....	8
2.4.	Obtenção do grafo e causalidade.....	9
2.5.	Obtenção das equações de estado .....	10
2.6.	Generalização dos elementos básicos .....	11
3.	MODELO DE TRÁFEGO VEICULAR.....	17
3.1.	Modelo de duas equações.....	17
3.2.	Discretização das variáveis independentes .....	18
3.3.	Bond Graph do modelo .....	19
3.4.	Equações de estado .....	21
4.	DISCRETIZAÇÃO .....	26
4.1.	Funções interpolantes e de peso.....	26
4.2.	Linearização das equações de estado .....	28
4.3.	Método de integração numérica.....	29
5.	SIMULAÇÃO .....	33
5.1.	Parâmetros do problema.....	33
5.2.	Degrau de carros simples .....	33
5.3.	Rampa de carros.....	42
5.4.	Problemas Stop-and-go .....	49
5.5.	Problemas com a viscosidade numérica.....	57
6.	CONCLUSÕES .....	60
7.	BIBLIOGRAFIA .....	62
8.	ANEXOS .....	64

## Lista de Figuras

Figura 2.1 - Exemplo de Grafo de ligação. ....	6
Figura 2.2 - Forma matricial das equações de estado de um sistema genérico.....	11
Figura 2.3 - Exemplo de um Multibond.....	12
Figura 2.4 - Exemplo de Campo C. ....	13
Figura 2.5 - Exemplo de Campo I.....	14
Figura 2.6 - Exemplo de Campo IC. ....	15
Figura 2.7 - Exemplo de Campo R. ....	16
Figura 3.1 – Bond Graph representativo do modelo.....	19
Figura 3.2 – Termo de acoplamento entre a velocidade e a densidade.....	20
Figura 3.3 – Termo que relaciona a velocidade do veículo com seu momento. ....	21
Figura 4.1 - Funções de peso e forma em nós internos.....	26
Figura 4.2 - Funções de peso e forma no primeiro nó. ....	27
Figura 4.3 - Funções de peso e forma no último nó.....	27
Figura 5.1 - Condição inicial da densidade para degrau unitário.....	34
Figura 5.2 - Condição inicial da velocidade para degrau unitário. ....	34
Figura 5.3 – Iteração no tempo da densidade para $t=133s$ .....	35
Figura 5.4 - Iteração no tempo da densidade para $t=400s$ .....	36
Figura 5.5 - Iteração no tempo da densidade para $t=666s$ .....	36
Figura 5.6 - Iteração no tempo da densidade para $t=933s$ .....	37
Figura 5.7 - Iteração no tempo da densidade para $t=1199s$ .....	37
Figura 5.8 - Iteração no tempo da densidade para $t=1266s$ .....	38
Figura 5.9 - Iteração no tempo da densidade. ....	39
Figura 5.10 - Iteração no tempo da velocidade. ....	40
Figura 5.11 – Evolução temporal de um tráfego de acordo com o modelo LWR. ....	41
Figura 5.12 – Evolução temporal de um tráfego de acordo com o modelo anisotrópico.....	41
Figura 5.13 - Condição inicial da densidade para rampa. ....	42
Figura 5.14 - Condição inicial da velocidade para rampa.....	43
Figura 5.15 - Iteração no tempo da densidade para $t=133s$ .....	44
Figura 5.16 - Iteração no tempo da densidade para $t=400s$ .....	44
Figura 5.17 - Iteração no tempo da densidade para $t=666s$ .....	45
Figura 5.18 - Iteração no tempo da densidade para $t=933s$ .....	45
Figura 5.19 - Iteração no tempo da densidade para $t=1199s$ .....	46
Figura 5.20 - Iteração no tempo da densidade para $t=1266s$ .....	46
Figura 5.21 - Iteração no tempo da densidade. ....	47
Figura 5.22 - Iteração no tempo da velocidade. ....	48
Figura 5.23 – Condição inicial para a densidade de $0,15 \text{ veh/m}$ , para o problema de stop-and-go.....	49
Figura 5.24 - Condição inicial para a velocidade ( $\rho = 0,15 \text{ veh/m}$ ). ....	50
Figura 5.25 – Iteração no tempo da densidade para $t=133s$ .....	50
Figura 5.26 - Iteração no tempo da densidade para $t=400s$ .....	51
Figura 5.27 - Iteração no tempo da densidade para $t=666s$ .....	51
Figura 5.28 - Condição inicial para a densidade de $0,15 \text{ veh/m}$ , para o problema de stop-	

and-go.....	52
Figura 5.29 - Condição inicial para a velocidade ( $\rho = 0,15$ veh/m). .....	53
Figura 5.30 - Iteração no tempo da densidade para $t=133s$ .....	53
Figura 5.31 - Iteração no tempo da densidade para $t=400s$ .....	54
Figura 5.32 - Iteração no tempo da densidade para $t=666s$ .....	54
Figura 5.33 – Aproximação da área final para o problema de stop-and-go, para densidade de $0,15$ veh/m. ....	55
Figura 5.34 - Aproximação da área final para o problema de stop-and-go, para densidade de $0,20$ veh/m. ....	56
Figura 6.1 – Propagação da densidade, para o caso das condições de contorno travadas. .....	58
Figura 6.2 - Propagação da velocidade, para o caso das condições de contorno travadas. .....	59

# 1. INTRODUÇÃO

O formalismo dos *Bond Graphs* ou Grafos de Ligação possui varias vantagens no que se diz respeito a sistemas dinâmicos que transformam diferentes formas de energia, como é o caso dos fenômenos fluídicos, por possuir uma forte relação com a termodinâmica. Portanto o interesse em desenvolver um estudo aprofundado sobre essas duas áreas se justifica.

Até onde se sabe a primeira tentativa na aplicação da metodologia dos Bond Graphs (BG) na resolução de problemas de CFD, foi publicada por Fahrenthold [12]. Porém esta formulação não levou em conta os fenômenos de condução de calor, o que leva a problemas de *convecção-difusão*.

Especificamente sobre escoamentos compressíveis, Gandolfo [4], aplicou com sucesso a metodologia BG-CFD na resolução de problemas com tal comportamento, simulando o conhecido problema do “tubo de choque” (*shock tube*). Foi mostrado que essa metodologia pode modelar problemas não lineares complexos, levando em conta efeitos como compressibilidade, convecção, condução de calor e dissipação de energia.

Neste cenário é possível verificar que alguns estudos já foram realizados com a aplicação da metodologia de grafos de ligação no estudo de escoamentos compressíveis.

O presente trabalho irá focar-se no problema do tráfego veicular, que possui algumas simplificações em relação ao problema original de escoamento compressível, pois não contabiliza as trocas de calor e não há dissipação de energia.

Muitos trabalhos nesta área de pesquisa foram publicados, com resultados satisfatórios. Porém muitos deles utilizaram apenas uma equação de conservação de massa, o que acaba reduzindo as possibilidades de estudo de um tráfego veicular pela simplicidade do modelo.

Trabalhos utilizando equações desacopladas para a densidade e a velocidade também já foram estudados, porém utilizando o método dos elementos finitos diretamente sobre as equações de equilíbrio. A metodologia dos grafos de ligação introduz o estudo dos fluxos energéticos do sistema, e deste modo, são discretizados

termos de mesma dimensão (potência). Deste modo, a metodologia dos grafos de ligação faz com que o modelo encontrado seja mais conciso e também conserve todas as formas de energia além de respeitar a equação de estado.

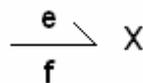
A organização do trabalho é a seguinte: no Capítulo 2 é feita uma apresentação do formalismo dos Bond Graphs; no Capítulo 3 é apresentado o modelo de tráfego veicular utilizando o formalismo dos *Bond Graphs*; no Capítulo 4 é apresentada a discretização utilizando o Método de Elementos Finitos; no Capítulo 5 são apresentadas algumas simulações numéricas e as dificuldades encontradas; no Capítulo 6 são apresentadas as conclusões do trabalho. Finalmente são apresentados as referências bibliográficas e um apêndice com o programa em MatLab utilizado.

## 2. O FORMALISMO DOS BOND GRAPHS

### 2.1. Definições iniciais

Um sistema dinâmico pode ser sempre representado por elementos básicos que interagem com outros elementos através de portas, e essa interação deve-se a passagem de energia por tais portas. Como essa afirmação é verdadeira para todos os tipos de sistemas sejam eles eletromagnéticos, mecânicos, fluídicos, térmicos, dentre outros, é vantajoso classificar esses diferentes tipos de energia em variáveis generalizadas. Tais variáveis são: esforço ( $e$ ), fluxo ( $f$ ), momento ( $p$ ) e deslocamento ( $q$ ). No caso, as duas variáveis principais são esforço e fluxo, pois o momento nada mais é que a integral no tempo do esforço, e o deslocamento a integral no tempo do fluxo. Com isso, conseguimos generalizar naturalmente modelos dinâmicos que comutam diferentes tipos de energia, já que o que interessa é a energia transferida, e pouco importa sua forma para o formalismo dos Grafos de Ligação.

Através dos Grafos de Ligação, a interconexão de diversos elementos multiportas é feita através de um gráfico no qual um número finito de objetos representativos ligados uns aos outros por laços que representam um fluxo de energia, como mostrado na Figura 3.1.



*Figura 2.1 - Exemplo de Grafo de ligação.*

Realizado este gráfico de objetos e laços, são atribuídas casualidades ao

sistema, ou seja, determinar os sinais de entrada e saída, e assim a obtenção das equações de estado que regem o sistema dinâmico é trivial.

## 2.2. Elementos básicos

A metodologia dos Grafos de Ligação possui elementos básicos generalizados, validos para todos os tipos de sistemas, de tal forma a descrever corretamente todas as transformações da energia total do sistema. São eles: resistores, capacitores, indutores, transformadores, e giradores.

Os resistores ( $R$ ) são elementos dissipadores de energia que possuem uma relação estática entre as variáveis de esforço e fluxo.

Capacitores ( $C$ ) são elementos que relacionam estaticamente as variáveis de esforço e deslocamento. Esse elemento armazena e libera energia para o sistema sem perdas, sendo basicamente elementos acumuladores de energia potencial.

Os indutores ( $I$ ) são elementos que relacionam estaticamente as variáveis de momento e fluxo, e estão relacionados com a indutância em circuitos elétricos, com a massa na analogia mecânica, dentre outras. Indutores podem ser vistos como elementos associados com a energia cinética.

Os elementos até agora citados, são considerados elementos de um porto. Os dois últimos elementos básicos citados, Transformadores ( $TF$ ) e Giradores ( $GY$ ) são elementos que também conservam energia, porém são elementos considerados duas portas.

Transformadores variam esforço e fluxo de modo que o produto entre eles, ou seja, a potencia transferida seja mantida inalterada. A relação constitutiva que rege um transformador está mostrada na Eq. 1.

$$e_1 = e_2 \cdot m$$

$$m \cdot f_1 = f_2$$

Eq. 1

Cujas variáveis  $e_1$  e  $e_2$  representam as variáveis de esforço nas respectivas portas 1 e 2, e  $f_1$  e  $f_2$  as variáveis de fluxo, e  $m$  é chamado de módulo do transformador.

Os giradores são elementos que também variam esforço e fluxo mantendo a potencia total inalterada, porém, eles possuem uma relação constitutiva bem diferente, como mostrado na Eq. 2.

$$e_1 = f_2 \cdot r$$

$$r \cdot f_1 = e_2$$

Eq. 2

No qual  $r$  é chamado de módulo do girador.

### 2.3. Elementos de junção

Os elementos de junção têm por finalidade conectar todos os outros elementos do sistema. Estes elementos são de extrema importância para o formalismo dos Grafos de Ligação, e a idéia por trás deles, é o que na analogia elétrica, conhecemos por ligação em série, e ligação em paralelo.

Existem dois tipos básicos de elementos de junção, o primeiro chamado de *Junção de Fluxo*, *0-junction* ou junção de esforço comum, como o próprio nome revela, é uma junção em que todas as ligações comutam o mesmo esforço. As equações constitutivas que regem tal junção são as Eq. 3.

$$e_1 = e_2 = e_3 = \dots = e_n$$

$$f_1 + f_2 + f_3 + \dots + f_n = 0$$

Eq. 3

O segundo tipo de junção é conhecido como Junção de Esforço, *1-junction* ou junção de fluxo comum. Todas as lições nessa junção comutam a mesmo fluxo, e as equações constitutivas que regem tal junção são as Eqs. 4.

$$e_1 + e_2 + e_3 + \dots + e_n = 0$$

$$f_1 = f_2 = f_3 = \dots = f_n$$

Eq. 4

É fácil observar que estes dois elementos de junção conservam a energia que flui por eles, não acumulando nada, simplesmente dividindo a energia em diferentes portas, de forma que a Eq. 5, que realiza o balanço total da energia no sistema, mostrada a seguir, está sendo respeitada.

$$e_1 \cdot f_1 + e_2 \cdot f_2 + e_3 \cdot f_3 + \dots + e_n \cdot f_n = 0$$

Eq. 5

## 2.4. Obtenção do grafo e causalidade

A obtenção dos grafos em um sistema dinâmico genérico se deve a um estudo geral de todos os elementos presentes em tal sistema, levando em conta os mecanismos de transferência da energia entre eles, de forma a se respeitar a conservação de energia total no sistema. É nesta etapa que devemos nos preocupar com a física envolvida no problema, para termos um

modelo realista e que gere os resultados esperados.

O problema em si está relacionado com a determinação de que tipos de junções estão envolvidos em cada particular conexão, de modo a se determinar se as uniões referidas comutam esforço ou comutam fluxo. Existem algumas técnicas específicas para cada tipo de problema, seja ele elétrico, mecânico, dentre outros, porém isto está aquém do objetivo final deste projeto e, portanto, iremos exemplificar tais usos de junções, no exemplo que virá em tópico a frente, por inspeção física.

As casualidades associadas a um gráfico qualquer que represente um sistema particular são atribuídas seguindo uma ordem pré-definida. Primeiramente devemos eleger as casualidades referentes às fontes (fonte de esforço ou fluxo), e a seguir repassar até onde possível as casualidades resultantes. Repete-se isto até que todas as fontes tenham suas casualidades eleitas. Após isso, caso ainda haja casualidades a serem eleitas, deve-se escolher algum elemento armazenador do tipo capacitor ou indutor, e associar a este a casualidade integral. Assim sendo, novamente repassam-se as casualidades resultantes com esta nova eleita e refaz-se este processo até não haver mais nenhuma casualidade não atribuída. Este processo também se fará mais claro no exemplo que virá em tópico a seguir.

## 2.5. Obtenção das equações de estado

As equações de estado regentes do sistema modelado estão diretamente relacionadas com as casualidades selecionadas. Após todas as casualidades estarem selecionadas, é possível saber quais são os elementos armazenadores que regem o sistema, ou seja, os elementos com casualidades integrais associadas. Assim sendo deve-se escrever, através das correlações entre os elementos básicos que compõe o gráfico, as equações de estado. Tais equações iram ter a forma mostrada na figura 3.2.

$$\begin{Bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dots \\ \dot{x}_n \end{Bmatrix} = \begin{bmatrix} f(x_1, x_2, \dots, x_n, U) \\ f(x_1, x_2, \dots, x_n, U) \\ \dots \\ f(x_1, x_2, \dots, x_n, U) \end{bmatrix}$$

Figura 2.2 - Forma matricial das equações de estado de um sistema genérico.

Onde  $x_1, x_2, \dots, x_n$  são as variáveis de estado do sistema, e  $U$  (fontes por exemplo) são as entradas que tal sistema pode vir a ter.

## 2.6. Generalização dos elementos básicos

### a) Multibonds (multi uniões) e transformadores e giradores multiportas

Multibonds são uma extensão da idéia dos já conhecidos gráficos representativos de um sistema dinâmico (*Bond Graphs*), que ao invés da idéia de portas unitárias onde potência é trocada com outros elementos, tem-se vetores de grafos de ligação que transmitem potência através de diversas portas. A utilização desta idéia pode ser compreendida em problemas de análise complexa, como no caso de modelos dinâmicos 3D, ou em problemas, até mesmo 1D, na área da termodinâmica, onde uma gama de efeitos deve ser contabilizada para se ter um modelo realista. A representação de um elemento Multibond pode ser verificada na figura 3.3.

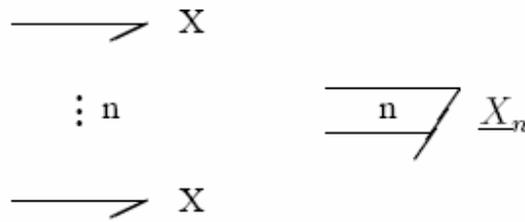


Figura 2.3 - Exemplo de um Multibond.

Está analogia pode ser implementada em todos os elementos básicos já discutidos. No caso de um Transformador Multibond, as características básicas já estudadas permanecem inalteradas, porém elas serão aplicadas a cada partição individual do Multibond. A antiga relação que existia antes em um transformador, chamada de módulo, relacionava os esforços e os fluxos, pode ser escrita em um Multibond através de uma relação matricial, como mostrado na Eq. 6.

$$\begin{bmatrix} e_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} 0 & T^T \\ T & 0 \end{bmatrix} \begin{bmatrix} f_1 \\ e_2 \end{bmatrix}$$

Eq. 6

Cuja matriz T representa o módulo do transformador multibond que possui a mesma dimensão do Multibond.

No caso do Girador Multibond, as relações são similares ao que acontecem com o Transformador Multibond, onde a equação matricial que rege tal elemento é mostrada na Eq. 7.

$$\begin{bmatrix} e_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} 0 & G^T \\ G & 0 \end{bmatrix} \begin{bmatrix} f_1 \\ e_2 \end{bmatrix}$$

Eq. 7

Da mesma forma que a matriz  $G$  representa o módulo do Girador Multibond, que possui a mesma dimensão do multibond.

### b) Campos C

Como citado nos elementos básicos, um Capacitor ou mesmo um Indutor eram elementos que armazenavam energia, e a liberavam sem perdas. Esta relação é válida também para os Campos C e os Campos I.

Um Campo C, ao contrário de um elemento C comum, possui vários grafos por onde se transfere potência. A figura 3.4 mostra um exemplo de tal elemento.

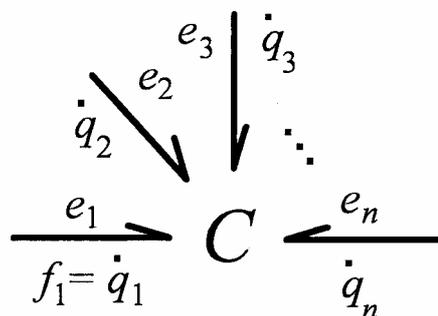


Figura 2.4 - Exemplo de Campo C.

No que diz respeito às casualidades atribuídas ao Campo C, elas não necessitam

necessariamente serem iguais, podendo haver no mesmo campo grafos com casualidade integral e grafos com casualidade derivativa. A escolha de cada uma respeita o mesmo processo já previamente discutido.

### c) Campos I

Da mesma forma que os Campos C, os Campos I também funcionam similarmente aos elementos básicos já citados I, armazenando e liberando energia sem perdas. Porém os Campos I possuem múltiplas portas ao contrário dos elementos I. A figura 3.5 mostra um exemplo de tal elemento.

As casualidades associadas a tais grafos nos Campos I, novamente não necessitam ser iguais, podendo ser integrais ou derivativas.

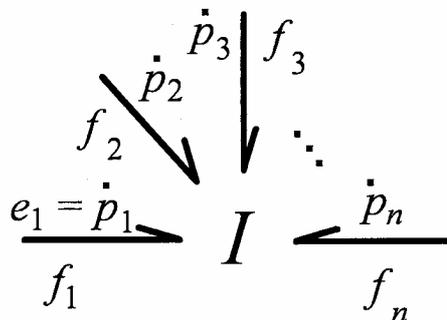


Figura 2.5 - Exemplo de Campo I.

### d) Campos IC

Como visto até agora, alguns elementos necessitam para serem descritos através da metodologia dos Grafos de Ligação não só elementos básicos como Capacitores ou Indutores, mas sim Campos C e Campos I. Porém em alguns casos Campos com comportamento puramente capacitivo ou puramente indutivo não iram ser suficientes para escrever o comportamento de

determinado elemento em um sistema dinâmico. Assim sendo se tem a necessidade do uso de um novo campo, que em algumas portas se comportará como um Capacitor, e em outras como um Indutor. Tal campo será chamado de Campo IC. Um exemplo genérico desse campo está mostrado na figura 3.6

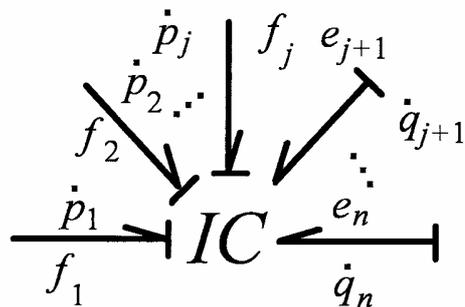


Figura 2.6 - Exemplo de Campo IC.

As casualidades nos grafos dos Campos IC não precisam necessariamente ser iguais, podendo assim ter em alguns grafos casualidade integral e em outros casualidade derivativa.

#### e) Campos R

Os Campos R têm funcionamento similar aos elementos básicos R, já citados previamente, dissipando energia, porém como ocorre nos Campos C e nos Campos I, estes possuem múltiplas portas por onde se transfere potência. Um exemplo desse campo pode ser visto na figura 3.7.

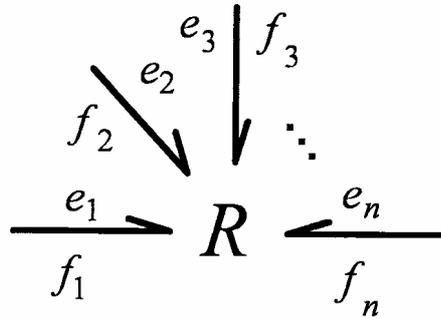


Figura 2.7 - Exemplo de Campo R.

f) Estruturas de junção

A Estrutura de Junção é um conjunto de restrições que acopla diferentes partes por todo o sistema dinâmico. Como a potência nunca é dissipada ou gerada numa estrutura de junção, o balanço da potência numa Estrutura de Junção resulta sempre em zero.

f) Estruturas de junção

A Estrutura de Junção é um conjunto de restrições que acopla diferentes partes por todo o sistema dinâmico. Como a potência nunca é dissipada ou gerada numa estrutura de junção, o balanço da potência numa Estrutura de Junção resulta sempre em zero.

Logo a Estrutura de Junção fornece relações entre esforços e fluxos, de acordo com as casualidades referidas no Grafo de Ligação.

### 3. MODELO DE TRÁFEGO VEICULAR

#### 3.1. Modelo de duas equações

Muitos modelos de uma equação foram desenvolvidos com o intuito de se simular um tráfego veicular. Como exemplo, podemos citar o modelo de Lighthill & Whitham and Richards [14], conhecido como LWR que utiliza uma equação diferencial para quantificar os fluxos massa e uma equação de equilíbrio para a velocidade, em função da massa. Porém este modelo apresenta algumas limitações, como por exemplo, a impossibilidade de se simular um tráfego com paradas de semáforo (*stop-and-go*).

Um modelo bem sucedido neste sentido, que será utilizado neste trabalho foi o de Jiang *et al.* [15], que desenvolveu um modelo de carro seguidor, como mostrado a seguir:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x} (\rho v) = 0$$

Eq. 8

$$\frac{\partial v}{\partial t} + v \frac{\partial v}{\partial x} = \frac{v_e - v}{\tau} + G$$

Eq. 9

No qual o termo genérico foi escrito como:

$$G = c \frac{\partial v}{\partial x}$$

Eq. 10

### 3.2. Discretização das variáveis independentes

As variáveis independentes, densidade e velocidade, foram discretizadas de modo a se ter um modelo num nível discreto, assim como no Método dos Elementos Finitos [13], por um número finito de elementos (nós) e funções interpolantes, como mostrado a seguir.

$$\rho(x,t) = \sum_{k=1}^{n_\rho} \rho_k(t) \varphi_{\rho k}(x) = \underline{\rho}^T \cdot \underline{\varphi}_\rho$$

Eq. 11

$$v(x,t) = \sum_{k=1}^{n_v} v_k(t) \varphi_{vk}(x) = \underline{v}^T \cdot \underline{\varphi}_v$$

Eq. 12

### 3.3. Bond Graph do modelo

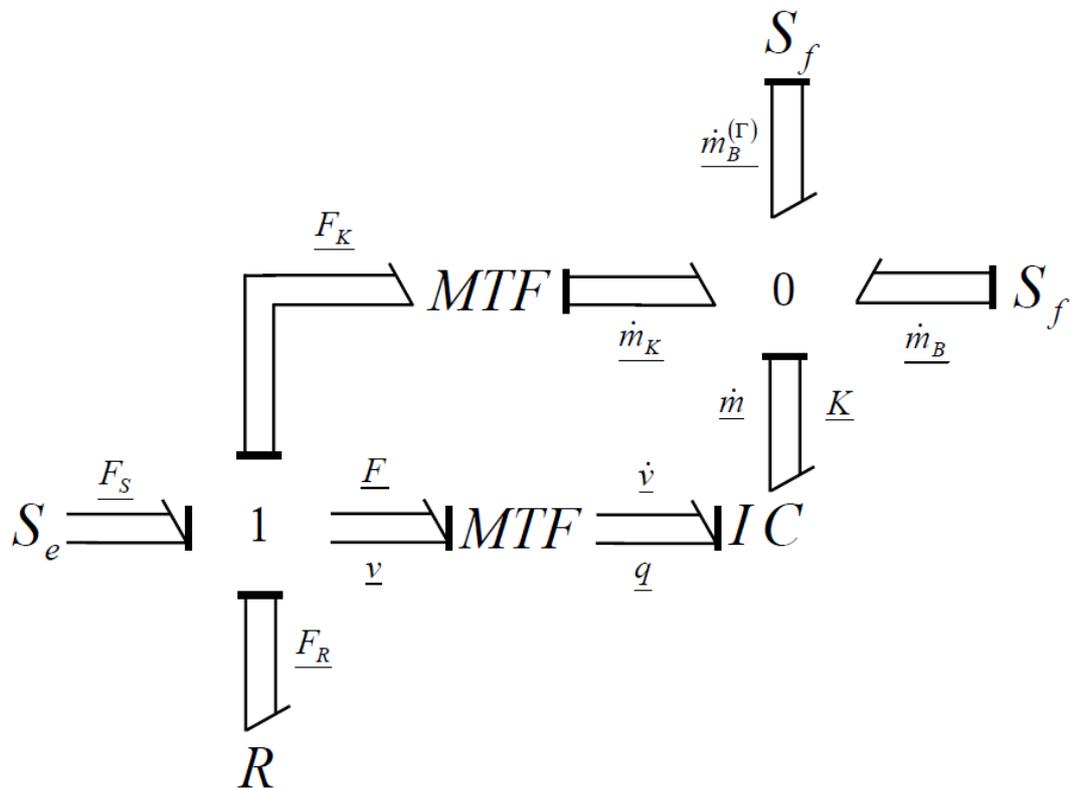


Figura 3.1 – Bond Graph representativo do modelo.

O bond graph e as equações subsequentes foram obtidos previamente em um trabalho em colaboração realizado pelo orientador Jorge Baliño com a *Ecole Centrale de Lille* (França).

Este modelo foi obtido a partir de duas equações diferenciais: uma de continuidade da massa (conservação da massa), e outra para a velocidade, como mostrado nas Eqs. 8 e 9. Deste modo, têm-se duas variáveis independentes, uma é a

densidade e a outra a velocidade, e por consequência duas condição de contorno.

É válido notar que no grafo de ligação proposto, aparecem alguns elementos especiais. O primeiro deles é o transformador modulado, representado por MTF, que nada mais é que um elemento transformador, com razão de modulação variável. O segundo seria o campo indutivo capacitivo, representados por IC, que é um elemento multiportas, em que uma das portas funciona como um elemento indutivo, e a outra um elemento capacitivo. Os elementos  $S_e$  e  $S_f$  tratam-se respectivamente de fontes de esforço e fluxo.

O primeiro transformador modulado que aparece no grafo de ligação da figura 4.2, representa um termo de acoplamento entre a porta de velocidade e a porta de densidade. A figura abaixo apresenta este termo separado.

$$\begin{array}{ccc} \underline{F_K} & \underline{K}^{-1} \cdot \underline{M_K} & \underline{K} \\ \hline \underline{v} & MTF & \underline{\dot{m}_K} \end{array}$$

Figura 3.2 – Termo de acoplamento entre a velocidade e a densidade.

O segundo transformador relaciona a velocidade do veículo com o momento veicular. Podemos observar este termo separado abaixo.

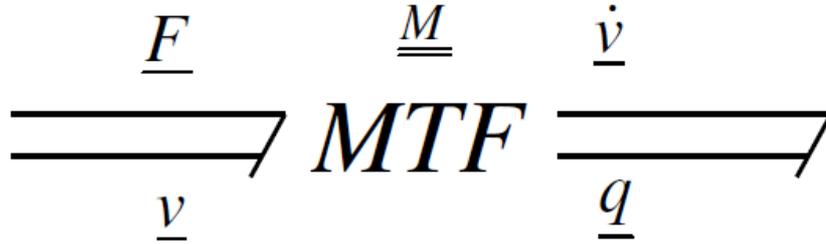


Figura 3.3 – Termo que relaciona a velocidade do veículo com seu momento.

### 3.4. Equações de estado

Como realizado no método de Petrov-Galerkin [13], os termos da equação de conservação de massa são multiplicados pela função de peso  $w_{\rho k}$  e integrada no domínio, obtendo:

$$\underline{\underline{K}} \cdot \underline{\dot{m}} = \underline{W}_B^{(\Gamma)} + \underline{W}_B + \underline{W}_K$$

Eq. 13

No qual, os termos mostrados na equação acima, podem ser escritos como:

$$\underline{W}_B^{(\Gamma)} = -\left[ \left( \underline{w}_{\rho} \kappa \rho v \right)_L - \left( \underline{w}_{\rho} \kappa \rho v \right)_0 \right]$$

Eq. 14

$$\underline{W}_B = \underline{\underline{W}}_B \cdot \underline{\rho}$$

Eq. 15

$$\underline{W}_B = \int_{\Omega} \frac{\partial \underline{w}_\rho}{\partial x} \kappa \rho v dx$$

Eq. 16

$$\underline{W}_K = \underline{\underline{W}}_K \cdot \underline{\rho}$$

Eq. 17

$$\underline{W}_k = \int_{\Omega} \underline{w}_\rho \rho v^2 \frac{\partial v}{\partial x} dx$$

Eq. 18

Podemos também escrever a equação de balanço de massa como:

$$\underline{\dot{m}} = \underline{\dot{m}}_B^{(\Gamma)} + \underline{\dot{m}}_B + \underline{\dot{m}}_K$$

Eq. 19

Como realizado no método de Petrov-Galerkin [13], os termos da equação de conservação do momento são multiplicados pela função de teste  $\varphi_{vk}$  e integrada no domínio, obtendo:

$$\underline{\underline{M}} \cdot \underline{\dot{v}} = \underline{F}_e + \underline{F}_G - \underline{F}_R - \underline{F}_K$$

Eq. 20

No qual, os termos mostrados na equação acima, podem ser escritos como:

$$\underline{F}_R = \underline{\underline{F}}_R \cdot \underline{v}$$

Eq. 21

$$\underline{F}_R = \int_{\Omega} \rho \frac{v}{\tau} \underline{\varphi}_{-v} dx$$

Eq. 22

$$\underline{F}_S = \underline{F}'_S + \underline{\underline{F}}_S \cdot \underline{v}$$

Eq. 23

$$\underline{F}_S = \int_{\Omega} \rho \left( \frac{v_e}{\tau} + G \right) \underline{\varphi}_{-v} dx$$

Eq. 24

$$\underline{F}_K = \underline{\underline{F}}_K \cdot \underline{v}$$

Eq. 25

$$\underline{F}_K = \int_{\Omega} \rho v \frac{\partial v}{\partial x} \underline{\varphi}_{-v} dx$$

Eq. 26

Nas equações acima,  $\kappa$  representa o potencial energia cinética por unidade de veículos, dado pela Eq. 27,  $\tau$  é o tempo reativo,  $v_e(\rho)$  é a velocidade de equilíbrio dado pela Eq. 28, dependente de  $\rho$ , e  $G$  é um termo de fonte.

$$\kappa = \frac{1}{2} v^2$$

Eq. 27

$$v_e = v_{\max} \left( 1 - \frac{\rho}{\rho_{\max}} \right)$$

Eq. 28

Pode-se mostrar também a relação entre a massa e a densidade como:

$$\underline{m} = \underline{\underline{\Omega}}_{\rho} \cdot \underline{\rho}$$

Eq. 29

No qual a matriz quadrada  $\underline{\underline{\Omega}}_{\rho}$  é a matriz de volume dada por:

$$(\underline{\underline{\Omega}})_{kn} = \underline{\underline{\Omega}}_{\rho k} \delta_{kn}$$

Eq. 30

No qual,

$$\underline{\underline{\Omega}}_{\rho k} = \int_{\Omega} \underline{\underline{\varphi}}_{\rho k} dx$$

Eq. 31

As matrizes  $K$  e  $M$  são respectivamente as matrizes de energia cinética e de inércia, e elas podem ser escritas como:

$$(K)_{kj} = \frac{1}{\Omega_{\rho j}} \int_{\Omega} \kappa w_{\rho k} \varphi_{\rho j} dx$$

Eq. 32

$$(M)_{mn} = \int_{\Omega} \rho \varphi_{\underline{vm}} \varphi_{\underline{vn}} dx$$

Eq. 33

## 4. DISCRETIZAÇÃO

### 4.1. Funções interpolantes e de peso

As funções interpolantes utilizadas foram funções lineares, e também foram introduzidas funções de peso  $w_{\rho k}(x,t)$  de modo a satisfazer as trocas de potência no sistema através das condições de contorno, como também ponderar a importância de diferentes termos de potência em nós vizinhos.

A figura 4.4 mostra as funções de peso e forma utilizadas em nós internos, a figura 6.5 mostra as funções de peso e forma utilizadas no primeiro nó e a figura 4.6 mostra as funções de peso e forma utilizadas no último nó.

A princípio foram utilizadas funções de forma constantes, mas como o problema estudado apresenta gradientes grandes de densidade e velocidade, para se conseguir chegar à convergência do programa, as funções lineares foram escolhidas.

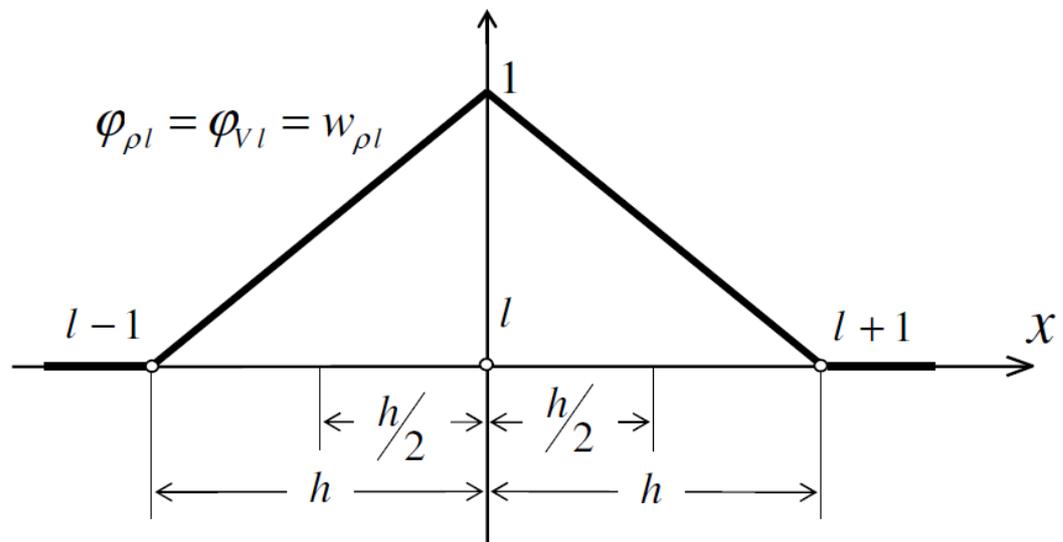


Figura 4.1 - Funções de peso e forma em nós internos.

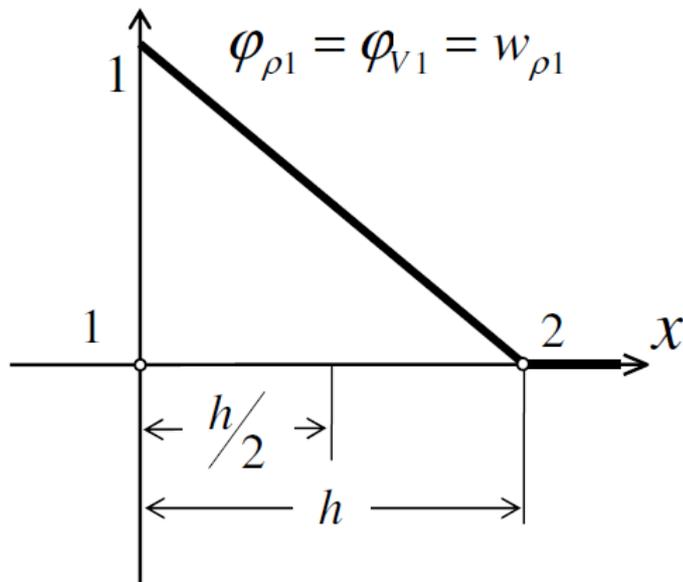


Figura 4.2 - Funções de peso e forma no primeiro nó.

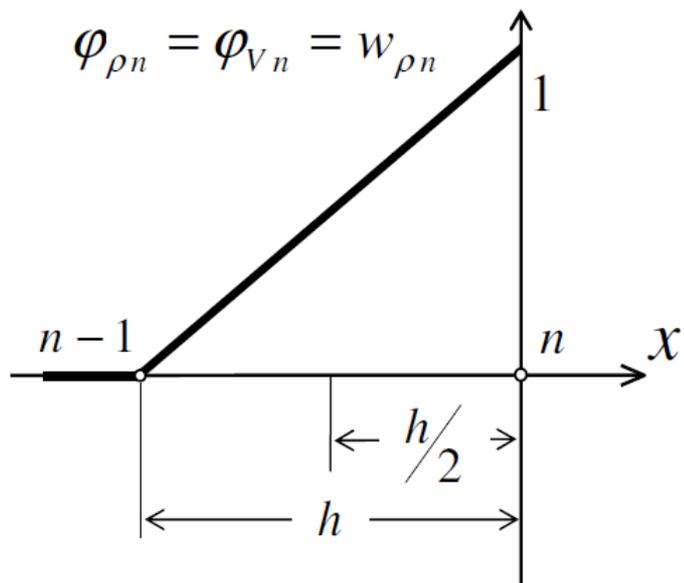


Figura 4.3 - Funções de peso e forma no último nó.

## 4.2. Linearização das equações de estado

O processo de integração numérico descrito acima é realizado para as duas equações de estado obtidas, a da conservação da massa (densidade) e a do momento (velocidade).

Para isso, as equações de estado Eq. 8 e Eq. 9 foram rearranjadas de modo a se ter um sistema linear, no qual as variáveis independentes são calculadas no tempo posterior, e termos não-lineares são linearizados supondo estes dependentes das variáveis no tempo atual.

As equações lineares obtidas através desse processo estão mostradas nas equações Eq. 21 e Eq. 22.

$$\left[ \frac{\underline{\underline{K}} \cdot \underline{\underline{\Omega}}_{\rho}}{dt} - \underline{\underline{W}}_B - \underline{\underline{W}}_K \right] \cdot \underline{\underline{\rho}}^{t+\Delta t} = \frac{\underline{\underline{K}} \cdot \underline{\underline{\Omega}}_{\rho}}{dt} \underline{\underline{\rho}}^t$$

Eq. 34

$$\left[ \frac{\underline{\underline{M}}}{dt} - \underline{\underline{F}}_G + \underline{\underline{F}}_R + \underline{\underline{F}}_K \right] \cdot \underline{\underline{v}}^{t+\Delta t} = \frac{\underline{\underline{M}}}{dt} \underline{\underline{v}}^t + \underline{\underline{F}}_e$$

Eq. 35

Na equação 49, o termo  $\underline{\underline{W}}_B^{(\Gamma)}$  irá aparecer do lado esquerdo ou do lado direito dependendo da condição de contorno imposta. Caso seja imposta a densidade como condição de contorno, este termo irá aparecer no lado direito, pois a densidade será conhecida. Caso a velocidade seja imposta, o termo aparecerá no lado esquerdo, pois a densidade deve ser calculada.

### 4.3. Método de integração numérica

O método utilizado para integrar numericamente as equações diferenciais obtidas, e por consequência iterar o sistema no tempo, foi o método preditor-corretor. O método preditor-corretor é um método de integração numérica em que são efetuadas algumas iterações para obter o valor de  $f(t+\Delta t)$  com alguma precisão desejada, resultando finalmente um esquema implícito.

Este método consiste em obter uma aproximação preditora, que como comentado, utiliza o fundamento do método de Euler. Consecutivamente calcula-se o termo preditor que é pensado de acordo com o teorema do valor médio, e usa um fator de relaxamento,  $\zeta$ , sempre entre zero e um.

Assumindo a equação diferencial a seguir:

$$\frac{df}{dt} = g(f)$$

*Eq. 36*

O termo geral, do termo corretor pode ser escrito como mostrado a seguir:

$$f_p^l = f(t) + g[f_c^{l-1}] \cdot \Delta t$$

*Eq. 37*

E o termo corrigido pode ser escrito como:

$$f_c^l = f_c^{l-1} + \zeta \cdot [f_p^l - f_c^{l-1}]$$

*Eq. 38*

Os valores iniciais para os termos preditor e corretor, podem ser escritos como:

$$f_p^0 = f(t) + g[f(t)] \cdot \Delta t$$

Eq. 39

$$f_c^0 = f(t) + \zeta \cdot [f_p^0 - f(t)]$$

Eq. 40

O processo acima descrito irá ocorrer até que o critério de convergência seja alcançado. O critério de convergência utilizado pode ser observado a seguir:

$$\left| \frac{f_p^{l+1} - f_c^l}{f_p^{l+1}} \right| < \varepsilon$$

Eq. 41

Caso  $f_p^{l+1}$  seja zero, utiliza-se o seguinte critério:

$$\left| f_p^{l+1} - f_c^l \right| < \varepsilon$$

Eq. 42

O método acima descrito funciona muito bem para diversos tipos de problemas, porém como o problema de tráfego veicular analisado apresenta gradientes de densidade, assim como de velocidade muito altos, ele acaba não alcançando a convergência.

Com o intuito de alcançar tal convergência, foi introduzida no programa uma viscosidade artificial numérica, como realizado por Beskos [16]. Tal processo, chamado de suavização (*smoothing*) minimiza as variações numéricas ocorridas pelos altos gradientes e auxilia na convergência do método preditor-corretor.

Temos que:

$$\mu = \frac{1}{1 + e^{-\gamma}}$$

Eq. 43

$$\gamma = \frac{(x_{i+1} - x_{i-1}))}{x_{\max}}$$

Eq. 44

No qual  $i$  varia de 2, até o número de nós utilizados para o sistema menos 1 ( $n-1$ ).  
Deste modo, com as Eqs. 43 e 44 podem-se calcular as variáveis suavizadas como:

$$\rho_{i,s}^l = \rho_i^l + \mu_{\rho}^l \cdot \frac{1}{2} (\rho_{i-1}^l - 2\rho_i^l + \rho_{i+1}^l)$$

Eq. 45

$$v_{i,s}^l = v_i^l + \mu_v^l \cdot \frac{1}{2} (v_{i-1}^l - 2v_i^l + v_{i+1}^l)$$

Eq. 46

Desta forma, o algoritmo utilizado na Eq. 36, para os nós internos fica:

$$f_c^l = f_{p,s}^{l-1} + \zeta \cdot [f_{p,s}^l - f_{p,s}^{l-1}]$$

Eq. 47

E para os nós externos fica:

$$f_c^l = f_c^{l-1} + \zeta \cdot [f_p^l - f_c^{l-1}]$$

Eq. 48

E também temos que as convergências serão medidas para os nós internos como mostrado abaixo:

$$\left| \frac{f_{p,s}^{l+1} - f_{p,s}^l}{f_{p,s}^{l+1}} \right| < \mathcal{E}$$

Eq. 49

Caso  $f_{p,s}^{l+1}$  seja menor que  $\mathcal{E}$ , utiliza-se o seguinte critério:

$$\left| f_{p,s}^{l+1} - f_{p,s}^l \right| < \mathcal{E}$$

Eq. 50

E para os nós externos, a convergência será medida da mesma forma como mostrado nas equações 39 e 40.

Vale notar que os nós internos e os nós externos foram tratados de maneiras diferentes pelo fato de que o método de viscosidade numérica só é aplicado para os nós internos. Assim sendo a integração numérica e a análise da convergência para os nós externos deve ser feita diferentemente do que é feito nos nós internos, utilizando o método preditor-corretor padrão.

## 5. SIMULAÇÃO

### 5.1. Parâmetros do problema

Com o intuito de se simular numericamente o modelo proposto, inicialmente foi desenvolvido um programa em linguagem C, que tinha como objetivo calcular as variáveis do problema, e interar no tempo um modelo de um tráfego unidimensional. Diante das complexas atividades encontradas e pela carga computacional, foi optada posteriormente a programação numérica do problema e Matlab 2008b.

Os parâmetros utilizados para este programa foram:

Variável	Valor
Número de nós (n)	101
$\rho_{\max}$	0,2 veh/m
$v_{\max}$	30 m/s
Comprimento do trecho (L)	20000 m
$\tau$	10 s
c	11 m/s
dt	1,333 s

*Tabela 1 – Parâmetros utilizados no programa.*

### 5.2. Degrau de carros simples

A primeira simulação foi realizada utilizando-se com condição inicial um degrau de carros simples (um único degrau na via) com densidade de 0,15 carros por metro e para a velocidade foi imposta a velocidade de equilíbrio para as respectivas densidades, como mostrado a seguir:

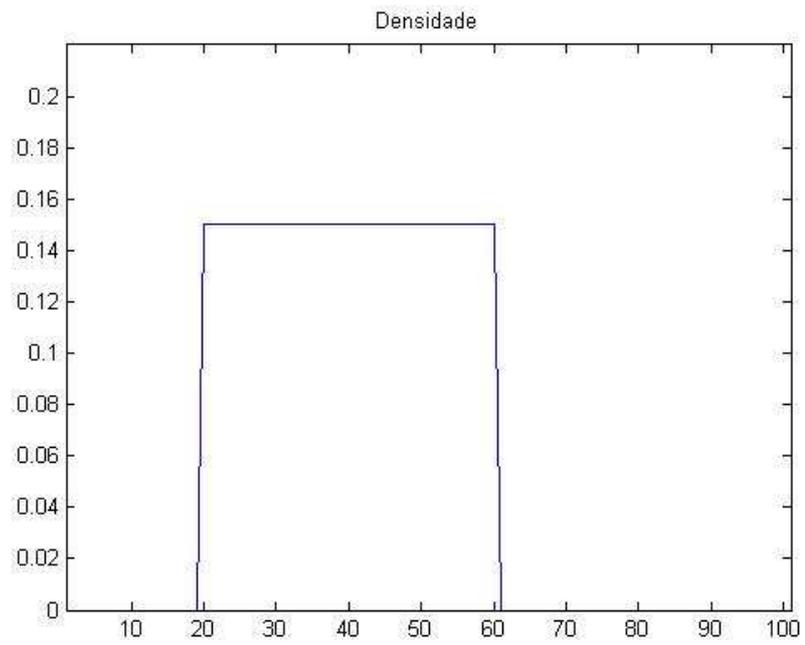


Figura 5.1 - Condição inicial da densidade para degrau unitário.

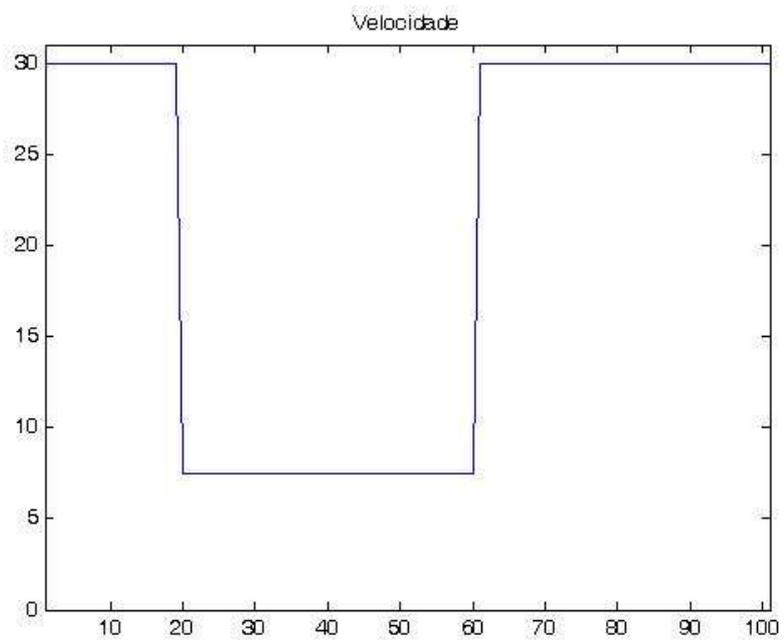
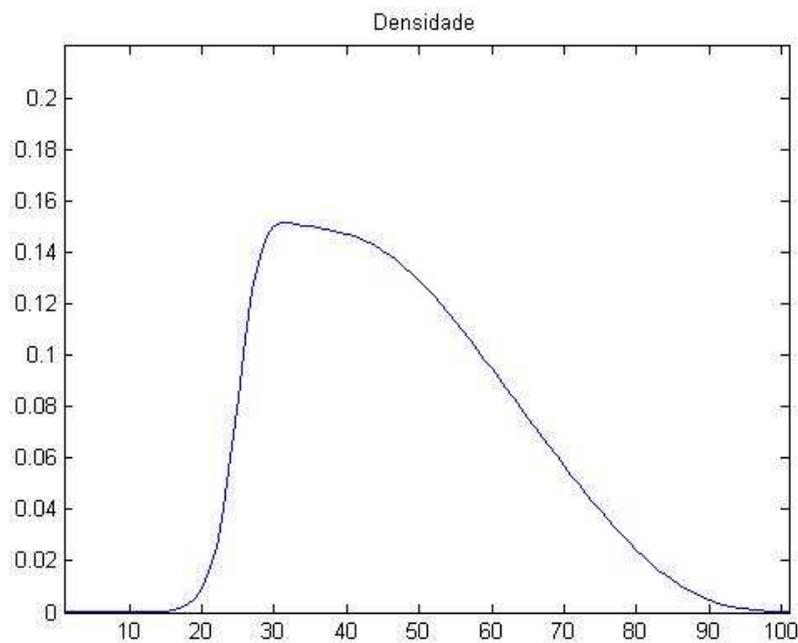


Figura 5.2 - Condição inicial da velocidade para degrau unitário.

Como condições de contorno foram impostas que tanto para o primeiro nó como para o último nó, as velocidades nestes dois pontos deveriam ser a velocidade de equilíbrio para a respectiva densidade, ou seja, não foi colocada nenhuma restrição a passagem de carros nestes dois pontos, resultando que os carros podiam sair livremente, e os carros que estariam entrando (nenhum) continuariam entrando. Pela óptica de um escoamento compressível, trata-se de um simples problema de propagação de onda.

Este problema inicial foi simulado por 1266 segundos. Esta simulação mostrou claramente à formação de uma frente de onda a jusante do degrau de carros, e o retardamento da propagação dos carros a montante do degrau de carros. As interações no tempo, tanto para a densidade como para a velocidade, podem ser observadas nas figuras abaixo:



*Figura 5.3 – Iteração no tempo da densidade para  $t=133s$ .*

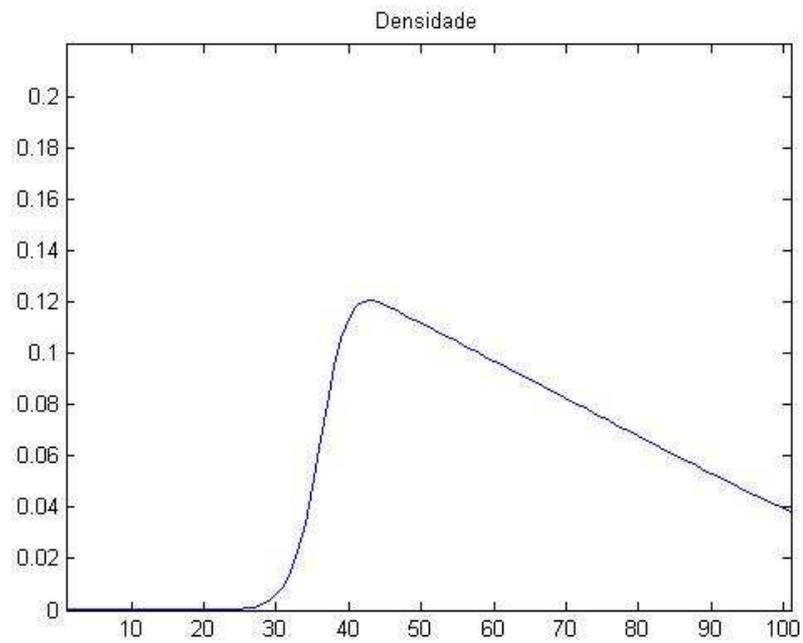


Figura 5.4 - Iteração no tempo da densidade para  $t=400s$ .

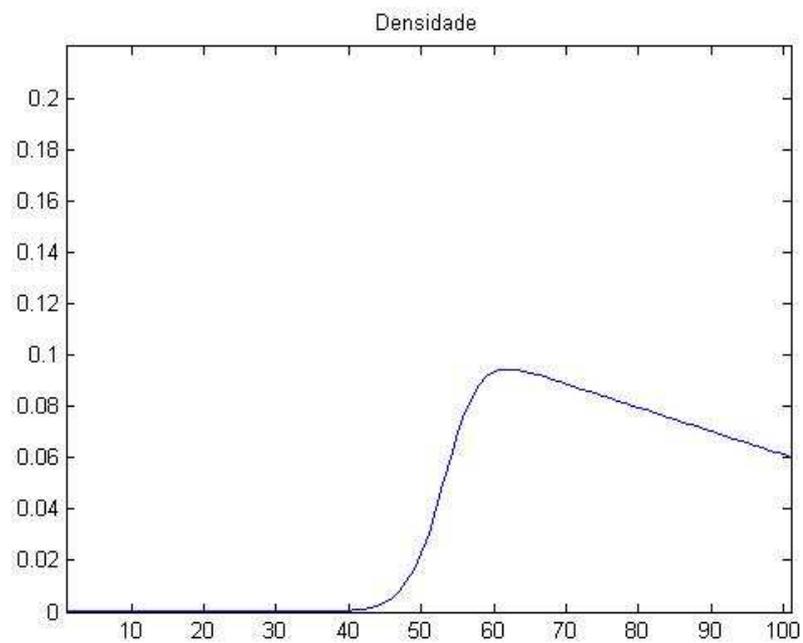
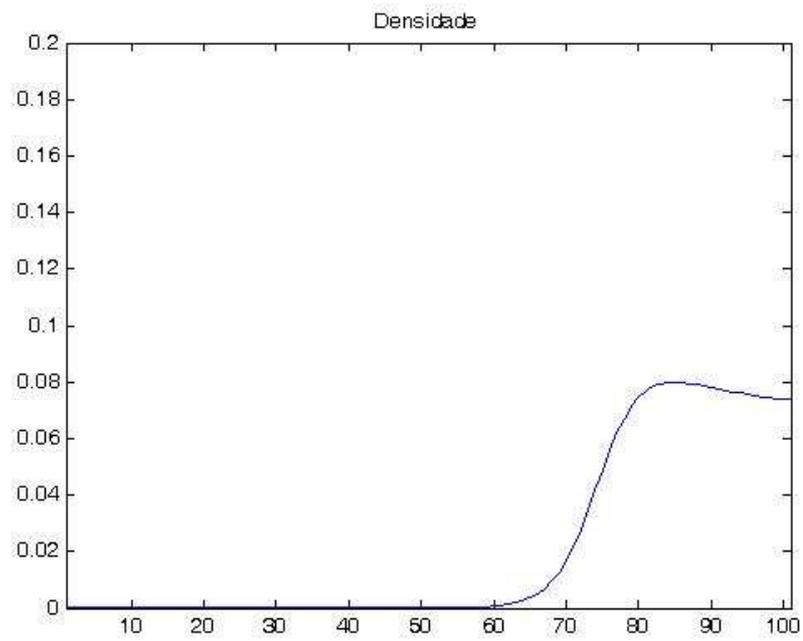
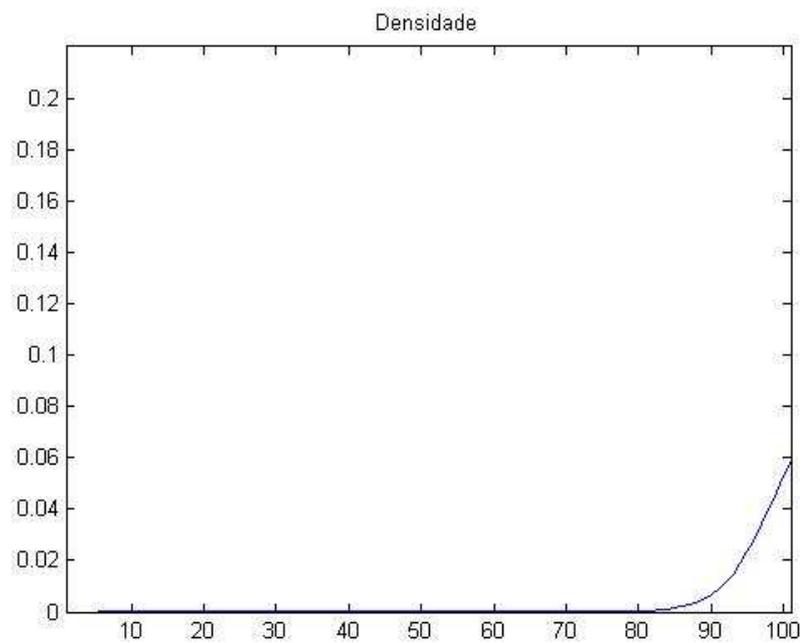


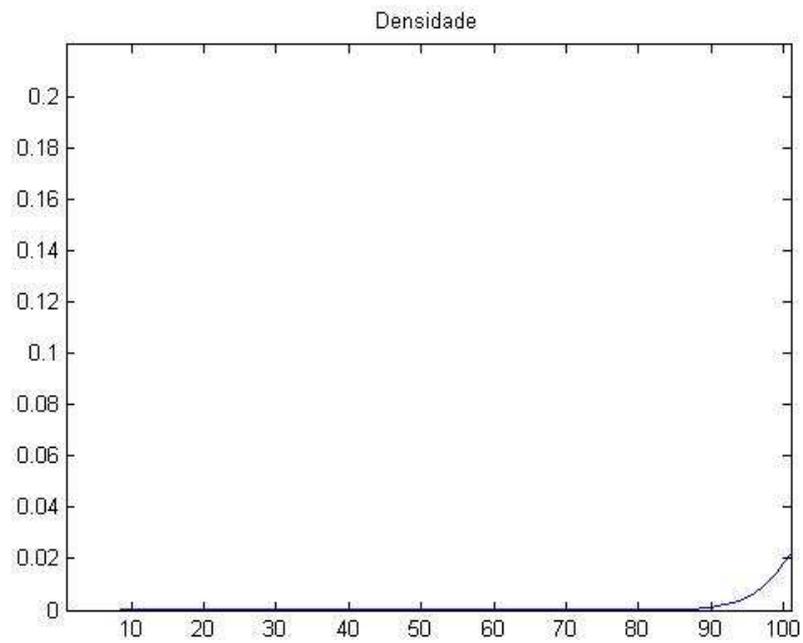
Figura 5.5 - Iteração no tempo da densidade para  $t=666s$ .



*Figura 5.6 - Iteração no tempo da densidade para  $t=933s$ .*

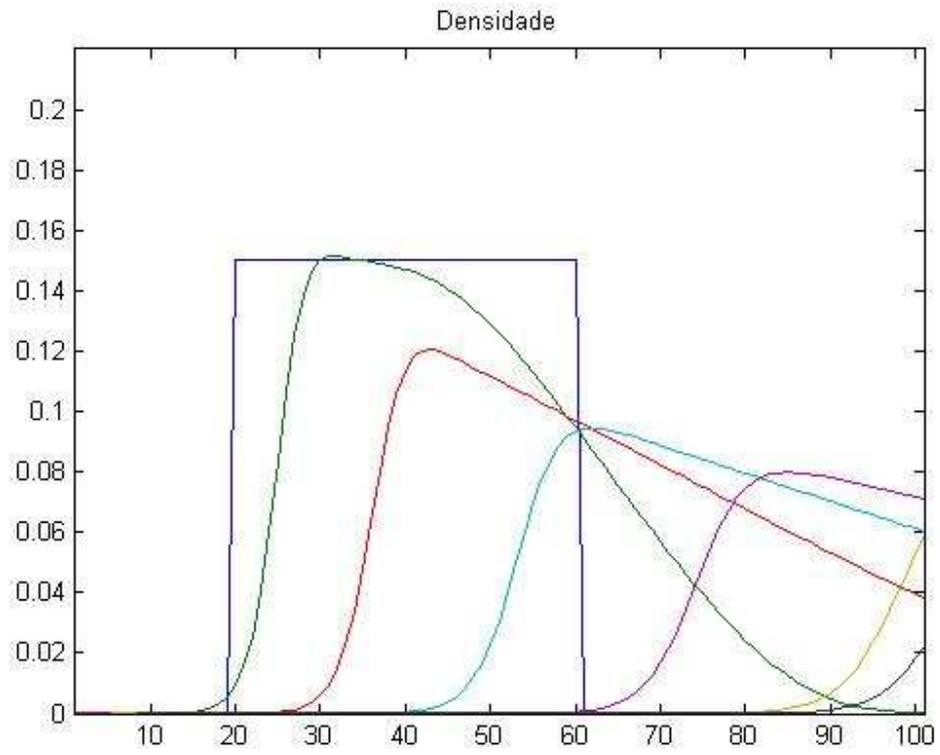


*Figura 5.7 - Iteração no tempo da densidade para  $t=1199s$ .*



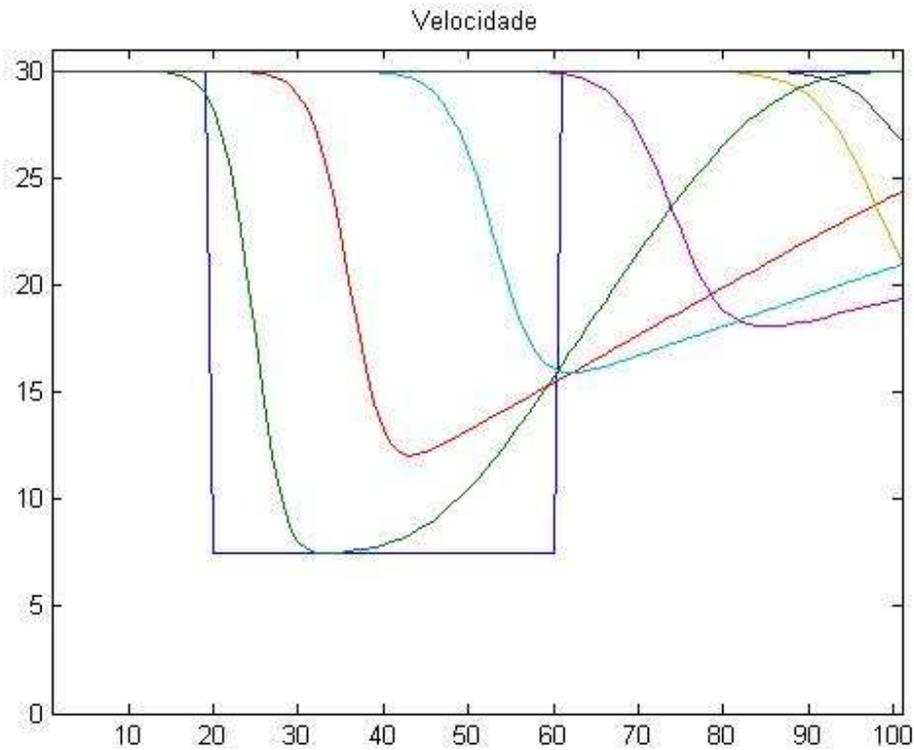
*Figura 5.8 - Iteração no tempo da densidade para  $t=1266s$ .*

Para se ter uma visão mais geral da propagação dos carros, pode-se observar na figura abaixo todas as iterações mostradas acima em um único gráfico:



*Figura 5.9 - Iteração no tempo da densidade.*

Para a velocidade o comportamento é similar, de modo que a iteração no tempo da velocidade, para os mesmos tempos sinalizados na densidade se dá como mostrado abaixo:



*Figura 5.10 - Iteração no tempo da velocidade.*

É possível observar que o tempo necessário para ocorrer a dissipação total dos carros na via analisada é de aproximadamente 1266 segundos, o que nos fornece uma velocidade média da onda de aproximadamente:

$$\bar{V} = \frac{16000m}{1266s} = 12,63m / s = 45,49km / h$$

*Eq. 51*

Pode-se comparar este resultado obtido com alguns modelos propostos na literatura. Como exemplo, temos o modelo LWR, mostrado abaixo:

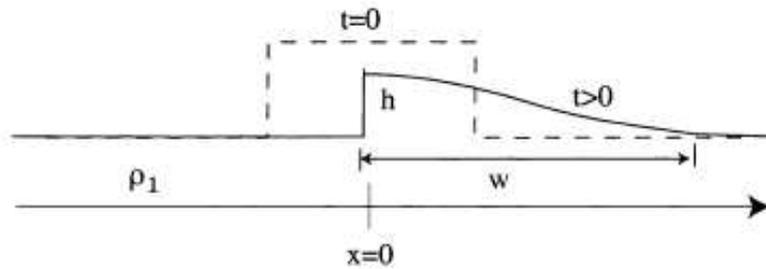


Figura 5.11 – Evolução temporal de um tráfego de acordo com o modelo LWR.

Este modelo apresenta a evolução temporal da densidade de uma forma satisfatória, porém, sua principal limitação é trabalhar com a densidade e a velocidade acopladas. Fazendo isso, não é possível trabalhar em situações de desequilíbrio e também ficam limitadas as opções de estudo pelo fato de se ter apenas uma condição de equilíbrio.

Outro modelo também estudado na literatura é o modelo anisotrópico, como mostrado a seguir:

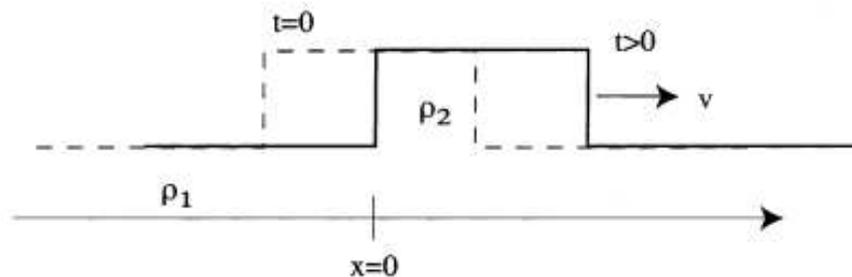
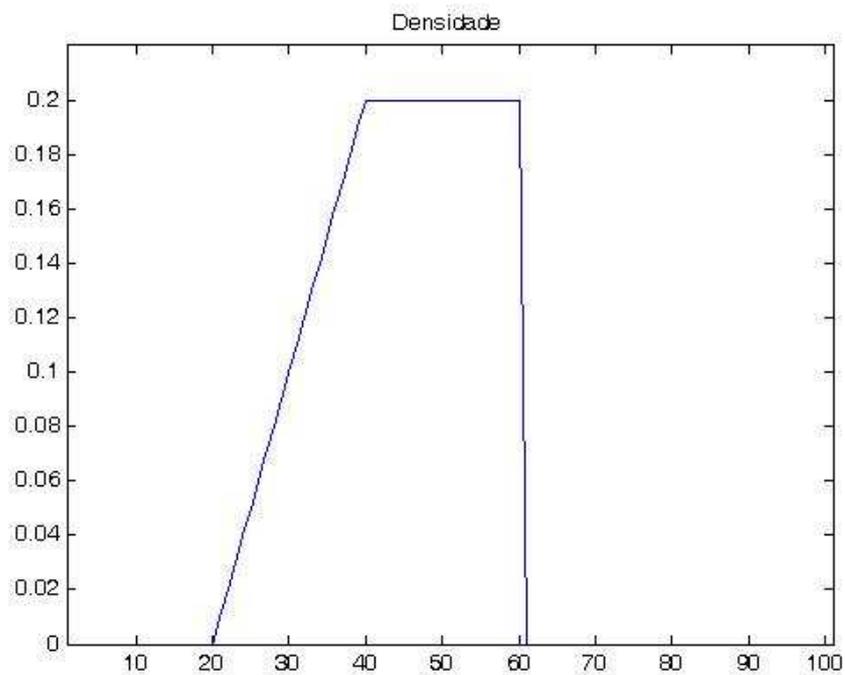


Figura 5.12 – Evolução temporal de um tráfego de acordo com o modelo anisotrópico.

Este modelo proposto possui um grande problema, pois ele considera que carros com diferentes velocidades a frente, tendem a se movimentar por igual. Obviamente isso não é verdade, e uma mostra empírica disso seria a abertura de um farol: os carros da frente começam a se movimentar, porém só segundos depois os de trás começam a ganhar velocidade.

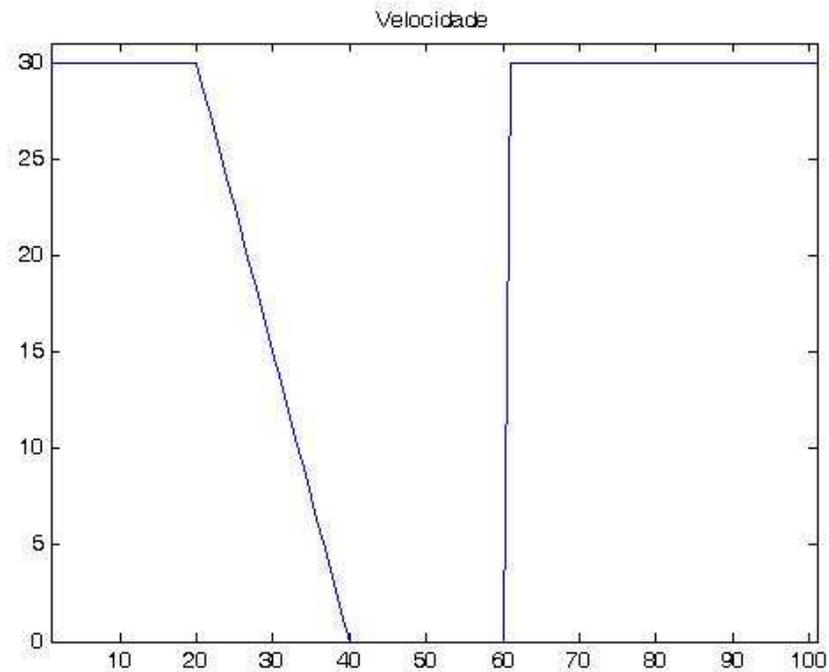
### 5.3. Rampa de carros

A segunda simulação foi realizada utilizando-se com condição inicial uma rampa de carros simples, iniciando no nó 20 com densidade 0 e crescendo até o nó 40 atingindo uma densidade de 0,20 carros por metro (máxima densidade possível) e para a velocidade foi imposta a velocidade de equilíbrio para as respectivas densidades, como mostrado a seguir:



*Figura 5.13 - Condição inicial da densidade para rampa.*

E para a velocidade:

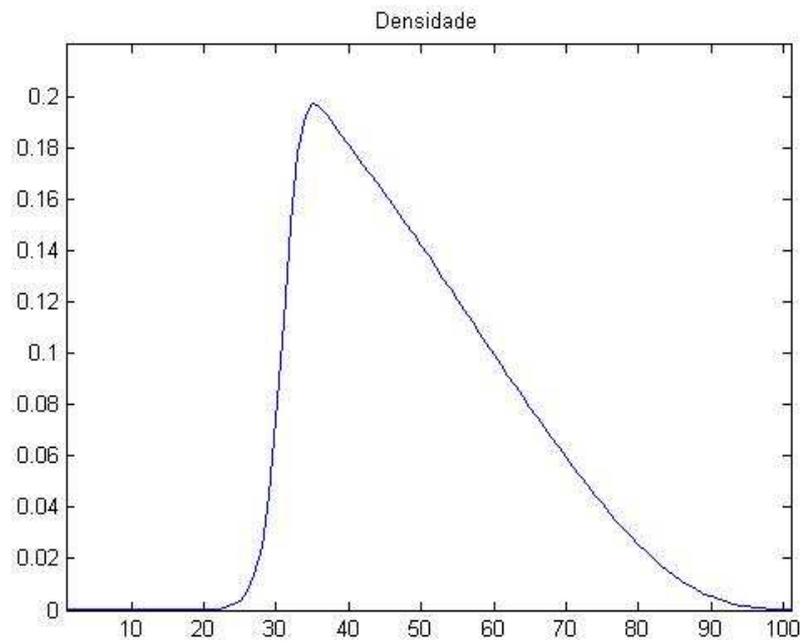


*Figura 5.14 - Condição inicial da velocidade para rampa.*

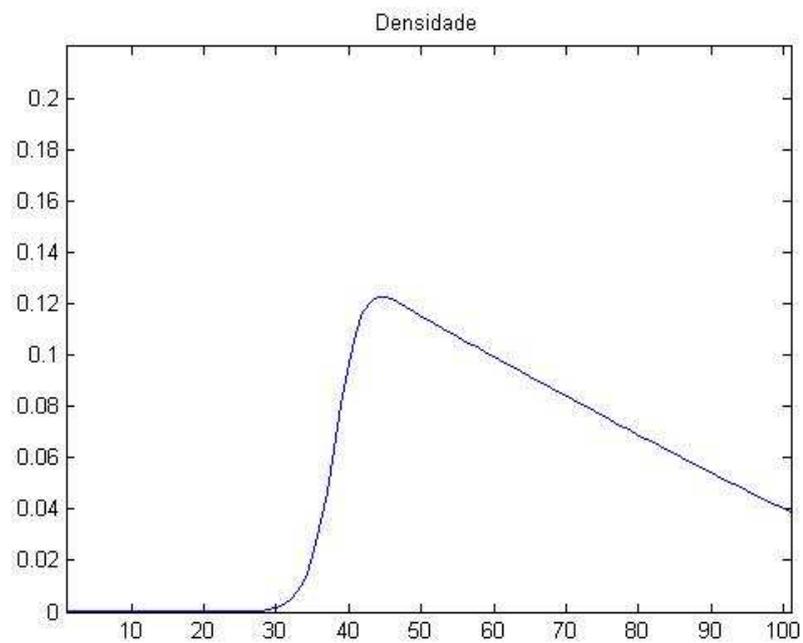
Como condições de contorno foram impostas que tanto para o primeiro nó como para o último nó, as velocidades nestes dois pontos deveriam ser a velocidade de equilíbrio para a respectiva densidade, ou seja, não foi colocada nenhuma restrição a passagem de carros nestes dois pontos, resultando que os carros podiam sair livremente, e os carros que entravam (nenhum) continuariam entrando.

Novamente este problema foi simulado por 1266 segundos. Com esta condição inicial foi possível verificar dois fenômenos distintos acontecendo simultaneamente, o primeiro, também observado na primeira simulação, é a propagação de uma frente de onda, a jusante da rampa de carros. O segundo é a formação de uma frente de onda a montante da rampa de carros, que acontece pelo fato de existir um trecho de veículos com velocidade superior a velocidade encontrada no trecho posterior ao fluxo de carros.

A iteração no tempo da variável densidade pode ser observada nas figuras a seguir:



*Figura 5.15 - Iteração no tempo da densidade para  $t=133s$ .*



*Figura 5.16 - Iteração no tempo da densidade para  $t=400s$ .*

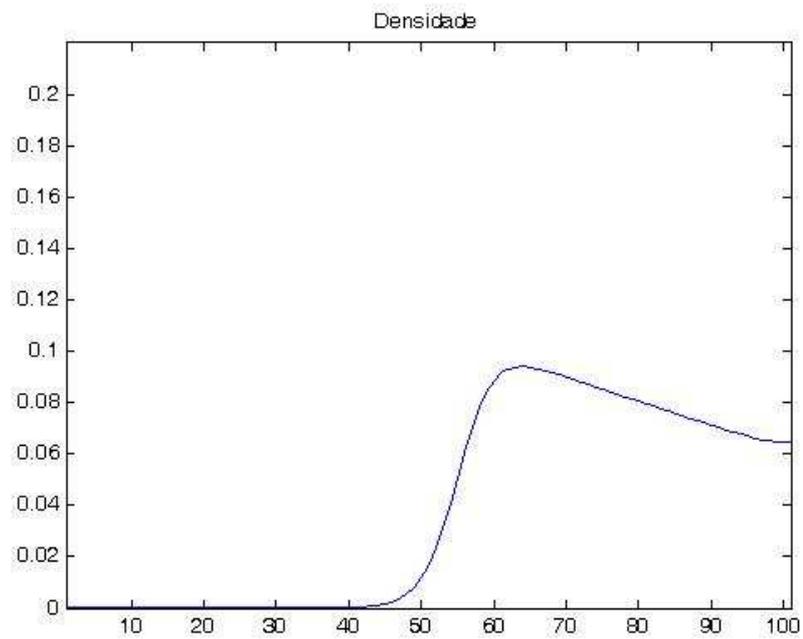


Figura 5.17 - Iteração no tempo da densidade para  $t=666s$ .

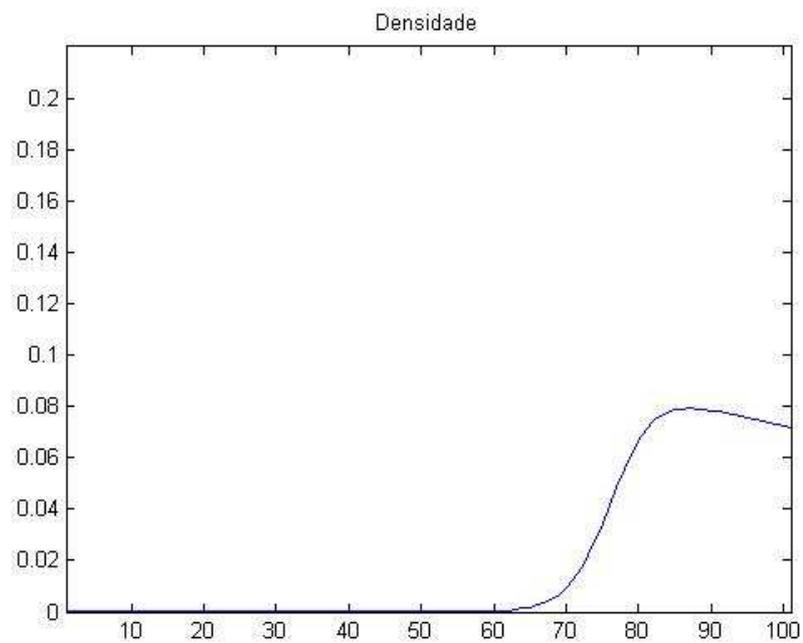


Figura 5.18 - Iteração no tempo da densidade para  $t=933s$ .

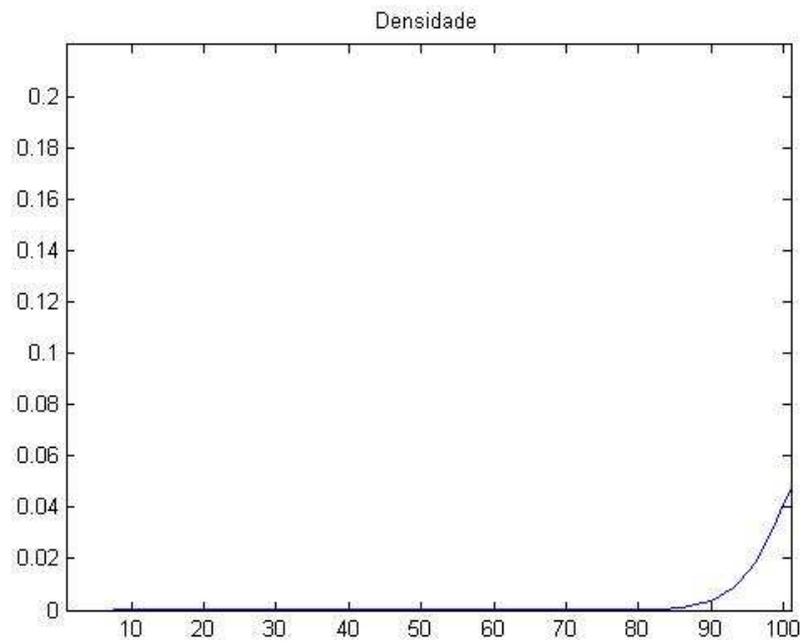


Figura 5.19 - Iteração no tempo da densidade para  $t=1199s$ .

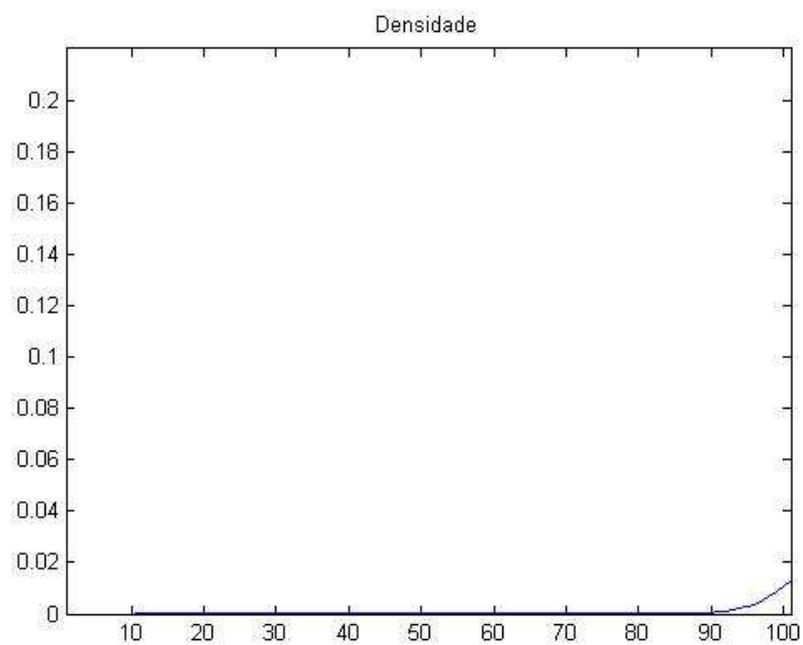
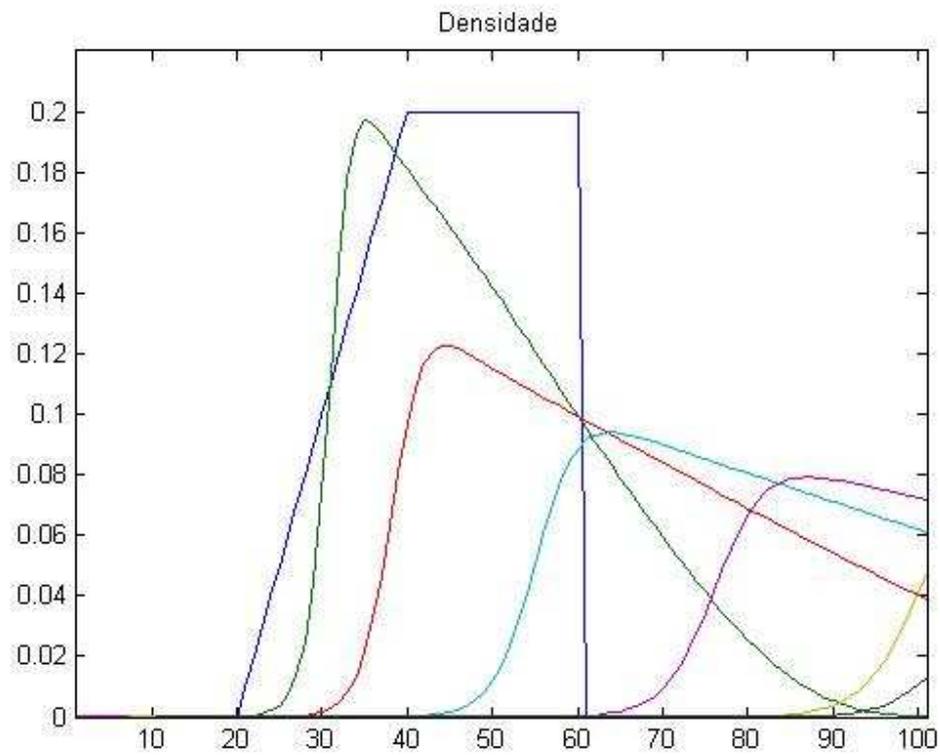


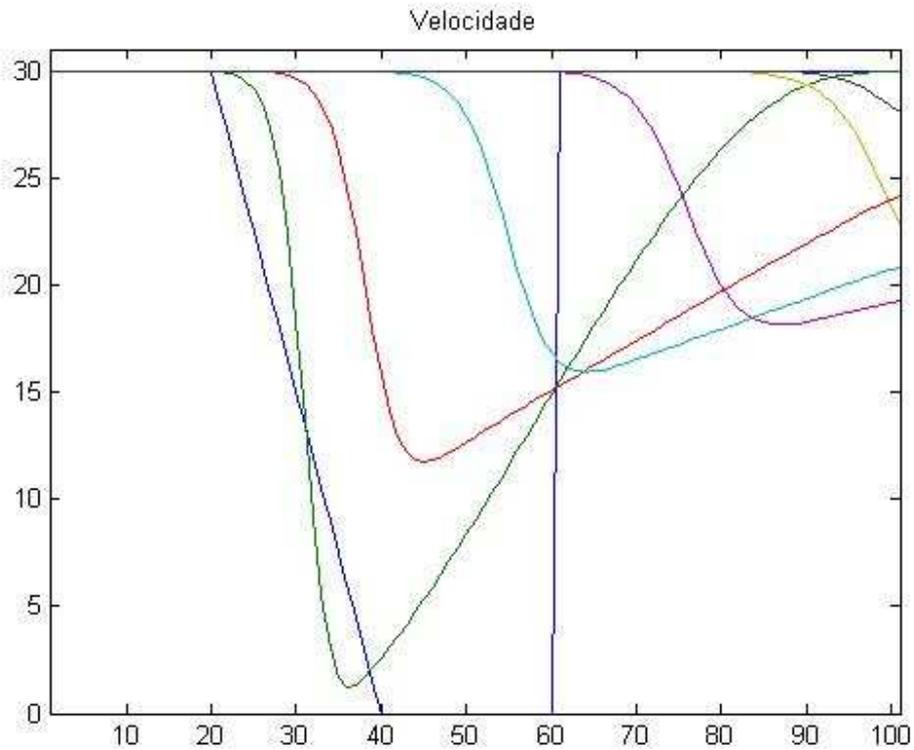
Figura 5.20 - Iteração no tempo da densidade para  $t=1266s$ .

Para se ter uma visão mais geral da propagação dos carros, pode-se observar na figura abaixo todas as iterações mostradas acima em um único gráfico:



*Figura 5.21 - Iteração no tempo da densidade.*

Para a velocidade o comportamento é similar, de modo que a iteração no tempo da velocidade, para os mesmos tempos sinalizados na densidade se dá como mostrado abaixo:



*Figura 5.22 - Iteração no tempo da velocidade.*

Na figura 4.27 fica evidente a formação da frente de onda a montante da rampa de carros.

Novamente o fluxo de carros se dissipa quase totalmente em 1266 segundos, de modo que a velocidade média de propagação da onda é novamente dada por:

$$\bar{V} = \frac{16000m}{1266s} = 12,63m / s = 45,49km / h$$

*Eq. 52*

## 5.4. Problemas Stop-and-go

O problema de *stop-and-go* consiste em para uma condição inicial de carros permitir o movimento livremente dos mesmos, e após certo tempo bloquear em certo trecho o movimento abruptamente.

Para este trabalho de formatura, foram consideradas duas condições iniciais distintas: ambas com densidades constantes em certo trecho (degrau unitário), porém a primeira com valor de densidade de 0,15 veículos por metro e a outra com o valor de densidade de 0,20 veículos por metro. Para a velocidade novamente foi considerado a velocidade de equilíbrio para a densidade imposta inicialmente.

Assim sendo, foram consideradas como condições de contorno para o primeiro nó como sendo a velocidade de equilíbrio para a respectiva densidade, e para o último nó foi imposta uma velocidade nula. Deste modo a distribuição de carros inicial propagaria seu movimento até chegar ao final do trecho, quando seria forçada a parar.

As simulações para estes dois casos podem ser visualizadas nas figuras abaixo:

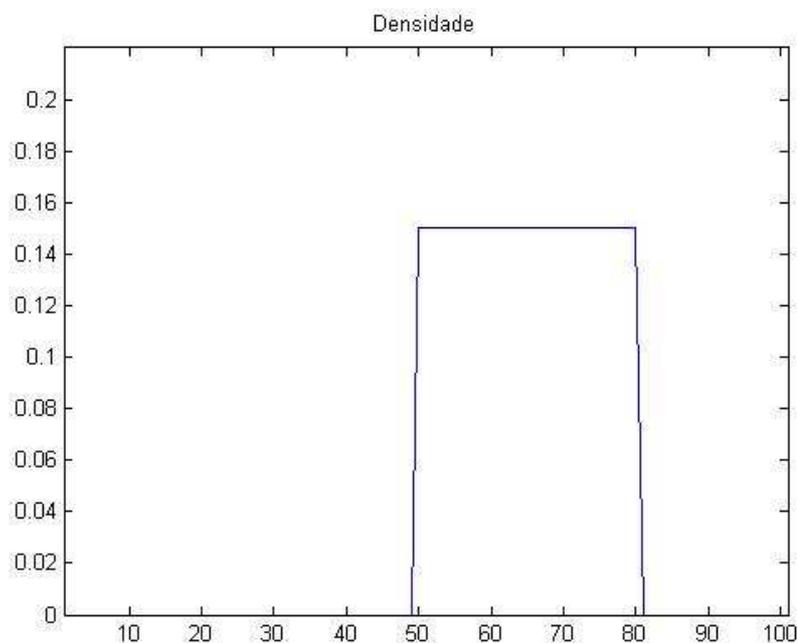


Figura 5.23 – Condição inicial para a densidade de 0,15 veh/m, para o problema de stop-and-go.

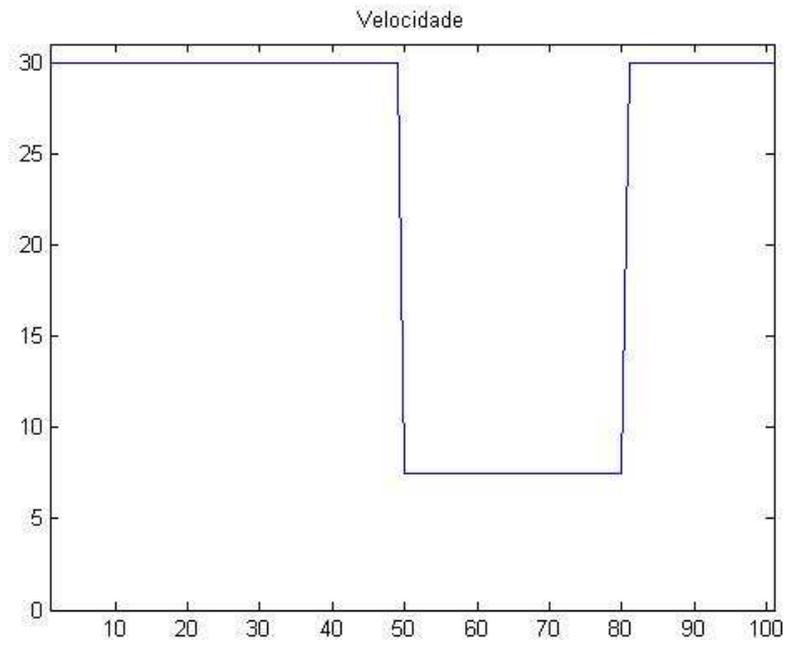


Figura 5.24 - Condição inicial para a velocidade ( $\rho = 0,15 \text{ veh/m}$ ).

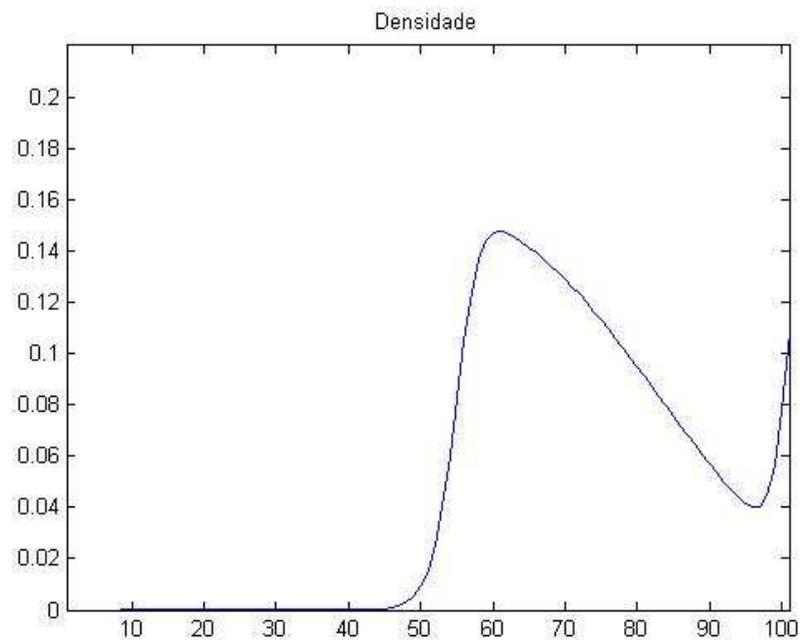


Figura 5.25 – Iteração no tempo da densidade para  $t=133s$ .

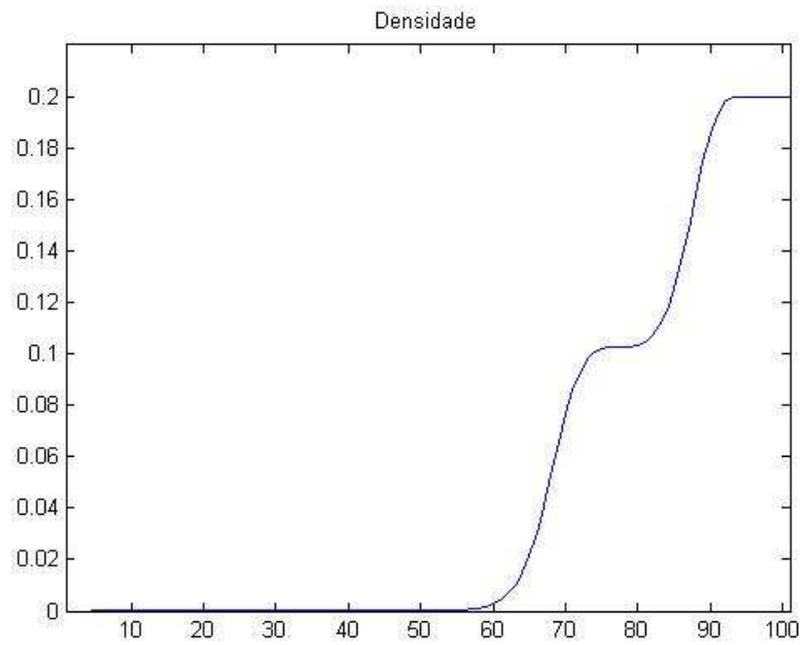


Figura 5.26 - Iteração no tempo da densidade para  $t=400s$ .

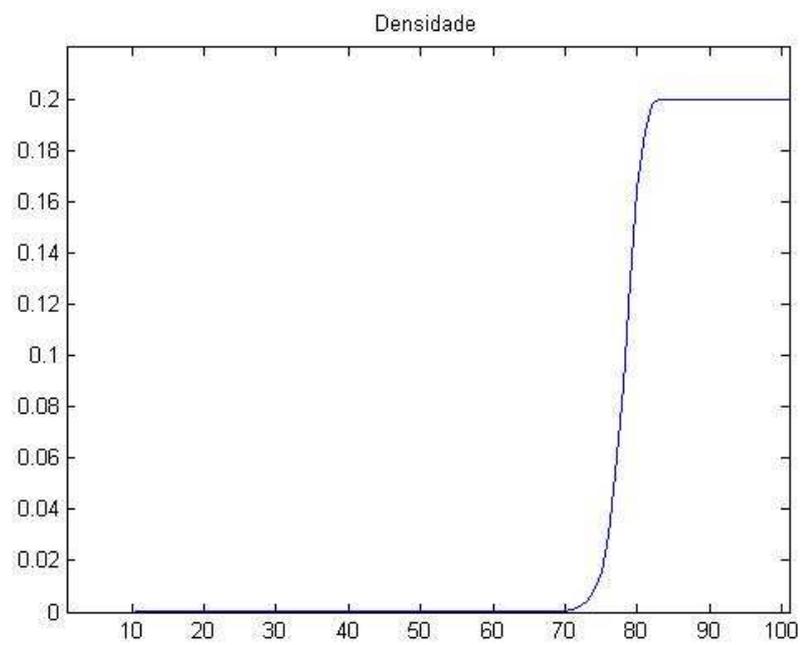
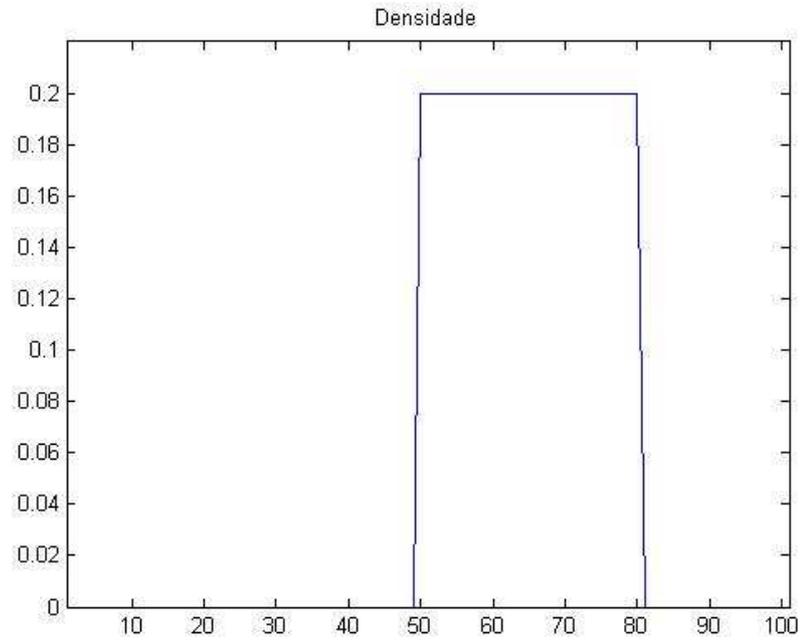


Figura 5.27 - Iteração no tempo da densidade para  $t=666s$ .

Para o caso da densidade de 0,20 veículos por metro, temos os seguintes gráficos:



*Figura 5.28 - Condição inicial para a densidade de 0,15 veh/m, para o problema de stop-and-go.*

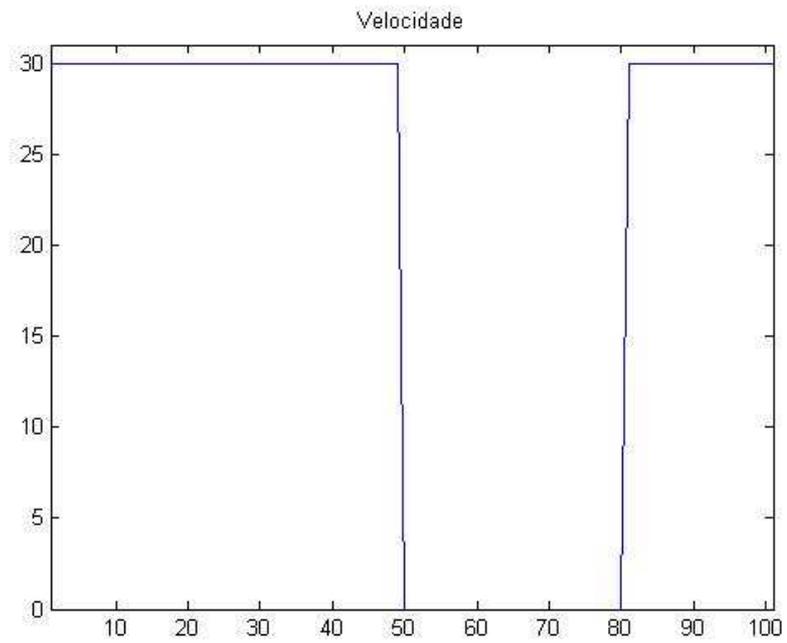


Figura 5.29 - Condição inicial para a velocidade ( $\rho = 0,15 \text{ veh/m}$ ).

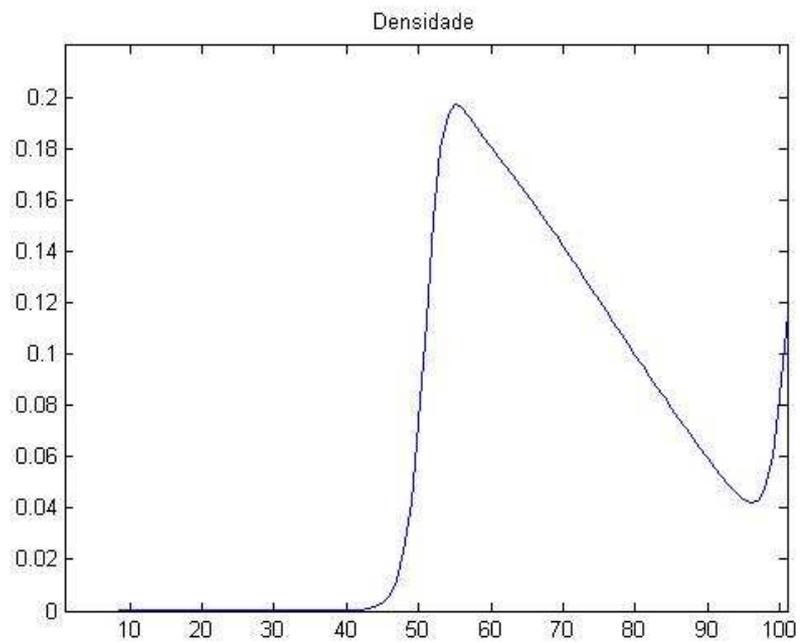


Figura 5.30 - Iteração no tempo da densidade para  $t=133s$ .

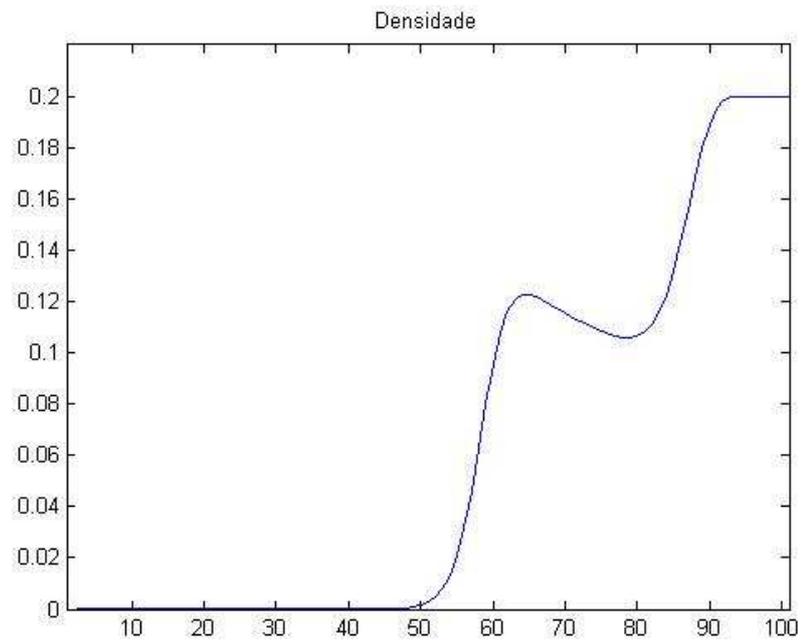


Figura 5.31 - Iteração no tempo da densidade para  $t=400s$ .

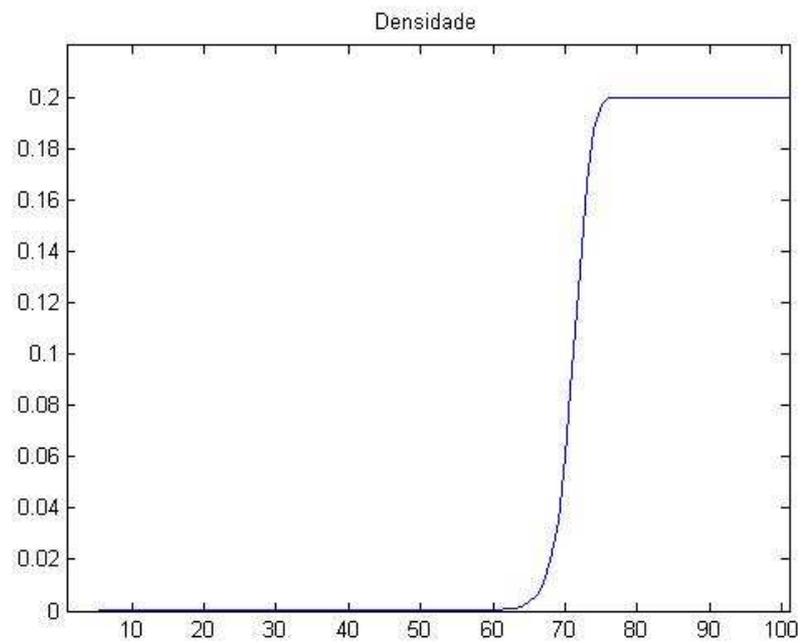


Figura 5.32 - Iteração no tempo da densidade para  $t=666s$ .

A primeira consideração a ser feita a respeito dos dois ensaios acima, é que pelo princípio da conservação da massa (neste caso dos carros), no instante final da simulação o degrau resultante deveria apresentar a mesma área que o degrau inicial, para os dois casos. Ainda deveríamos notar que o degrau de carros deveria ser o mesmo no instante final, porém isso não ocorre pelo fato de que a introdução da viscosidade numérica acaba por “arredondar” as arestas mais acentuadas.

Para o primeiro caso apresentado, a área inicial total é equivalente a:

$$A_{inicial} = Base \cdot Altura = 30 \cdot 0,15 = 4,5$$

Eq. 53

Para o caso final, podemos aproximar a área total pelo retângulo vermelho mostrado na figura abaixo:

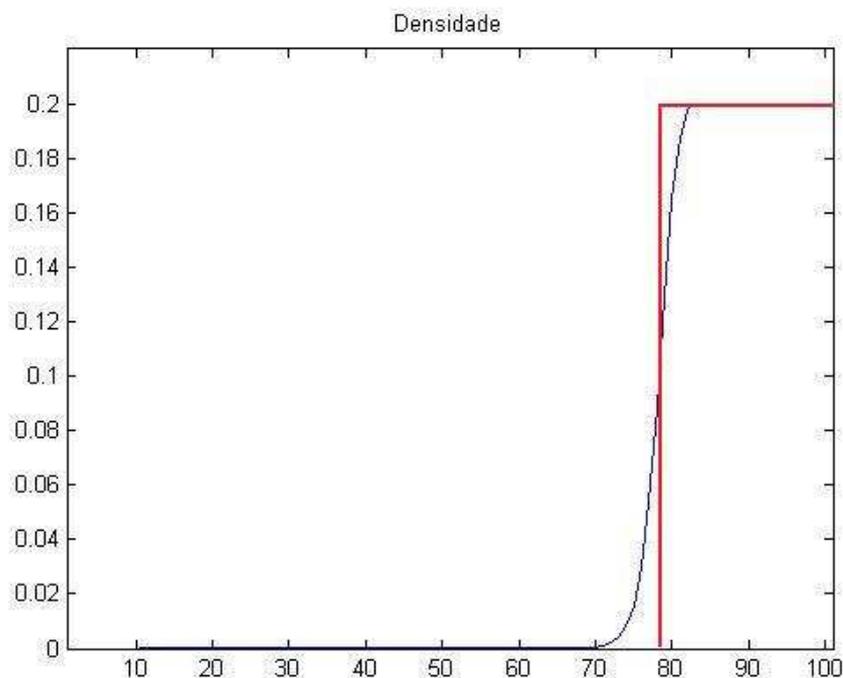


Figura 5.33 – Aproximação da área final para o problema de stop-and-go, para densidade de 0,15 veh/m.

$$A_{final} = Base \cdot Altura = 22 \cdot 0,20 = 4,4$$

Eq. 54

Para o segundo caso, podemos fazer o mesmo processo:

$$A_{inicial} = Base \cdot Altura = 30 \cdot 0,20 = 6,0$$

Eq. 55

E para o caso final:

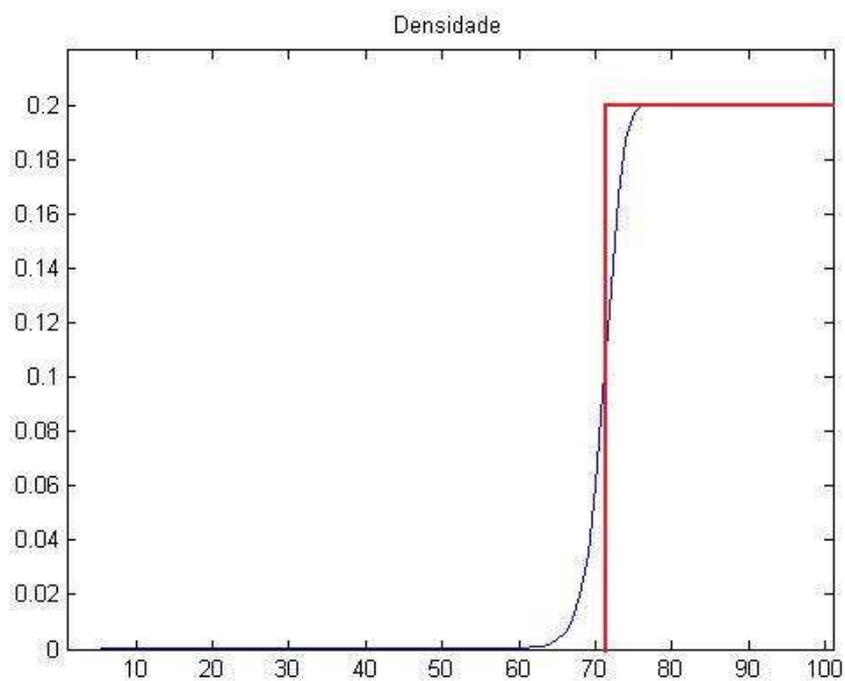


Figura 5.34 - Aproximação da área final para o problema de stop-and-go, para densidade de 0,20 veh/m.

Que resulta numa área de:

$$A_{final} = Base \cdot Altura = 28 \cdot 0,20 = 5,6$$

Eq. 56

Em ambos os casos é possível verificar que área inicial e a área final não se conservam. Em outras palavras, podemos dizer que durante a simulação ocorre uma “dissipação de carros”.

Este fato ocorre principalmente por causa da introdução da viscosidade numérica, para a resolução numérica do problema, porém isto será discutido mais detalhadamente a seguir.

Problemas de *stop-and-go* podem ser mais bem entendidos no dia a dia como vias que possuem semáforos. Periodicamente os mesmos são abertos e fechados, fazendo com que os carros possam fluir normalmente ou parar.

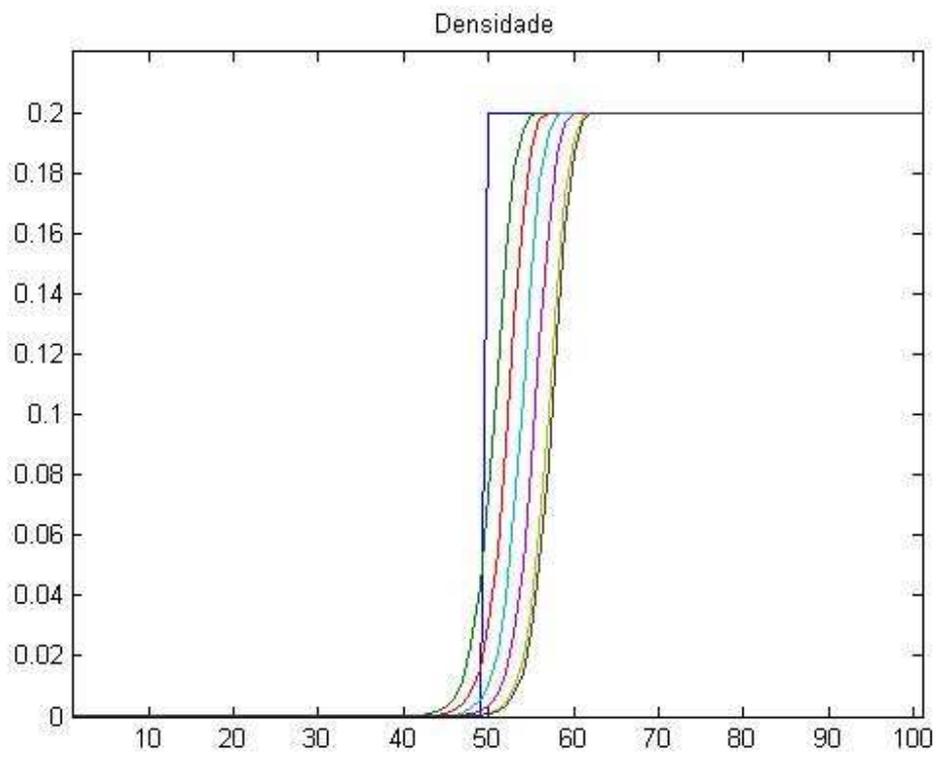
Assim sendo, o estudo deste tipo de problema é imprescindível para um correto entendimento da dinâmica de um tráfego veicular.

## 5.5. Problemas com a viscosidade numérica

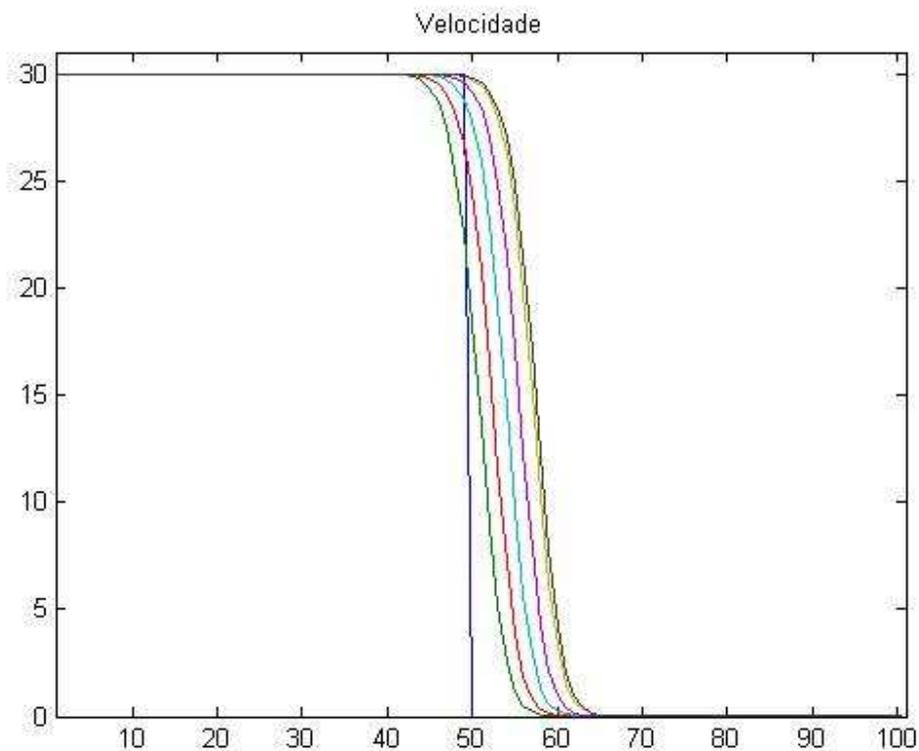
Com o intuito de se alcançar a convergência do sistema, foi introduzida no programa uma viscosidade numérica.

Tal “artifício numérico” foi utilizado por causa da natureza instável do sistema e por causa dos altos gradientes da densidade e da velocidade que se observaram nos exemplos simulados. Porém, em contra partida, tal artifício numérico acabou apresentando o fenômeno de difusividade numérica. A cada iteração o programa acaba por dissipar carros, de modo que a conservação de carros não é respeitada.

Tal fenômeno pode ser mais bem visualizado na figura abaixo:



*Figura 5.35 – Propagação da densidade, para o caso das condições de contorno travadas.*



*Figura 5.36 - Propagação da velocidade, para o caso das condições de contorno travadas.*

As condições de contorno impostas para este caso foram velocidades nulas nos dois extremos, ou seja, não deveria entrar nem sair nenhum carro. Entretanto pode-se observar uma propagação gradual da frente de onda. Isso é explicado pelo fato de que o método da viscosidade numérica acaba por suavizar os gradientes altos da velocidade e da densidade, fazendo assim com que ocorra uma dissipação dos carros.

Para o caso acima exposto, o que deveria ocorrer é que todas as linhas se concentrassem em cima da linha inicial azul, pois a densidade de carros já é máxima e não é permitida a passagem de carros no último nó.

## 6. CONCLUSÕES

Em relação aos resultados obtidos, pode-se concluir que o objetivo deste trabalho de graduação alcançou os objetivos propostos, pois apresentou resultados de uma simulação de um tráfego veicular coerentes com o observado em outros trabalhos e com constatações empíricas de um tráfego veicular real.

Sobre a simulação de um tráfego em uma via, tendo como condição inicial um degrau de carros, pode-se observar a propagação da frente de onda a jusante do degrau de carros, e certa manutenção da frente de onda a montante do degrau de carros. Isso se deve ao fato de que a informação de ausência de carros à frente, chega primeiramente aos primeiros carros da fila, e esta informação vai se propagando passo a passo aos carros de trás, com certo atraso. Deste modo, ao se iniciar a simulação, para os carros mais atrás do degrau, a informação presente é a existência de carros à frente, de modo que o movimento tarda a se propagar. Com o passar da simulação, alguns carros passam a sair da via, de modo que a frente de onda, até então com um formato de rampa, começa a ficar horizontal, pelo fato de o escoamento ficar mais homogêneo.

A respeito da segunda simulação, além dos aspectos já observados na primeira simulação, pode-se verificar a formação de uma frente de onda a montante da rampa de carros. Como os carros mais atrás possuem uma velocidade maior que a velocidade encontrada a frente, estes tendem a frear, fazendo com que a densidade ao se aproximar do grupo de carros tenda a densidade máxima admissível.

Em relação aos problemas de *stop-and-go*, foi possível se verificar inicialmente a propagação da frente de onda, e em seguida observou-se a concentração de carros ao final da via. De um modo geral, os resultados para estas simulações foram bem satisfatórios, pecando apenas no que tange a não conservação dos carros e ao formato “arredondado” do degrau unitário de carros. Ao final da simulação, as dissipações de carros observadas foram, respectivamente, 2,2% e 6,7%.

Um próximo passo para o estudo do problema de um tráfego veicular seria a obtenção de uma nova metodologia para a iteração temporal do sistema, de modo que

não se empregasse a viscosidade numérica para obter a convergência do programa, eliminando assim o problema de dissipação de carros, e as formas “arredondadas”. Para isto, deveria ser introduzido um método de estabilização através das funções de peso.

Estudos de um tráfego veicular estão cada vez mais presentes, e isso não ocorre simplesmente pelo caráter acadêmico, mas sim pela demanda das grandes cidades. O espaço físico para locomoção de veículos é fixo, de modo que para se otimizar o fluxo de carros e diminuir as longas esperas no trânsito surge a necessidade de se controlar os tráfegos nas vias.

## 7. BIBLIOGRAFIA

[1] GREIFENEDER, J.; CELLIER, F. E. Modeling Convective Flows Using Bond Graphs. INTERNATIONAL CONFERENCE ON BOND GRAPH MODELING AND SIMULATION, 5<sup>th</sup>, 2001, Phoenix. Proceedings ICBGM'01. pp. 276-284.

[2] BALIÑO, J. L.; LARRETEGUY, A. E.; GANDOLFO RASO, E. F. A general bond graph approach for computational fluid dynamics. Part I: Theory, 2001 International Conference on Bond Graphs Modeling and Simulation (ICBGM 2001), The Society for Computer Simulation, pp. 41-46. ISBN 1-56555-221-0, 2001.

[3] BALIÑO, J. L. Galerkin finite element method for incompressible thermofluid flows framed within the bond graph theory. Simulation Modelling Practice and Theory (2009).

[4] GANDOLFO RASO, E. F.; LARRETEGUY, A. E.; BALIÑO, J. L. Bond-Graph Modeling of 1-D Compressible Flows, Second IEEE International Conference on System, Man and Cybernetics (SMC'02), Paper TA1B4 (in CD, IEEE No. 02CH37349C, ISBN 0-7803-7438-X), 6p., 2002.

[5] BALIÑO, J. L. Modeling one-dimensional incompressible duct flows. In: EUROPEAN CONFERENCE ON MODELLING AND SIMULATION, 20<sup>th</sup>, 2006, Sankt Agustin. Proceedings. Sankt Agustin: Fachhochschule Bonn-Rhein-Sieg, 2006. pp. 657-662. CD-ROM.

[6] GANDOLFO RASO, E. F.; BALIÑO, J. L. A general bond graph approach for computational fluid dynamics. Part II: Applications, International Conference on Bond Graphs Modeling and Simulation (ICBGM'2001), The Society for Computer Simulation, pp. 47-52. ISBN 1-56555-221-0, 2001.

[7] BORUTZKY, W. Bond Graphs: A methodology for modelling multidisciplinary dynamic systems. Erlangen: SCS, 2004. 440 p.

[8] COOK, R. D. Finite element modeling for stress analysis. New York: Wiley, 1995. 320 p.

- [9] FLETCHER, C. A. J. Computational Techniques for Fluid Dynamics 1: Fundamental and General Techniques. 2<sup>nd</sup> ed. Berlin: Springer-Verlag, 1991. 401 p.
- [10] HUNTER, P.; PULLAN, A. FEM/BEM Notes. Apostilas para disciplina. Department of Engineering Science. The University of Auckland. New Zealand, 2003.
- [11] KARNOPP, D. C.; MARGOLIS, D. L.; ROSENBERG, R. C. System Dynamics: Modeling and Simulation of Mechatronic Systems. 3<sup>rd</sup> ed. New York: Wiley-Interscience, 2000. 507 p.
- [12] E. P. FAHRENTHOLD, M. VENKATARAMAN, Eulerian bond graph for fluid continuum dynamics modeling , ASME Journal of Dynamic Systems, Measurement, and Control 118 (1996) 48-57.
- [13] CUVELIER, C., SEGAL, A. & VAN STEENHOVEN, A. A., Finite Element Methods and Navier-Stokes Equations, D. REIDEL PUBLISHING COMPANY, Holland, 1986.
- [14] LIGHHIL, M.J. & WHITHAM, G.B., On kinematic waves II. *A theory of traffic flow on long crowded roads*, *Proc. Roy. Soc. Ser.*, pp317-345, 1955.
- [15] JIANG, R., WU, K.S. & ZHU, Z.J., A new continuum model for traffic flows and numerical tests, *Transportation Research Part B*, vol. 36, pp.405-419, 2002.
- [16] BESKOS, D.E., MICHALOPOULOS, P.G. & LIN, J.K, Analysis of traffic flow by the finite element method, *Appl. Math. Modelling*, vol. 9, pp. 358-365, 1985.

## 8. ANEXOS

### A) Programa em Matlab

```
function traffic_model_03_11_correto_CC_des_1
warning off all
%format short
%initialisation
% L- the length of the road; T- total time of simulation;
% c,tau -parameters; left, right - between these two points we have a
different
% density
%
L[m],romax[veh/m],vmax[m/s],n[number of
elements],T[s],c[m/s],tau[s],epsi, fi
[h,romax,vmax,n,T,c,tau,epsi,fi,rhot,vt]=initialisation(20000, 0.2,
30, 101, 1270, 11, 10, 0.0000001,0.2);

%other elements
rho=zeros(n,1);
v=zeros(n,1);

rhonew=zeros(n,1);
vnew=zeros(n,1);

ro_n=zeros(n,1);
v_n=zeros(n,1);

t=0; % starting time
clc;

Afisare_(rhot,vt); % drawing the initial conditions
rho_plot(:,1)=rhot;
v_plot(:,1)=vt;
cont=1;
pause();

dt=h/(5*vmax); % dt - time step

while (t+dt<=T) % main loop
```

```

rho=rhot;
v=vt;

t=t+dt %#ok<NOPRT>
cont=cont+1;

errorflag=1;
step=0; %to see at which step of iteration we are

while (errorflag==1)

    step=step+1;

    if step>1

        for i=2:n-1
            rho(i)=ro_nn(i)+fi*(ro_n(i)-ro_nn(i));
            v(i)=v_nn(i)+fi*(v_n(i)-v_nn(i));
        end

        rho(1)=rho(1)+fi*(rhonew(1)-rho(1));
        rho(n)=rho(n)+fi*(rhonew(n)-rho(n));

        v(1)=v(1)+fi*(vnew(1)-v(1));
        v(n)=v(n)+fi*(vnew(n)-v(n));

    end

    %the matrices are calculated
    Ar=Aro_(v,dt,h);
    br=bro_(rhot,v,h,dt);
    Avv=Av_(rho,v,dt,h,tau,c);
    bvv=bv_(rho,vt,dt,h,tau);

    %boundary condition
    Ar(1,:)=0;
    Ar(1,1)=1;
    br(1)= 0;
    Ar(n,:)=0;
    Ar(n,n)=1;
    br(n)= 0.2;

    %the correction of the matrices: we verify if we have lines
with all elements equal
    %to zero. If we find a line and the corresponding element from
the right vector equal also with zero
    %we put 1 on the diagonal of the matrix and the value
corresponding to the anterior step in the right vector
    for i=1:n
        poflag=0;
        for j=1:n

```

```

        if (Ar(i,j)~=0)
            poflag=1;
        end
    end
    if (poflag==0) && (br(i)==0)
        Ar(i,i)=1;
        br(i)=rhot(i);
    end
end

for i=1:n
    poflag=0;
    for j=1:n
        if (Avv(i,j)~=0)
            poflag=1;
        end
    end
    if (poflag==0) && (bv(i)==0)
        Avv(i,i)=1;
        bv(i)=vt(i);
    end
end

%find the solution of the system: Ax=b

P=zeros(n,n);
for ii=1:n
    for jj=1:n
        P(ii,ii)=P(ii,ii)+sqrt(Ar(ii,jj)*Ar(ii,jj));
    end
end

G=P\Ar;
ty=P\br;

rhonew=G\ty; %the solution

P1=zeros(n,n);
for ii=1:n
    for jj=1:n
        P1(ii,ii)=P1(ii,ii)+sqrt(Avv(ii,jj)*Avv(ii,jj));
    end
end

G1=P1\Avv;
ty1=P1\bv;

vnew=G1\ty1; %the solution

% because the density and velocity must be only positive we
verify

```

```

% if we have negative values and we put 0 there

errorflag=0;

for i=1:n
    if rhonew(i)<0
        rhonew(i)=0;
    end
    if vnew(i)<0
        vnew(i)=0;
    end
end

% the density must have a maximal value

for i=1:n
    if rhonew(i)>romax
        rhonew(i)=romax;
    end
end

ro_nn=ro_n; %save the old smoothed variables
v_nn=v_n;

if step==1
    ro_nn=rho;
    v_nn=v;
end;

%artificial viscosity
for i=2:n-1
    gamma_ro(i)=(rhonew(i+1)-rhonew(i-1))/romax;
    gamma_v(i)=(vnew(i+1)-vnew(i-1))/vmax;
    niu_ro(i)=1/(1+exp(-gamma_ro(i)));
    niu_v(i)=1/(1+exp(-gamma_v(i)));
    ro_n(i)=rhonew(i)+(1-niu_ro(i))*(0.5*(rhonew(i+1)-
rhonew(i))+0.5*(rhonew(i-1)-rhonew(i)));
    v_n(i)=vnew(i)+(1-niu_v(i))*(0.5*(vnew(i+1)-
vnew(i))+0.5*(vnew(i-1)-vnew(i)));
end

%we verify if the vectors converge to solution for inner nodes

for i=2:n-1
    if ro_n(i)<=epsi
        if abs(ro_n(i)-ro_nn(i))>=epsi
            errorflag=1;
            x='inner1!!!'
        end
    else
        if abs((ro_n(i)-ro_nn(i))/ro_n(i))>=epsi

```

```

        errorflag=1;
        x='inner2!!!'
    %
    end
end
for i=2:n-1
    if v_n(i)<=epsi
        if abs(v_n(i)-v_nn(i))>=epsi
            errorflag=1;
            x='inner3!!!'
        %
        end
    else
        if abs((v_n(i)-v_nn(i))/v_n(i))>=epsi
            errorflag=1;
            x='inner4!!!'
        %
        end
    end
end

%we verify if the vectors converge to solution for boundary nodes

if rhonew(1)<=epsi % for the first node - density
    if abs(rhonew(1)-rho(1))>=epsi
        errorflag=1;
        x='boundary1!!!'
    %
    end
else
    if abs((rhonew(1)-rho(1))/rhonew(1))>=epsi
        errorflag=1;
        x='boundary2!!!'
    %
    end
end

if rhonew(n)<=epsi %for the last node - density
    if abs(rhonew(n)-rho(n))>=epsi
        errorflag=1;
        x='boundary3!!!'
    %
    end
else
    if abs((rhonew(n)-rho(n))/rhonew(n))>=epsi
        errorflag=1;
        x='boundary4!!!'
    %
    end
end

% rhonewn=rhonew(n)

```

```

%      rhon=rho(n)
%      pause();

if vnew(1)==0 % for the first node - velocity
    if abs(vnew(1)-v(1))>=epsi
        errorflag=1;
%         x='boundary5!!!'

    end
else
    if abs((vnew(1)-v(1))/vnew(1))>=epsi
        errorflag=1;
%         x='boundary6!!!'

    end
end

if vnew(n)==0 %for the last node - velocity
    if abs(vnew(n)-v(n))>=epsi
        errorflag=1;
%         x='boundary7!!!'

    end
else
    if abs((vnew(n)-v(n))/vnew(n))>=epsi
        errorflag=1;
%         x='boundary8!!!'

    end
end

%      vnewn=vnew(n)
%      vn=v(n)
%      pause();

end      %for while with errorflag

rhot=ro_n;
vt=v_n;
rhot(1)=rhonew(1);
vt(1)=vnew(1);
rhot(n)=rhonew(n);
vt(n)=vnew(n);

Afisare_(rhot,vt); %the values for the current step
rho_plot(:,cont)=rhot;
v_plot(:,cont)=vt;

```

```

    % when we want to see which are the values in a specific moment of
time
    % we fix the value and we stop the execution of the loop

end      % for while with dt

pause;

figure(2)
x=1:1:101;
plot(x,rho_plot(:,1),x,rho_plot(:,100),x,rho_plot(:,300),x,rho_plot(:,5
00),x,rho_plot(:,700),x,rho_plot(:,900),x,rho_plot(:,950)); %#ok<COLND>
axis([1 n 0 0.22]) ;
title('Densidade');

figure(3)
x=1:1:101;
plot(x,v_plot(:,1),x,v_plot(:,100),x,v_plot(:,300),x,v_plot(:,500),x,v_
plot(:,700),x,v_plot(:,900),x,v_plot(:,950)); %#ok<COLND>
axis([1 n 0 31]) ;
title('Velocidade');

figure(4)
x=1:1:101;
plot(x,rho_plot(:,1)); %#ok<COLND>
axis([1 n 0 0.22]) ;
title('Densidade');

figure(5)
x=1:1:101;
plot(x,rho_plot(:,100)); %#ok<COLND>
axis([1 n 0 0.22]) ;
title('Densidade');

figure(6)
x=1:1:101;
plot(x,rho_plot(:,300)); %#ok<COLND>
axis([1 n 0 0.22]) ;
title('Densidade');

figure(7)
x=1:1:101;
plot(x,rho_plot(:,500)); %#ok<COLND>
axis([1 n 0 0.22]) ;
title('Densidade');

figure(8)
x=1:1:101;
plot(x,rho_plot(:,700)); %#ok<COLND>
axis([1 n 0 0.22]) ;
title('Densidade');

```

```
figure(9)
x=1:1:101;
plot(x,rho_plot(:,900)); %#ok<COLND>
axis([1 n 0 0.22]) ;
title('Densidade');
```

```
figure(10)
x=1:1:101;
plot(x,rho_plot(:,950)); %#ok<COLND>
axis([1 n 0 0.22]) ;
title('Densidade');
```

```
figure(11)
x=1:1:101;
plot(x,v_plot(:,1)); %#ok<COLND>
axis([1 n 0 31]) ;
title('Velocidade');
```

```
figure(12)
x=1:1:101;
plot(x,v_plot(:,100)); %#ok<COLND>
axis([1 n 0 31]) ;
title('Velocidade');
```

```
figure(13)
x=1:1:101;
plot(x,v_plot(:,300)); %#ok<COLND>
axis([1 n 0 31]) ;
title('Velocidade');
```

```
figure(14)
x=1:1:101;
plot(x,v_plot(:,500)); %#ok<COLND>
axis([1 n 0 31]) ;
title('Velocidade');
```

```
figure(15)
x=1:1:101;
plot(x,v_plot(:,700)); %#ok<COLND>
axis([1 n 0 31]) ;
title('Velocidade');
```

```
figure(16)
x=1:1:101;
plot(x,v_plot(:,900)); %#ok<COLND>
axis([1 n 0 31]) ;
title('Velocidade');
```

```
figure(17)
x=1:1:101;
plot(x,v_plot(:,950)); %#ok<COLND>
axis([1 n 0 31]) ;
```

```

title('Velocidade');

%
% here start the subroutines use in the main program
%

%
% here we make the initialisation of the variables
%

function
[h_, romax_, vmax_, n_, T_, c_, tau_, epsi_, fi_, rhot_, vt_] = initialisation(L, romax, vmax, n, T, c, tau, epsi, fi)
    %initialisation
    h_ = L / (n - 1); % m
    romax_ = romax; % veh/m
    vmax_ = vmax; % m/s
    n_ = n; % number of elements
    T_ = T; % s
    c_ = c; % m/s
    tau_ = tau; % s
    epsi_ = epsi;
    fi_ = fi;

    %initial conditions
    rhot_ = zeros(n, 1);
    vt_ = zeros(n, 1);

    %we put the same value for all the segments
    rhot_(1:n) = 0.0;
    vt_(1:n) = 30;

    %we put a different value for the density and velocity for some
    %segments

    for i = 50:80
        rhot_(i) = 0.15;
    end

%
%
%
    for i = left:right
        rhot_(i) = 0.2;
    end

    for i = 1:n
        vt_(i, 1) = vmax_ * (1 - rhot_(i) / romax_);
    end

```

```

end

%
%here we have the drawing of the values
%

function nothing=Afisare_(ro,v)
    subplot(2,1,1),plot(ro,'-');
    axis([1 n 0 0.2]) ;
    title('Density');
    subplot(2,1,2),plot(v,'-');
    axis([1 n 0 30.5]) ;
    title('velocity');
    pause(0.01);
end

%
%the matrix Aro from the system: Aro*ronew=bro
%

function matAro=Aro_(v_l,dt_l,h)
    %matrix Aro
    matAro=zeros(n,n);
    matW=Wb_(v_l)+Wk_(v_l);
    matAro=Kprim_(v_l,h)-matW*dt_l;
end

%
%the vector bro from the system: Aro*ronew=bro
%

function vecbro=bro_(ro_l,v_l,h,dt_l)
    %vector bro
    vecbro=zeros(n,1);
    vecbro=Kprim_(v_l,h)*ro_l+Wg_(v_l)*dt_l;
end

%
%the matrix Av from the system: Av*vnew=bv
%

function matAv=Av_(ro_l,v_l,dt_l,h,tau,c)
    %matrix Av
    matAv=zeros(n,n);
    matAv=M_(ro_l,h)-(Fg_(ro_l,c)-Fr_(ro_l,h,tau)-
Fk_(ro_l,v_l))*dt_l;

```

```

end

%
%the vector bv from the system: Av*vnew=bv
%

function vecbv=bv_(ro_l,v_l,dt_l,h,tau)
    %vector bv
    vecbv=zeros(n,1);
    vecbv=Fe_(ro_l,h,tau)*dt_l+M_(ro_l,h)*v_l;
end

%
%
%

function vector=Wg_(v_l)

    vector=zeros(n,1);
    Wg=zeros(n,1);
    %vector Wg
    Wg(1)=(1.0/2.0)*(v_l(1)^3);
    Wg(n)=-(1.0/2.0)*(v_l(n)^3);
    %final equation formation
    vector=Wg;
end

%
%
%

function mat1=Wb_(v_l)

    mat1=zeros(n,n);
    Wb=zeros(n,n);
    %matrix Wb
    Wb(1,1)=(1.0/40.0)*(-4.0*(v_l(1)^3)-3.0*(v_l(1)^2)*v_l(2)-
2.0*v_l(1)*(v_l(2)^2)-(v_l(2)^3));
    Wb(1,2)=(1.0/40.0)*(-(v_l(1)^3)-2.0*(v_l(1)^2)*v_l(2)-
3.0*v_l(1)*(v_l(2)^2)-4.0*(v_l(2)^3));
    Wb(n,n-1)=(1.0/40.0)*(4.0*(v_l(n-1)^3)+3.0*(v_l(n-
1)^2)*v_l(n)+2.0*v_l(n-1)*(v_l(n)^2)+(v_l(n)^3));
    Wb(n,n)=(1.0/40.0)*((v_l(n-1)^3)+2.0*(v_l(n-
1)^2)*v_l(n)+3.0*v_l(n-1)*(v_l(n)^2)+4.0*(v_l(n)^3));

    for i=2:n-1
        Wb(i,i-1)=(1.0/40.0)*(4.0*(v_l(i-1)^3)+3.0*(v_l(i-
1)^2)*v_l(i)+2.0*v_l(i-1)*(v_l(i)^2)+(v_l(i)^3));

```

```

        Wb(i,i)=(1.0/40.0)*((v_l(i-1)^3)+2.0*(v_l(i-
1)^2)*v_l(i)+3.0*v_l(i-1)*(v_l(i)^2)-3.0*(v_l(i)^2)*v_l(i+1)-
2.0*v_l(i)*(v_l(i+1)^2)-(v_l(i+1)^3));
        Wb(i,i+1)=(1.0/40.0)*(-(v_l(i)^3)-2.0*(v_l(i)^2)*v_l(i+1)-
3.0*v_l(i)*(v_l(i+1)^2)-4.0*(v_l(i+1)^3));
    end
    %final equation formation
    mat1=Wb;
end

```

```

%
%
%

```

```

function mat2=Wk_(v_l)

```

```

    mat2=zeros(n,n);
    Wk=zeros(n,n);
    %matrix Wk
    Wk(1,1)=(1.0/60.0)*(v_l(2)-
v_l(1))*(12.0*(v_l(1)^2)+6.0*v_l(1)*v_l(2)+2.0*(v_l(2)^2));
    Wk(1,2)=(1.0/60.0)*(v_l(2)-
v_l(1))*(3.0*(v_l(1)^2)+4.0*v_l(1)*v_l(2)+3.0*(v_l(2)^2));
    Wk(n,n-1)=(1.0/60.0)*(v_l(n)-v_l(n-1))*(3.0*(v_l(n-
1)^2)+4.0*v_l(n-1)*v_l(n)+3.0*(v_l(n)^2));
    Wk(n,n)=(1.0/60.0)*(v_l(n)-v_l(n-1))*(2.0*(v_l(n-
1)^2)+6.0*v_l(n-1)*v_l(n)+12.0*(v_l(n)^2));
    for i=2:n-1
        Wk(i,i-1)=(1.0/60.0)*(v_l(i)-v_l(i-1))*(3.0*(v_l(i-
1)^2)+4.0*v_l(i-1)*v_l(i)+3.0*(v_l(i)^2));
        Wk(i,i)=(1.0/60.0)*(v_l(i)-v_l(i-1))*(2.0*(v_l(i-
1)^2)+6.0*v_l(i-1)*v_l(i)+12.0*(v_l(i)^2))+(1.0/60.0)*(v_l(i+1)-
v_l(i))*(12.0*(v_l(i)^2)+6.0*v_l(i)*v_l(i+1)+2.0*(v_l(i+1)^2));
        Wk(i,i+1)=(1.0/60.0)*(v_l(i+1)-
v_l(i))*(3.0*(v_l(i)^2)+4.0*v_l(i)*v_l(i+1)+3.0*(v_l(i+1)^2));
    end
    %final equation formation
    mat2=Wk;
end

```

```

%
%
%

```

```

function mat3=Kprim_(v_l,h)

```

```

    mat3=zeros(n,n);
    Kprim=zeros(n,n);
    %matrix K

```

```

Kprim(1,1)=(1/2.0)*h*(1.0/30.0)*(6.0*(v_l(1)^2)+3.0*v_l(1)*v_l(2)+(v_l(
2)^2));

Kprim(2,1)=(1/2.0)*h*(1.0/60.0)*(3.0*(v_l(1)^2)+4.0*v_l(1)*v_l(2)+3.0*(
v_l(2)^2));
    Kprim(n-1,n)=(1/2.0)*h*(1.0/60.0)*(3.0*(v_l(n-1)^2)+4.0*v_l(n-
1)*v_l(n)+3.0*(v_l(n)^2));
    Kprim(n,n)=(1/2.0)*h*(1.0/30.0)*((v_l(n-1)^2)+3.0*v_l(n-
1)*v_l(n)+6.0*(v_l(n)^2));

        for i=2:n-1
            Kprim(i-1,i)=h*(1.0/120.0)*(3.0*(v_l(i-1)^2)+4.0*v_l(i-
1)*v_l(i)+3.0*(v_l(i)^2));
            Kprim(i,i)=h*(1.0/60.0)*((v_l(i-1)^2)+3.0*v_l(i-
1)*v_l(i)+12.0*(v_l(i)^2)+3.0*v_l(i)*v_l(i+1)+(v_l(i+1)^2));

Kprim(i+1,i)=h*(1.0/120.0)*(3.0*(v_l(i)^2)+4.0*v_l(i)*v_l(i+1)+3.0*(v_l
(i+1)^2));
        end
        %final equation formation
        mat3=Kprim;
    end

function matfr=Fr_(ro_l,h,tau)

    matfr=zeros(n,n);
    Fr=zeros(n,n);
    %matrix Fr
    Fr(1,1)=(h/(12.0*tau))*(3.0*ro_l(1)+ro_l(2));
    Fr(1,2)=(h/(12.0*tau))*(ro_l(1)+ro_l(2));
    Fr(n,n-1)=(h/(12.0*tau))*(ro_l(n-1)+ro_l(n));
    Fr(n,n)=(h/(12.0*tau))*(ro_l(n-1)+3.0*ro_l(n));
    for i=2:n-1
        Fr(i,i-1)=(h/(12.0*tau))*(ro_l(i-1)+ro_l(i));
        Fr(i,i)=(h/(12.0*tau))*(ro_l(i-1)+6.0*ro_l(i)+ro_l(i+1));
        Fr(i,i+1)=(h/(12.0*tau))*(ro_l(i)+ro_l(i+1));
    end
    %final equation formation
    matfr=Fr;
end

%
%
%

function matFg=Fg_(ro_l,c)

    matFg=zeros(n,n);
    Fg=zeros(n,n);
    %matrix Fs
    Fg(1,1)=-(1.0/6.0)*c*(2.0*ro_l(1)+ro_l(2));

```

```

Fg(1,2)=(1.0/6.0)*c*(2.0*ro_l(1)+ro_l(2));
Fg(n,n-1)=-(1.0/6.0)*c*(ro_l(n-1)+2.0*ro_l(n));
Fg(n,n)=(1.0/6.0)*c*(ro_l(n-1)+2.0*ro_l(n));
for i=2:n-1
    Fg(i,i-1)=-(1.0/6.0)*c*(ro_l(i-1)+2.0*ro_l(i));
    Fg(i,i)=-(1.0/6.0)*c*(ro_l(i+1)-ro_l(i-1));
    Fg(i,i+1)=(1.0/6.0)*c*(2.0*ro_l(i)+ro_l(i+1));
end
%final equation formation
matFg=Fg;
end

%
%
%

function vecFe=Fe_(ro_l,h,tau)

    vecFe=zeros(n,1);
    Fe=zeros(n,1);
    ve_l=zeros(n,1);
    for i=1:n
        ve_l(i)=vmax*(1.0-ro_l(i)/romax);
    end
    %vector Fsprim

Fe(1)=((h*vmax)/(12.0*tau*romax))*(romax*(4.0*ro_l(1)+2.0*ro_l(2))-
3.0*(ro_l(1)^2)-2.0*ro_l(1)*ro_l(2)-(ro_l(2)^2));
    Fe(n)=((h*vmax)/(12.0*tau*romax))*(romax*(2.0*ro_l(n-
1)+4.0*ro_l(n))-(ro_l(n-1)^2)-2.0*ro_l(n-1)*ro_l(n)-3.0*(ro_l(n)^2));
    for i=2:n-1
        Fe(i)=((h*vmax)/(12.0*tau*romax))*(romax*(2.0*ro_l(i-
1)+8.0*ro_l(i)+2.0*ro_l(i+1))-(ro_l(i-1)^2)-2.0*ro_l(i)*ro_l(i-1)-
6.0*(ro_l(i)^2)-2.0*ro_l(i)*ro_l(i+1)-(ro_l(i+1)^2));
    end
    %final equation formation
    vecFe=Fe;
end

%
%
%

function matFk=Fk_(ro_l,v_l)

    matFk=zeros(n,n);
    Fk=zeros(n,n);
    %matrix Fk
    Fk(1,1)=-
(1.0/12.0)*(v_l(1)*(3.0*ro_l(1)+ro_l(2))+v_l(2)*(ro_l(1)+ro_l(2)));

```

```

Fk(1,2)=(1.0/12.0)*(v_l(1)*(3.0*ro_l(1)+ro_l(2))+v_l(2)*(ro_l(1)+ro_l(2
)));
    Fk(n,n-1)=-(1.0/12.0)*(v_l(n-1)*(ro_l(n-
1)+ro_l(n))+v_l(n)*(ro_l(n-1)+3.0*ro_l(n)));
    Fk(n,n)=(1.0/12.0)*(v_l(n-1)*(ro_l(n-
1)+ro_l(n))+v_l(n)*(ro_l(n-1)+3.0*ro_l(n)));
    for i=2:n-1
        Fk(i,i-1)=-(1.0/12.0)*(v_l(i-1)*(ro_l(i-
1)+ro_l(i))+v_l(i)*(ro_l(i-1)+3.0*ro_l(i)));
        Fk(i,i)=(1.0/12.0)*(v_l(i-1)*(ro_l(i-1)+ro_l(i))-
v_l(i)*(ro_l(i+1)-ro_l(i-1))-v_l(i+1)*(ro_l(i)+ro_l(i+1)));

Fk(i,i+1)=(1.0/12.0)*(v_l(i)*(3.0*ro_l(i)+ro_l(i+1))+v_l(i+1)*(ro_l(i)+
ro_l(i+1)));
    end
    %final equation formation
    matFk=Fk;
end

%
%
%

function matM=M_(ro_l,h)

    matM=zeros(n,n);
    M=zeros(n,n);
    %matrix M
    M(1,1)=(1.0/12.0)*h*(3.0*ro_l(1)+ro_l(2));
    M(2,1)=(1.0/12.0)*h*(ro_l(1)+ro_l(2));
    M(n-1,n)=(1.0/12.0)*h*(ro_l(n-1)+ro_l(n));
    M(n,n)=(1.0/12.0)*h*(ro_l(n-1)+3.0*ro_l(n));
    for i=2:n-1
        M(i-1,i)=(h/12.0)*(ro_l(i-1)+ro_l(i));
        M(i,i)=(h/12.0)*(ro_l(i-1)+6.0*ro_l(i)+ro_l(i+1));
        M(i+1,i)=(h/12.0)*(ro_l(i)+ro_l(i+1));
    end
    %final equation formation
    matM=M;
end

end

```