MAC0209 — Primeiro EP - 2021

Roberto Marcondes Cesar Jr. - Roberto Hirata Jr. - Artur André A. M. Oliveira

17 de maio de 2021

1 Introdução

A disciplina de Modelagem e Simulação do curso de Bacharelado em Ciência da Computação tem como objetivo principal que o aluno se familiarize com a modelagem de sistemas físicos reais e seja capaz de simulá-los através da implementação de algoritmos.

A disciplina tem uma parte teórica e uma prática e é essa que nos interessa neste documento. A prática é cobrada a partir de exercícios programa (*EPs*) que são feitos pelo aluno **individualmente** no seu computador pessoal, ou em algum computador a que tenha acesso. A especificação do exercício será sempre divulgada no *edisciplinas*, assim como a data de entrega e o "link" para a entrega.

Para esta disciplina, para efeitos de avaliação, serão considerados:

- Realização dos experimentos reais.
- Modelagem matemática do sistema físico do experimento e simulação do experimento usando algoritmos baseados no modelo matemático.
- Rigor científico na realização do experimento real, simulado e documentação dos resultados.
- Funcionamento do código. Este item é de fundamental importância para um exercício programa ser considerado entregue. Por funcionamento, entenda-se: o código apresentado implementa o que foi especificado no enunciado?
- Organização e clareza do código. O código é fácil de ler e entender?
- Documentação do código. As passagens mais difíceis do algoritmo tem frases que ajudem o seu entendimento? As variáveis e constantes estão associadas a frases que dizem para que elas servem?

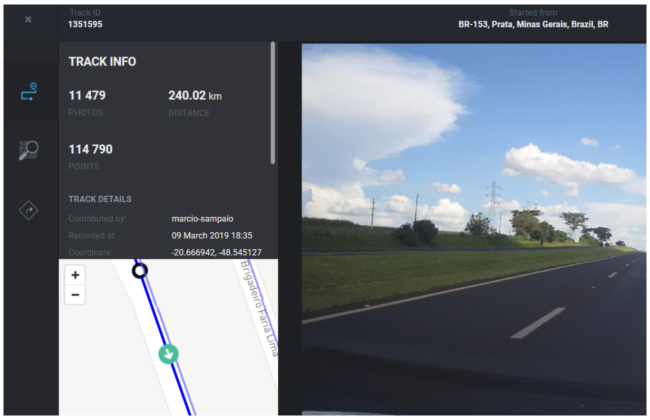


Figura 1: Exemplo de trecho e imagem do KartaView

2 O primeiro exercício programa (EP)

O primeiro EP deste semestre será realizar e relatar experimentos que envolvem a modelagem do Movimento Retilíneo Uniforme (MRU). Para isso, você deve escolher dois trechos de duas rodovias diferentes no KartaView, uma no exterior e uma no Brasil. Os trechos devem ser tais que possa acontecer MRU. Uma vez escolhidos, você vai coletar os dados da trajetória do veículo que forneceu os dados e analisá-los da mesma forma como já fez nos labs, com alguns cuidados a mais que serão explicados nas seções que seguem.

2.1 Modelagem

A modelagem dos sistemas acima não é difícil e foi abordada em sala de aula. Nesta seção faremos uma breve recapitulação.

2.1.1 Coletando dados do KartaView

A plataforma KartaView mantém imagens de ruas, avenidas, rodovias, etc. coletadas pelos próprios usuários da plataforma (i.e. crowdsourced) e disponibilizadas publicamente. No endereço a seguir podemos ver um exemplo de trajeto exibido diretamente na plataforma: `https://kartaview.org/details/1351595/8014`. Note que no canto superior esquerdo temos o identificador do trecho `Track ID 1351595` (veja a Fig. 1). Ao escolher um trecho precisamos observar qual seu identificador afim de podermos coletar do trajeto os dados posteriormente.

Tendo escolhido um trajeto e conhecendo seu identificador, através do endereço de API (Application Programming Interface) `https://api.openstreetcam.org/details` podemos coletar os dados e as imagens deste trajeto. A requisição à um endereço (também chamado de endpoint) de API de um serviço web normalmente se faz através de uma requisição HTTP, neste caso observamos que o verbo HTTP usado é o POST com dados, isso é, um *payload*, que neste caso é um formulário com um campo chamado `id` cujo valor é o identificador de trecho

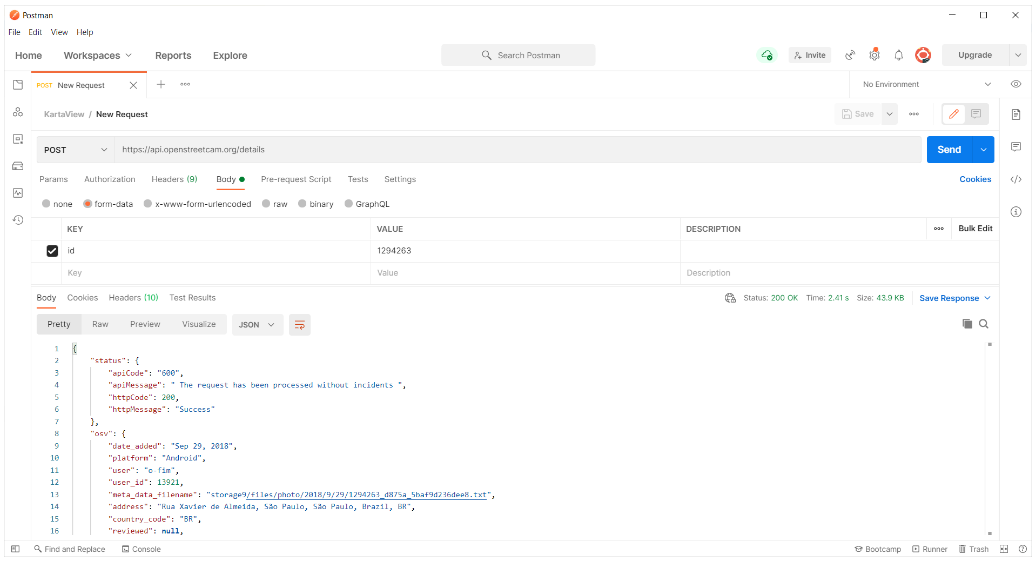


Figura 2: Exemplo de requisição usando o Postman

`Track ID` coletado diretamente pela interface do website do KartaView. Na figura 2 é exibida a interface da aplicação Postman onde o endereço do endpoint, o verbo `POST` e o *payload* contendo o identificador do trajeto foram usados para coletar (em JSON) os dados de um trajeto.

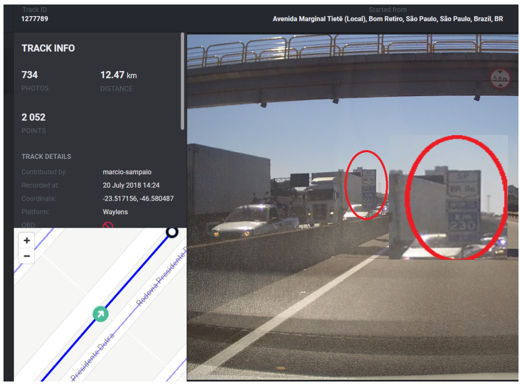
2.1.2 Imagens de Pontos Fiduciais

Uma forma de validarmos as distâncias medidas é através de imagens de pontos fiduciais na estrada, mais especificamente placas que marcam a distância percorrida a partir do início de uma estrada. Por exemplo, na figuras 3a e 3b podemos ver duas placas (Km 229 e Km 230, respectivamente) indicando duas posições de um veículo percorrendo a Rodovia Presidente Dutra. Tendo como base essas placas, sabemos que a distância percorrida entre essas duas imagens deve ser de aproximadamente 1 Km. É claro que há que se levar em conta a distância da câmera até as placas, muito embora ela seja pequena. Por exemplo, usando a fórmula de Haversine explicada abaixo[1], a distância foi de 1.009 Km.

Note que buscar imagens fiduciais no KartaView pode ser um pouco trabalhoso dado que a câmera na maior parte dos trajetos aponta somente para frente, e a qualidade da imagem pode atrapalhar um pouco as vezes. Porém, uma vez definido o trajeto, uma possível solução é buscar as imagens dos pontos fiduciais usando o Google Street View. Uma vez encontrados os locais com os pontos fiduciais, podemos voltar ao KartaView e buscar as mesmas imagens. Apenas como exemplo ilustrativo, observe que nas figuras 4a e 4b temos as mesmas placas capturadas pelo carro do Google Street View.

[1]O site `https://www.movable-type.co.uk/scripts/latlong.html` pode ser usado para fazer o cálculo também.

(a) Km 229 ao longo da Rod. Pres. Dutra



(b) Km 230 ao longo da Rod. Pres. Dutra

Figura 3: Imagens dos pontos fiduciais com placas indicando a localização do veículo ao longo da Rodovia Presidente Dutra.

(a) Km 229 ao longo da Rododovia Presidente Dutra observado no Google Street View

(b) Km 230 ao longo da Rododovia Presidente Dutra observado no Google Street View

Figura 4: Imagens dos pontos fiduciais, observadas no Google Street View, com placas indicando a localização do veículo ao longo da Rodovia Presidente Dutra.

2.1.3 Projetando as coordenadas esféricas num plano

O posicionamento geoespacial dos dados do KartaView são definidos pelo Datum WGS 84 (World Geodetic System 1984), o qual também é identificado pelo código EPSG:4326[2].

Um Datum consiste em:

1. um modelo para o planeta, (no geral elipsoidal, geoidal ou esférico);

2. um ponto de referência;

3. e um sistema de coordenadas.

No caso do WGS 84, o sistema de coordenadas é esférica e o modelo usado para o planeta é elipsoidal. Usando o Datum WGS84, o ponto onde a linha do equador e o meridiano de Greenwich se cruzam possui as coordenadas de latitude (i.e. norte-sul) zero e longitude (i.e. leste-oeste) zero[3].

Uma vez que sabemos qual modelo é usado para representar o planeta e qual o sistema de coordenadas usado para representar a localização de um objeto ao longo de um trajeto, podemos medir a distância entre dois pontos diferentes projetando as posições desses pontos em um plano (usando por exemplo a projeção de Mercator para latitudes até 45 graus a partir da linha do equador). A distância pode ser a euclidiana e, alternativamente, podemos usar formulações mais complicadas para medir distâncias diretamente na superfície de uma esfera ou elipse.

Nos notebooks usados para os labs 1, 2 e 3, foi usado o pacote `pyproj` que faz a projeção de coordenadas esféricas do WGS84 (já levando em conta o modelo usado para o planeta) para coordenadas no plano.

2.1.4 Fórmula de haversine

A fórmula de haversine nos permite medir a distância d, na superfície de uma esfera de raio r, entre dois pontos cujas coordenadas (em latitude e longitude) são ϕ_1, λ_1 e ϕ_2, λ_2 respectivamente, isso é:

$$d = 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\phi_2 - \phi_1}{2}\right) + \cos\phi_1 \cos\phi_2 \sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right) \tag{1}$$

Esta fórmula é baseada na função de mesmo nome, haversine, que por usa vez vem de *half-versine*, onde *versine* é outra função trigonométrica definida em termos da função cosseno,

$$\mathrm{versin}(\theta) = 1 - \cos\theta = 2\sin^2\frac{\theta}{2}$$

[2]O sistema geodético do KartaView é o mesmo do projeto OpenStreetMap, uma vez que ele visa extender o OpenStreetMap com imagens `https://en.wikipedia.org/wiki/KartaView`

[3]Um fato curioso é que este ponto se localiza no oceano e é conhecido como *Null Island*

sendo assim a função haversine é

$$\mathrm{hav}(\theta) = \frac{\mathrm{versin}(\theta)}{2} = \sin^2\frac{\theta}{2}$$

Podemos localizar um ponto em uma esfera, através de coordenadas como latitude e longitude. Note que latitude e longitude formam um sistema ortogonal de coordenadas esféricas. Para calcular o ângulo central (i.e. o ângulo com vértice no centro da esfera) formado entre dois pontos arbitrários na superfície de uma esfera podemos usar a **fórmula** de haversine, a qual irá permitir expressarmos o valor da **função** haversine de θ em termos da latitude e longitude, isso é, considere um par de pontos com as coordenadas (ϕ_1, λ_1) e (ϕ_2, λ_2) respectivamente, sendo θ o ângulo central formado entre eles:

$$\mathrm{hav}(\theta) = \sin^2(\frac{\phi_2 - \phi_1}{2}) + \cos\phi_1\cos\phi_2\sin^2(\frac{\lambda_2 - \lambda_1}{2}) \tag{2}$$

Uma vez que calculamos o valor $\mathrm{hav}(\theta)$ podemos calcular a função inversa de haversine, $\mathrm{archav}(\theta)$, por analogia com a inversa de outras funções trigonométrica como arco seno arcsin) e com isso teremos a distância d na superfície da esfera entre os dois pontos:

$$d = r\,\mathrm{archav}(\mathrm{hav}(\theta)) = r\theta = 2r\frac{\theta}{2} = 2r\arcsin\left(\sqrt{\sin^2\frac{\theta}{2}}\right) = 2r\arcsin\left(\sqrt{\mathrm{hav}(\theta)}\right)$$

onde r é o raio da esfera. Se substituirmos $\mathrm{hav}(\theta)$ pela formula de haversine da equação (2) chegamos na fórmula de distância da equação (1).

2.1.5 Trigonometria esférica

Através da lei esférica dos cossenos podemos derivar a fórmula de haversine e também podemos calcular distâncias na superfície de uma esfera. Usando como referência a esfera unitária da figura 5 a (primeira) lei dos cossenos nos diz que:

$$\cos(c) = \cos(a)\cos(b) + \sin(a)\sin(b)\cos(C) \tag{3}$$

Sendo assim:

$$\cos(C) = \frac{\cos(c) - \cos(a)\cos(b)}{\sin(a)\sin(b)} \tag{4}$$

Podemos generalizar este resultado para uma esfera de raio r, contudo neste caso o lado de tamanho c, oposto ao ângulo subentendido C será rc, ou seja, no caso da esfera unitária o comprimento lado c do triângulo na esfera unitária coincide com o ângulo C, mas se $r \neq 1$ então para aplicar a lei dos cossenos primeiro precisamos dividir o tamanho de cada lado do triângulo na esfera (i.e. a, b e c) por r, isso é,

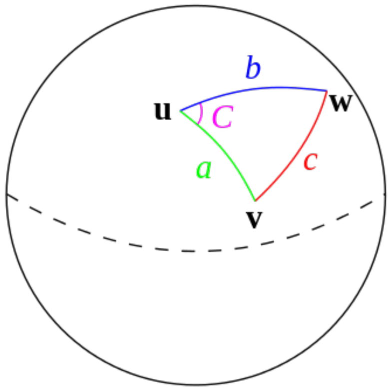


Figura 5: Spherical triangle solved by the law of cosines. - Fonte: `https://en.wikipedia.org/wiki/Haversine_formula`

$$\cos(C) = \frac{\cos\left(\frac{c}{r}\right) - \cos\left(\frac{a}{r}\right)\cos\left(\frac{b}{r}\right)}{\sin\left(\frac{a}{r}\right)\sin\left(\frac{b}{r}\right)} \tag{5}$$

uma vez calculado o valor de $\cos(C)$ podemos calcular o valor da distância c,

$$c = r\arccos(\cos(C)) = rC$$

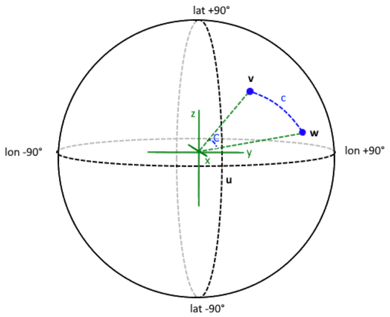
Note que se usarmos como unidade métrica o metro, e assumirmos que o planeta é uma esfera com raio de 6371km, teremos $r = 6371000m$, e ao tentarmos tomar o arco-cosseno de $\cos(C)$ para medir a distância percorrida entre dois pontos relativamente próximos (como no lab 1 em que a distância entre um ponto e o próximo é no geral entre 30 e 35 metros) é esperado que ocorram alguns problemas de precisão numérica. Além disso, o datum usado no KartaView usa um modelo elipsoidal para o planeta, portanto, sabemos que as distâncias medidas usando-se o método esférico pode ter resultados diferentes do resultado obtido com a projeção de Mercator.

Para aplicar a lei esférica dos cossenos nos dados que pegamos pelo KartaView, podemos considerar que o ponto **u** da figura 5 é a origem do sistema, isso é, sua latitude e longitude são zero, e que os pontos **v** e **w** correspondem aos pontos do trajeto cuja distância queremos medir.

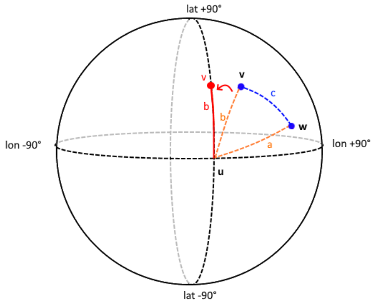
$$C = \sin(\phi_1)\sin(\phi_2) + \cos(\lambda_2 - \lambda_1)\cos(\phi_1)\cos(\phi_2)$$

e consequentemente a distância c considerando uma esfera de raio r será

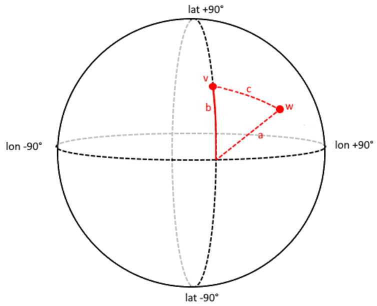
$$c = \frac{r\pi}{180}\arccos(C)$$



(a) Pontos na esfera



(b) Rotação da esfera para alinhar o grande círculo b com um meridiano.



(c) Triângulo esférico na esfera rotacionada.

2.2 Experimentos

Como discutimos em sala de aulas, ter um protocolo experimental é fundamental para o sucesso e para a reprodutibilidade de um experimento. Assim, vamos estabelecer um protocolo para nossos experimentos que consistem em analisar os dados vindos do passeio de dois veículos em uma estrada brasileira e uma estrada no exterior.

1. Escolher um trecho de rodovia no KartaView. O trecho deve ser tal que possa acontecer MRU.
2. Coletar os dados da rodovia com o Postman.
3. Filtrar os dados para ficar apenas com os que interessam para as análises.
4. Realizar as mesmas analises feitas nos exercícios dos labs das aulas 2 e 3.
5. Calcular e comparar as distâncias obtidas pela fórmula de Haversine, trigonometria esférica e a projeção de Mercator.
6. Comparar as distâncias medidas com distâncias calculadas com imagens dos pontos fiduciais.

Para cada subtrecho de velocidade aproximadamente constante, você deve medir:

- A distância percorrida entre cada par de pontos.
- O tempo decorrido entre cada par de pontos
- A velocidade entre cada par de pontos
- A distância total percorrida no subtrecho

- O tempo total decorrido no subtrecho
- A velocidade média do subtrecho

Note que as distâncias devem ser medidas usando-se a projeção de Mercador (i.e. com o pacote `pyproj`), a fórmula de haversine e também com trigonometria esférica, ou seja, esta sequencia de medidas será realizada três vezes.

Para facilitar a visualização dos dados, faça um gráfico da distância percorrida (medida das três formas) no eixo das ordenadas pelo tempo decorrido, no eixo das abscissas, para o trecho completo.

2.3 Entrega

A entrega do EP consistirá no envio (“upload” até 23h55m do dia indicado no E-DISCIPLINAS), via e-disciplinas, de um arquivo zip contendo:

- Relatório no formato definido no ”Appendix 1A, chapter 1, page 9”, do livro do Gould (reproduzido abaixo). O relatório **deve obrigatoriamente ter as seções indicadas nesse apêndice**. Além disso, uma última seção deve ser anexada com a contribuição dos autores, em que deve constar as responsabilidades de cada membro da equipe. Como exemplo, veja a seção Author’s Contributions em: `http://www.biomedcentral.com/1471-2105/16/35`.
- Os códigos fonte.
- Os arquivos json dos dados.

2.4 Reprodução do Apêndice 1A do livro do Gould

Para facilitar sua vida, reproduzimos aqui o apêndice 1A do livro: ”An Introduction to Computer Simulation Methods Applications to Physical System- ”Laboratory reports”

”Laboratory reports should reflect clear writing style and obey proper rules of grammar and correct spelling. Write in a manner that can be understood by another person who has not done the research. In the following, we give a suggested format for your reports.

- **Introduction**. Briefly summarize the nature of the physical system, the basic numerical method or algorithm, and the interesting or relevant questions.
- **Method**. Describe the algorithm and how it is implemented in the program. In some cases this explanation can be given in the program itself. Give a typical listing of your program. Simple modifications of the program can be included in an appendix if necessary. The program should include your name and date and be annotated in a way that is as self-explanatory as possible. Be sure to discuss any important features of your program.

- **Verification of program**. Confirm that your program is not incorrect by considering special cases and by giving at least one comparison to a hand calculation or known result.

- **Data**. Show the results of some typical runs in graphical or tabular form. Additional runs can be included in an appendix. All runs should be labeled, and all tables and figures must be referred to in the body of the text. Each figure and table should have a caption with complete information, for example, the value of the time step.

- **Analysis**. In general, the analysis of your results will include a determination of qualitative and quantitative relationships between variables and an estimation of numerical accuracy.

- **Interpretation**. Summarize your results and explain them in simple physical terms whenever possible. Specific questions that were raised in the assignment should be addressed here. Also give suggestions for future work or possible extensions. It is not necessary to answer every part of each question in the text.

- **Critique**. Summarize the important physical concepts for which you gained a better under-standing and discuss the numerical or computer techniques you learned. Make specific comments on the assignment and suggestions for improvements or alternatives.

- **Log**. Keep a log of the time spent on each assignment and include it with your report.

3 Plágio

Plágio é a copia/modificação não autorizada e/ou sem o conhecimento do autor original. O plágio é um problema grave que pode levar até a expulsão do aluno da universidade. Leia o Código de Ética da USP (em particular, a seção V): `http://www.mp.usp.br/sites/default/files/arquivosanexos/codigo_de_etica_da_usp.pdf`.