

MCTA017-17 Programação Matemática -- 2022-Q1 -- Prof. Jerônimo

[Painel](#) / [Meus cursos](#) / [Programação Matemática - prof Jreonimo 2022q1](#)

Avisos

Onde estamos - que material usar

Não disponível

Lista de Exercícios (para avaliação)

Não disponível

Ementa

Introdução: Revisões de álgebra linear e conjuntos convexos. Programação linear: Modelagem; Resolução Gráfica; Teoremas Básicos; O método simplex; Simplex revisado; Dualidade; Algoritmos primal-dual e dual-simplex; Análise de sensibilidade. Programação Dinâmica

Requisitos

Álgebra Linear, Funções de uma Variável.

Geometria Analítica. (GA já é recomendação para AL, mas não custa relembrar)

Objetivo

- Desenvolver a capacidade de formular, estruturar e solucionar modelos matemáticos;
- Compreender os fundamentos dos métodos mais básicos de otimização linear;
- Dar continuidade ao processo de desenvolvimento de raciocínio rigoroso e abstrato, completando uma cadeia de disciplinas teóricas. Esse desenvolvimento tem efeitos de longo prazo e não necessariamente ligados diretamente aos tópicos estudados.

Dinâmica

Teremos sessões síncronas nos horários estabelecidos para a disciplina, e um grupo Telegram para dúvidas.

- As aulas síncronas servirão como guia para estudo das notas de aula. É recomendado ler o material indicado antes da aula, e tirar as dúvidas no encontro síncrono.
- As avaliações incluirão alguns exercícios de modelagem, mas haverá principalmente exercícios teóricos (perguntas sobre os algoritmos, demonstrações de propriedades etc).

Horário das sessões síncronas:

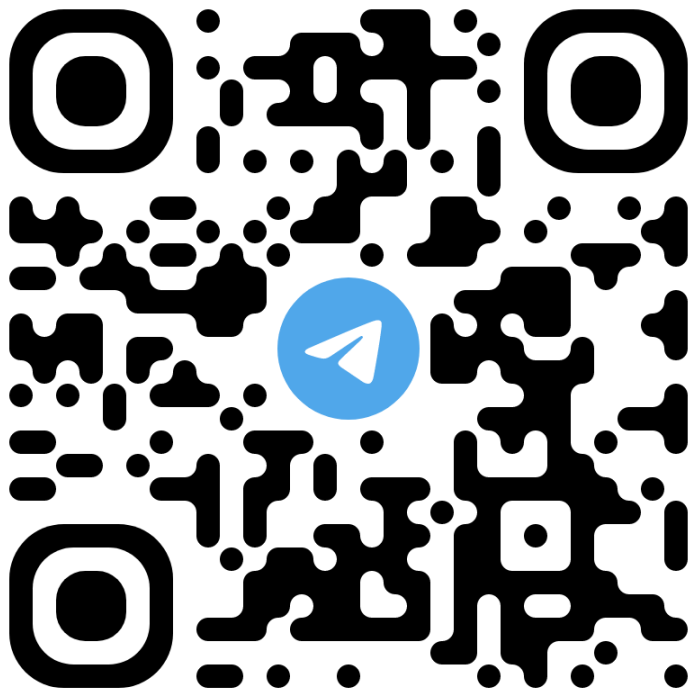
- TERÇA 19:00
- SEXTA 21:00

Sessão de atendimento:

- TERÇA 21:00

- Aulas síncronas <https://conferenciaweb.rnp.br/webconf/jeronimo-6>

- Chave de acesso para aulas síncronas: [7a3b7742](#)
- Grupo no Telegram: <https://t.me/+xMYCZAuuR20yNTMx>



Avaliação

O conceito final da disciplina poderá ser:

- F - Reprovado. O aluno deve cursar novamente a disciplina.
- C - Desempenho mínimo satisfatório, demonstrando capacidade de uso adequado dos conceitos da disciplina, habilidade para enfrentar problemas relativamente simples e prosseguir em estudos avançados.
- B - Bom desempenho, demonstrando boa capacidade de uso dos conceitos da disciplina.
- A - Desempenho excepcional, demonstrando excelente compreensão da disciplina e do uso da matéria.

A avaliação consistirá de:

- Exercícios, L, valendo 0, 1 ou 2. Os exercícios serão indicados ao longo das aulas -- deixarei a indicação será feito AQUI NO MOODLE.
- Duas atividades, A_1 e A_2 , valendo entre 0, 1, 2, 3 ou 4 cada.

As notas serão convertidas em conceito de acordo com a seguinte regra.

$$n = A_1 + A_2 + L$$

O conceito final será:

- $n \in [0, 5) \rightarrow F$
- $n \in [5, 7) \rightarrow C$
- $n \in [7, 8.75) \rightarrow B$
- $n \in [8.75, 10] \rightarrow A$

Quem ficar com F pode fazer o exame; a nota final será $0.6n + 0.4e$, onde n é a nota dos testes e e é a nota do exame.

Programa

Este programa está sujeito a mudanças simples. Grandes mudanças não devem acontecer.

- Programação Linear: modelagem e resolução gráfica
- Análise de algoritmos (noções rudimentares)

- Conjuntos convexos
- O método simplex
- Dualidade
- Algoritmos primal-dual e dual-simplex
- Análise de sensibilidade

Bibliografia

Os livros disponíveis na biblioteca tem sua identificação entre colchetes -- por exemplo, [[519 / PAPAco](#)]. Os que não existem na biblioteca tem o ISBN entre parênteses: ([ISBN-13: 978-8131203767](#)).

Principal

- [Notas de aula](#) -- principal texto a ser usado na disciplina.
- **Matousek, Jiri; Gärtner, Bernd.** *Understanding and Using Linear Programming*. Springer, 2007. Mantém o foco somente em programação linear, mas dentro do que faz é extremamente didático. ([ISBN-13: 978-3-540-30697-9](#))
- **Sinha, S. M.** *Mathematical Programming: Theory and Methods* Elsevier, 2006. ([ISBN-13: 978-8131203767](#)). Não tem na biblioteca. Inclui uma breve revisão de Álgebra Linear. Tem vários pequenos erros.
- **Luenberger, David G.; Ye, Yinyu.** *Linear and Nonlinear Programming* Springer. Springer, 2008. ([ISBN-13: 978-0-387-74502-2](#)). [Disponível pela Springer](<http://www.springer.com/mathematics/applications/book/978-0-387-74502-2>), dentro da UFABC.
- [Notas de aula de Criptografia](#): o apêndice sobre Análise de Algoritmos será usado somente para a aula sobre Análise de Algoritmos
- [Notas de aula de Álgebra Linear](#): o Capítulo sobre formas quadráticas aborda o tópico de convexidade de funções. Além disso, pode ser útil para revisar ou relembrar conceitos de Álgebra Linear usados neste curso.
- **Dasgupta, S.; Papadimitriou, C.; Vazirani, U.** *Algoritmos* McGraw-Hill, 2009 [[518.1 / DASa](#)], também em Inglês [[518.1 / DASa](#)] -- há um capítulo muito resumido sobre programação linear; mas o livro pode ser útil para entender complexidade de algoritmos.

Secundária

- **Glenn Hurlbert.** *Linear Optimization: The Simplex Workbook* Springer, 2009 ([ISBN-13: 978-0387791470](#)) Muito bom livro! O texto é intercalado com muitos exercícios.
- **Papadimitriou, Christos H.; Steiglitz, Kenneth.** *Combinatorial Optimization: algorithms and complexity*. Dover, 1988. [[519 / PAPAco](#)] A [errata está na página de Ken Steiglitz](<http://www.cs.princeton.edu/~ken/PUBS.html>)
- **Griva, I, Nash, S. G., Sofer, A.** *Linear and Nonlinear Optimization* Society for Industrial Mathematics, 2008. ([ISBN-13: 978-0898716610](#)) Muito bom livro, extenso.
- **Sierksma, G.** *Linear & Integer Programming: Theory and Practice* CRC Press, 2001. ([ISBN-13: 978-0824706739](#))
- **Stancu-Minasian, I. M.,** *Fractional Programming: theory, methods and applications*. Kluwer, 1997. ([ISBN-13: 978-94-010-6504-7](#))
- **Bajalinov, Erik,** *Linear-Fractional Programming: theory, methods, applications and software*. Springer, 2003. ([ISBN-13: 978-1-4613-4822-1](#))
- **Strayer, J. K.** *Linear Programming and Its Applications* Springer, 1989. ([ISBN-13: 978-0387969305](#))
- **Vanderbei, Robert J.** *Linear Programming: foundations and extensions* Kluwer Academic Publishers, 2007. Há uma errata [aqui](<http://www.princeton.edu/~rvdb/LPbook/index.html>).
- **Goldberg M.C., Luna H.P.L.,** "Otimização combinatória e programação linear- modelos e algoritmos". Campus, RJ, 2000 [[511.8 / GOLo2 / 2ed](#)] Não usarei, mas pode ajudar.
- **Bazaraa, M. S., Jarvis, J. J. e Sherali, H. D.** *Linear Programming and Network Flows*. New York: John Wiley & Sons, 1990. [[519.72 / BAZI4 / 4ª. ed](#)] Muito bom.
- **Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.,** *Algoritmos - Teoria e Prática*. Campus, 2002. [[005.1 / CORi2](#)] [[005.1 / CORa](#)] Aborda o assunto somente tangencialmente.
- **Bertimas, Dimitris; Tsitsiklis, John N.** *Introduction to linear Optimization*. Nashua: Athena Scientific, 1997. [[519.7 / BERi](#)] Bom livro!
- **Schrijver, A.** *Theory of Linear and Integer Programming* Wiley, 1999. ([ISBN-13: 978-0471982326](#)) Ótimo livro, para quem quiser mergulhar na teoria de otimização linear.

Software

Não será necessário para o desenvolvimento da disciplina, mas pode interessar.

Há uma enorme quantidade de programas que resolvem programação linear. Alguns exemplos:

- [Maxima](#) é um concorrente livre do Mathematica. O prof. Daniel Miranda e eu fizemos um [tutorial muito básico do Maxima](#); veja também [parte do manual que mostra como usar o simplex](#)
- [GLPK](#) (GNU Linear Programming Kit)
 - No Linux, instale o pacote "glpk".
 - Há um [pacote para Windows](#)
- [Scilab](#) -- o Scilab resolve programação linear usando o algoritmo de Karmarkar (veja a [documentação](#))
- [lp_solve](#)
 - No Linux, instale o pacote "lp-solve"
 - Há versão Windows no Sourceforge
- [GNU Octave](#) é um concorrente livre do Matlab. Aqui está a [página do manual](#) que trata de otimização
- O [LibreOffice](#) tem um solver semelhante ao do seu concorrente proprietário, Excel
- [Simplexme](#), resolvidor online que mostra passos intermediários

Exemplos de uso:

GLPK:

```
$ cat exemplo.lp
maximize x1 + 4x2
subject to
x1 + x3 = 9
x2 + 2x4 = 3
end
```

```
$ glpsol --lp exemplo.lp
```

```
GLPSOL--GLPK LP/MIP Solver 5.0
Parameter(s) specified in the command line:
  --lp exemplo.lp
Reading problem data from 'exemplo.lp'...
2 rows, 4 columns, 4 non-zeros
5 lines were read
GLPK Simplex Optimizer 5.0
2 rows, 4 columns, 4 non-zeros
Preprocessing...
~   0: obj =  2.100000000e+01 infeas =  0.000e+00
OPTIMAL SOLUTION FOUND BY LP PREPROCESSOR
Time used:  0.0 secs
Memory used: 0.0 Mb (39693 bytes)
```

Maxima

Carregamos o pacote "simplex":

```
(%i1) load("simplex")$
```

Definimos a função objetivo e a lista de restrições:

```
(%i2) obj: x1+4*x2$
```

```
(%i3) res: [x1 + x3 = 9, x2 + 2*x4 = 3]$
```

Usamos "nonnegative_lp=true", de outra forma o Maxima permitirá que as variáveis sejam negativas:

```
(%i4) maximize_lp(obj,res),nonnegative_lp=true;
```

```
(%o4) [21, [x4 = 0, x2 = 3, x3 = 0, x1 = 9]]
```

Ali estão o valor objetivo (21) e os valores das variáveis para a solução ótima.

[lp_solve](#):

```
$ cat exemplo.lpsolve
```

```
max: x1 + 4x2;
```

```
x1 + x3 = 9;
```

```
x2 + 2x4 = 3;
```

```
$ lp_solve exemplo.lpsolve
```

```
Value of objective function: 21.00000000
```

```
Actual values of the variables:
```

```
x1                9
```

```
x2                3
```

```
x3                0
```

```
x4                0
```



Este é o Ambiente Virtual de Aprendizagem da UFABC para apoio ao ensino presencial e semipresencial. Esta plataforma permite que os usuários (educadores/alunos) possam criar cursos, gerenciá-los e participar de maneira colaborativa.

Informação

[Conheça a UFABC](#)

[Conheça o NTI](#)

[Conheça o Netel](#)

Contato

Av. dos Estados, 5001. Bairro Bangu - Santo André /SP – Brasil. CEP 09210-580.

Siga-nos



Universidade Federal do ABC - Moodle (2020)

[Português - Brasil \(pt_br\)](#)

[English \(en\)](#)

[Português - Brasil \(pt_br\)](#)

[Obter o aplicativo para dispositivos móveis](#)