TÉCNICO LISBOA

Máxima flexibilidade em simulação: programação de modelos discretos de incremento fixo

Francisco Regateiro e Fernando Durão

IST / UL – Departamento de Engenharia Civil, Arquitectura e Georrecursos

CNaPPES.16

3º Congresso Nacional de Práticas Pedagógicas no Ensino Superior

Índice

TÉCNICO LISBOA

Introdução

- Explorar a sinergia no ensino/aprendizagem entre Investigação Operacional (IO) e Introdução à Programação.
- Reconhecer a programação informática como uma ferramenta essencial para a resolução competente de problemas em IO.
- Experimentar com MATLAB e um problema de IO: “Uma empresa que utiliza camiões próprios e alugados para satisfazer pedidos”.

Introdução

- A simulação em IO abarca um conjunto de técnicas baseadas na utilização de um programa de computador com o objetivo de replicar o funcionamento de um sistema produtivo ao longo de um período de tempo determinado e para diferentes possibilidades de valor para uma ou mais variáveis de decisão.

- Modelos discretos de incremento fixo.

Oliveira, R. (2016) “Modelos de Simulação: sete questões fundamentais” em “A investigação operacional em Portugal - Novos desafios, novas Ideias”, 978-989-8481-49-8, Editora IST Press (pp. 451-478).

Modelação

- Constantes

  M – Penalização por dia de atraso

  CP – Custo fixo por camião próprio

  CV – Custo utilização de camião próprio

  CA – Custo aluguer de um camião

- Parâmetros da simulação

  K – Nº de dias de simulação

  FP – Nº de camiões da frota própria (VD)

Modelação

- Variáveis de estado
  PF – Nº de pedidos que ficam pendentes
  DR – Nº de dias com pedidos pendentes
- Variáveis exógenas estocásticas
  PN – Nº de pedidos novos
  FA – Nº de camiões disponíveis para alugar

Modelação

- Variáveis internas estocásticas
  - PS – Nº de pedidos a satisfazer
  - FD – Nº total de camiões disponíveis
- Variáveis manipuláveis
  - FPU – Nº de camiões próprios utilizados
  - FAU – Nº de camiões de aluguer utilizados
- Variáveis de saída estocásticas
  - CG – Custo global em euros

Inicializar a solução / sistema

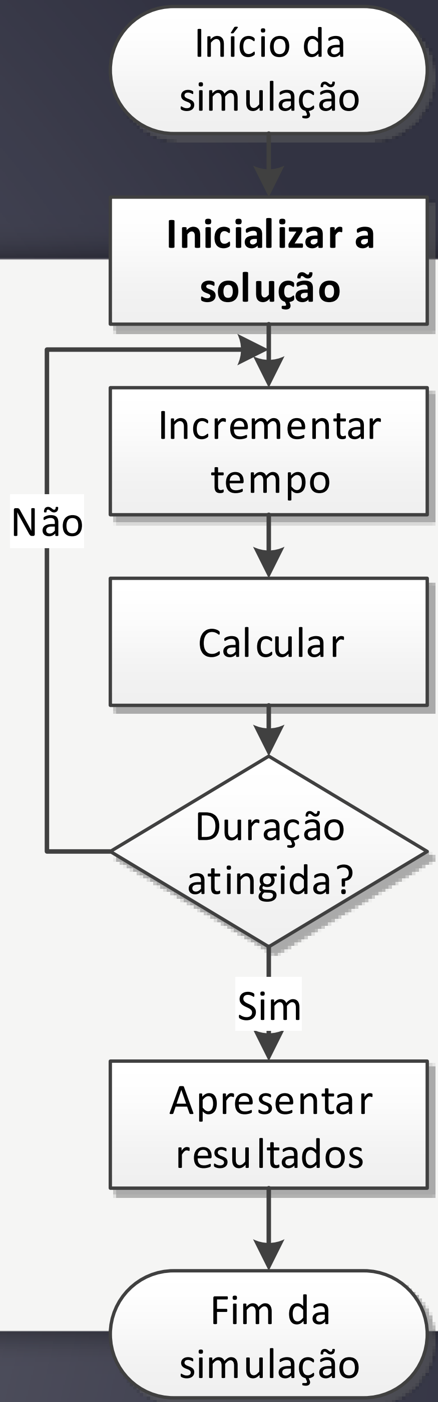
FP = ?

K = ?

i = 0

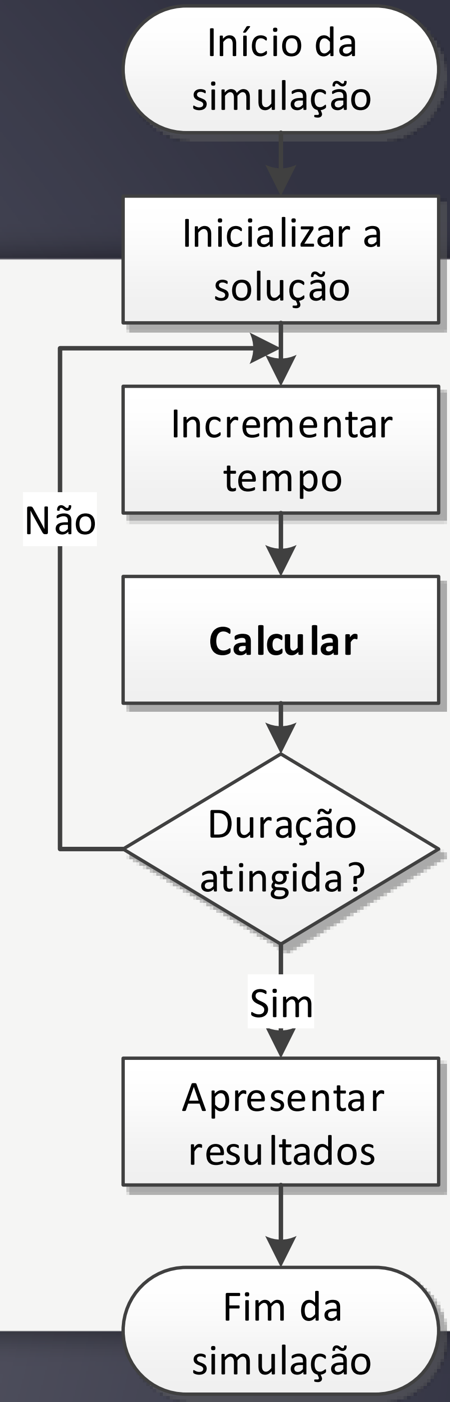
PF(0) = 0

DR = 0



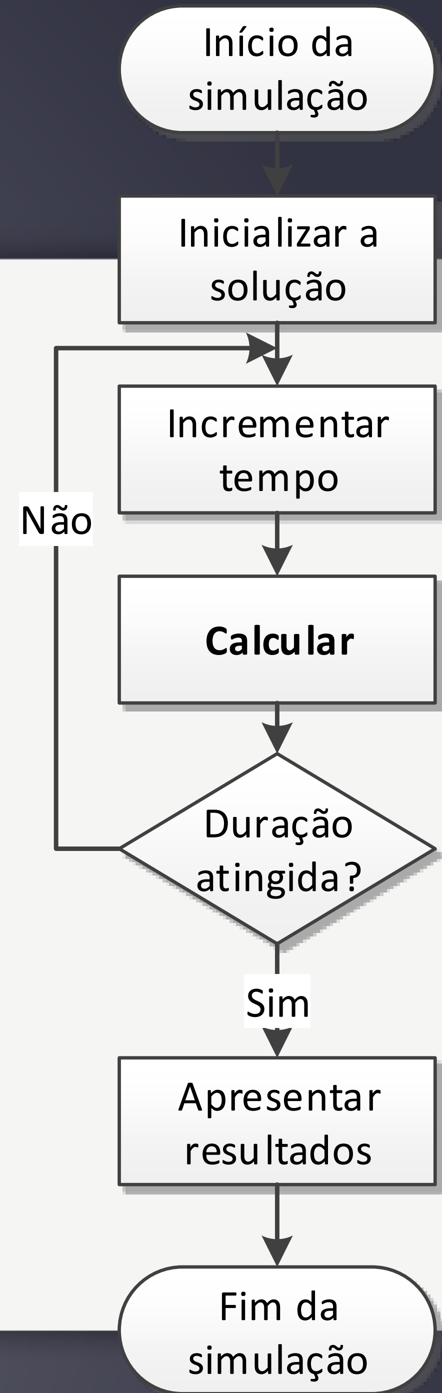
Calcular

1. Contabilizar o nº de pedidos que chegam no dia *i* e acumular com pedidos do dia anterior.
2. Verificar se frota própria é suficiente e, em caso negativo, verificar quantos camiões de aluguer estão disponíveis.
3. Determinar o nº de pedidos que ficam para o dia seguinte e calcular os custos de funcionamento no dia.



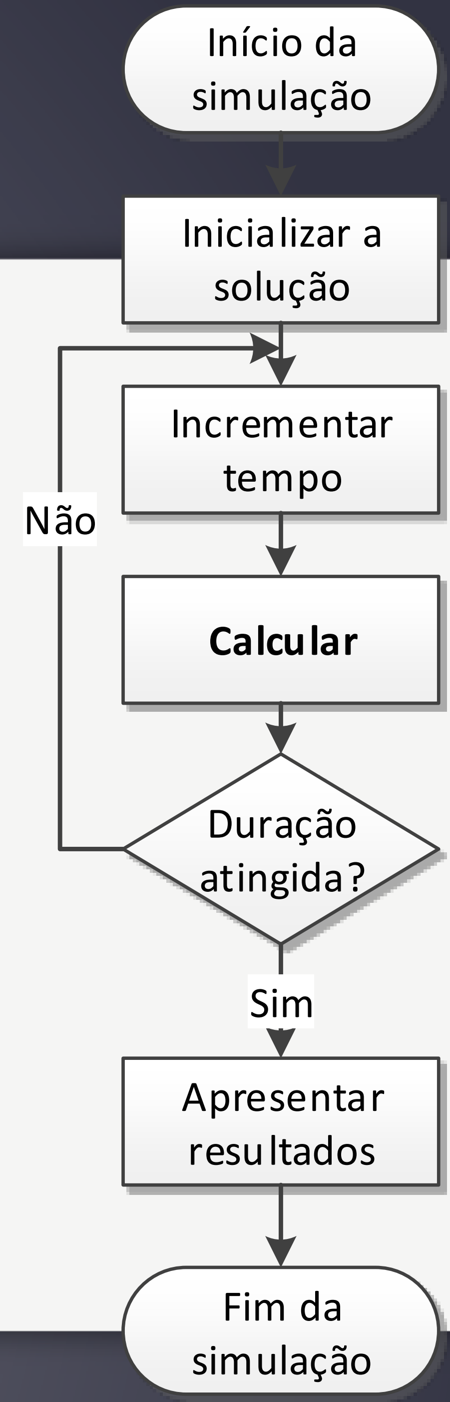
Gerar nº de pedidos novos

PN	P(PN = pn)	F(pn)
30	1/21	1/21
31	1/21	2/21
32	1/21	3/21
…	…	…
48	1/21	19/21
49	1/21	20/21
50	1/21	1



Gerar nº de camiões de aluguer

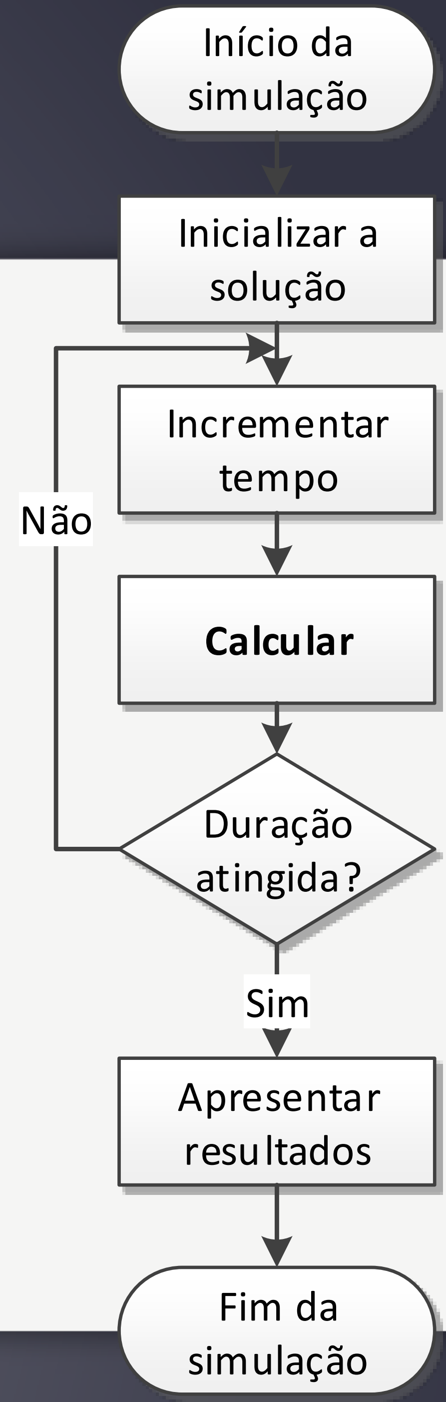
FA	P(FA = fa)	F(fa)
5	0.35	0.35
6	0.25	0.60
7	0.20	0.80
8	0.15	0.95
9	0.05	1



Codificação

1.

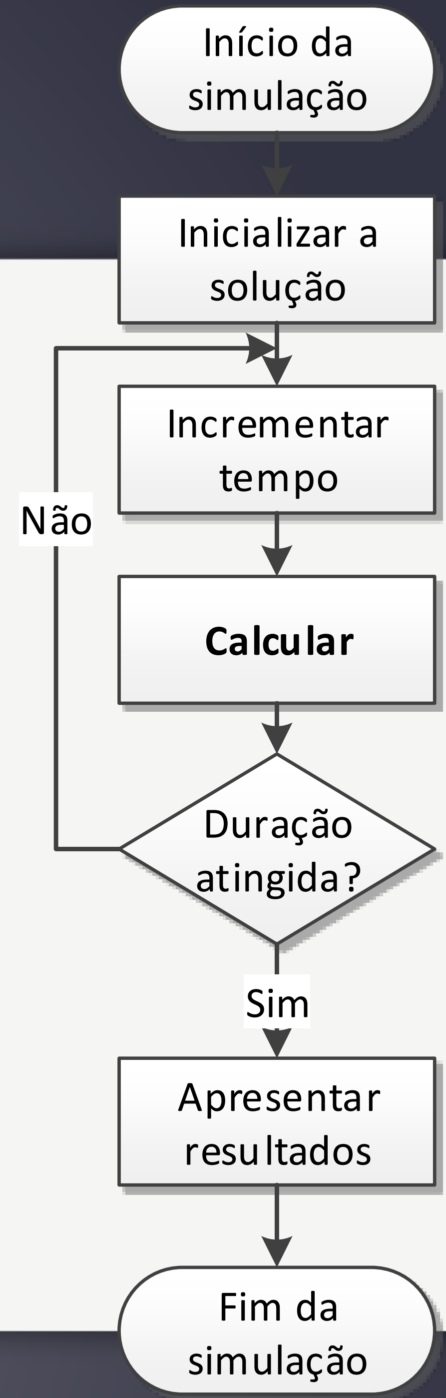
```
PN(i) = gerarPedidosNovos;
PS(i) = PF(i-1) + PN(i);
```



Codificação

2.

```
se PS(i) <= FP
  PF(i) = 0;
  FPU(i) = PS(i);
  FAU(i) = 0;
caso contrário
  FPU(i) = FP;
  FA(i) = gerarFrotaAluguer;
```

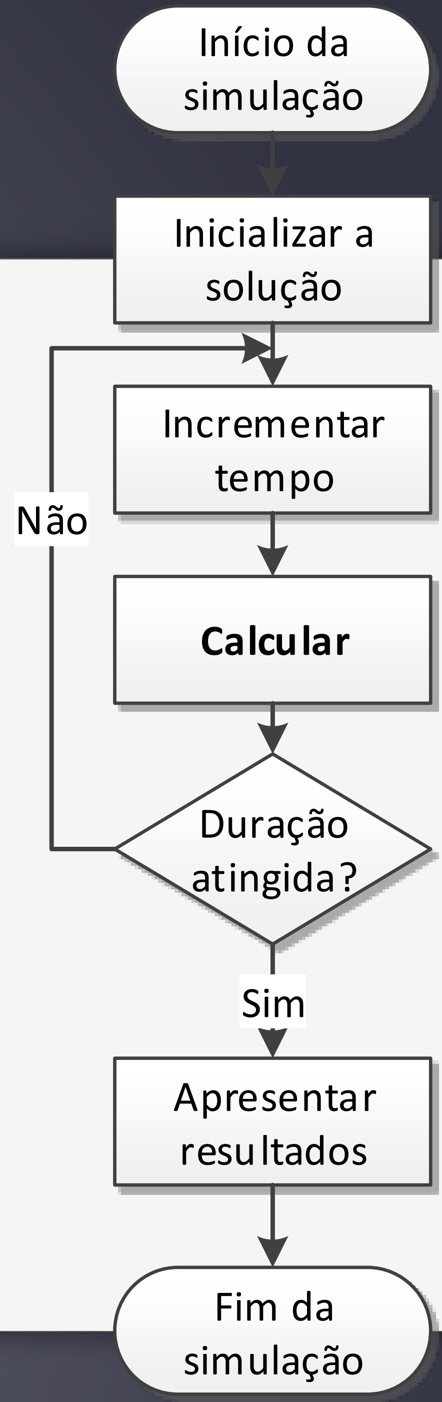


Codificação

3.

```
se PS(i) <= FP + FA(i)
  PF(i) = 0;
  FAU(i) = PS(i) - FP;
caso contrário
  PF(i) = PS(i) - FP + FA(i);
  DR = DR + 1;
  FAU(i) = FA(i);

CG(i) = CP * FP + CV * FPU(i) + ...
        CA * FAU(i) +  M * PF(i);
```



Máxima Flexibilidade

1. Utilizar quaisquer valores em parâmetros da simulação.

```
K =  365 * 100;
```

2. Automatizar a simulação. Por exemplo, utilizando um ciclo exterior para testar diferentes possibilidades de valor para uma variável de decisão.

```
hipotesesFP = 35:53;
quantidadeHipotesesFP = length(hipotesesFP);
CG = zeros(K, quantidadeHipotesesFP);
```

Máxima Flexibilidade

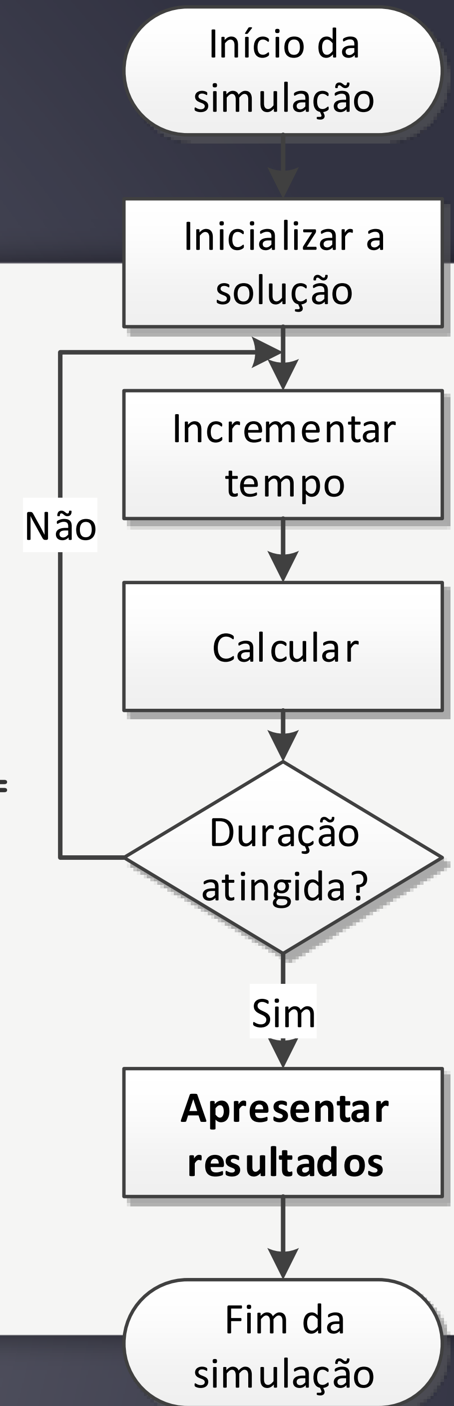
3. **Produzir resultados.**

   – Valores de FP que minimizam CG

```
hipotesesFP(min(sum(CG))==
sum(CG))

ans =

    44
```



Máxima Flexibilidade

– Custo médio

$\sum CGi / K$

```
plot(hipotesesFP, mean(CG))
```

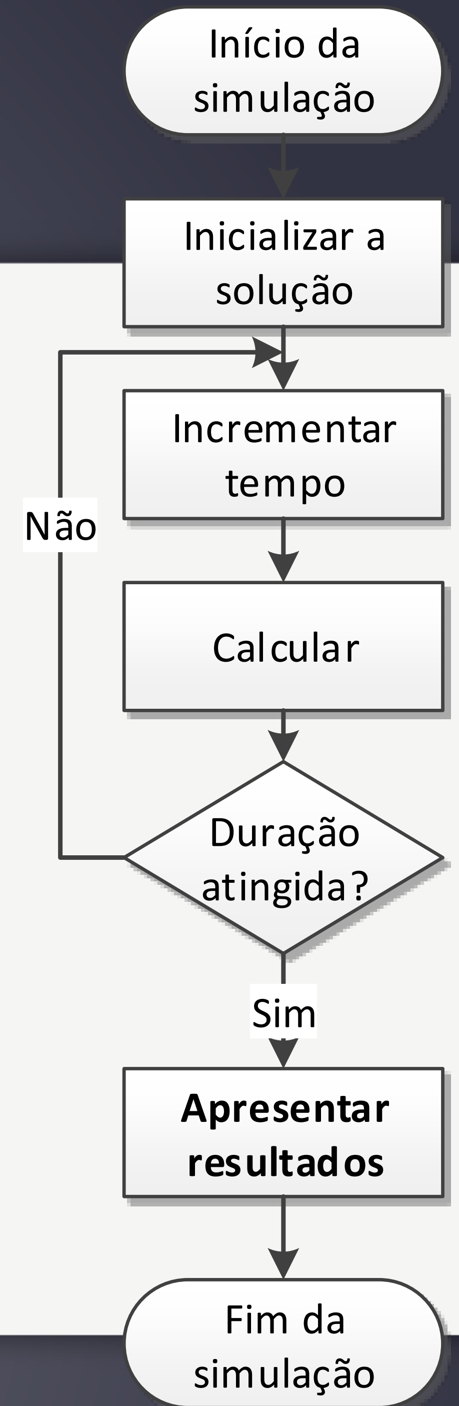
– Nº médio de pendentes

$\sum PFi / K$

```
plot(hipotesesFP, mean(PF))
```

– Nº máximo de pedidos pendentes

```
plot(hipotesesFP, max(PF))
```



Máxima Flexibilidade

– Taxa de utilização da FP

∑FPUi / (K.FP)

```
plot(hipotesesFP, sum(FPU)./
(K*hipotesesFP))
```

– Média de camiões alugados

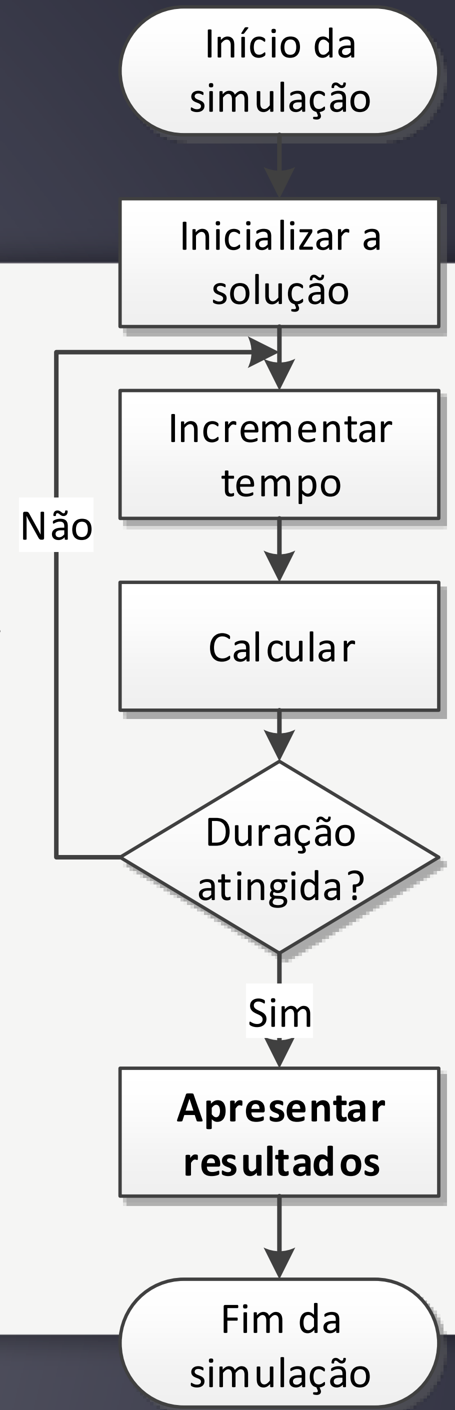
∑FAUi / K

```
plot(hipotesesFP, mean(FAU))
```

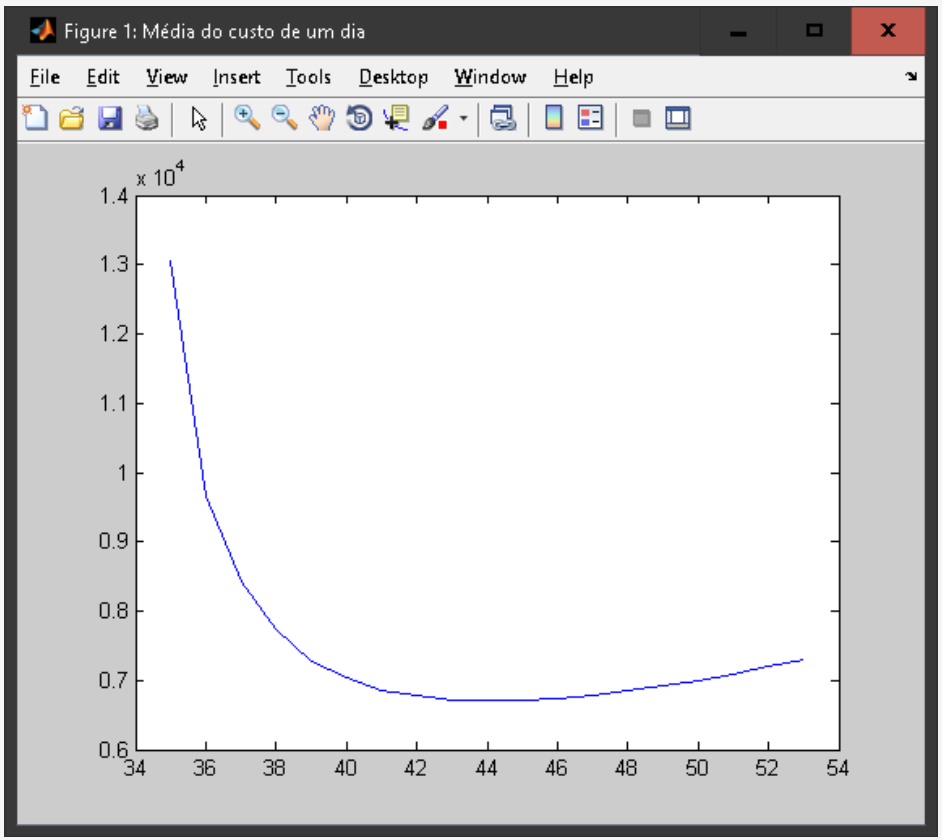
– Risco de rotura

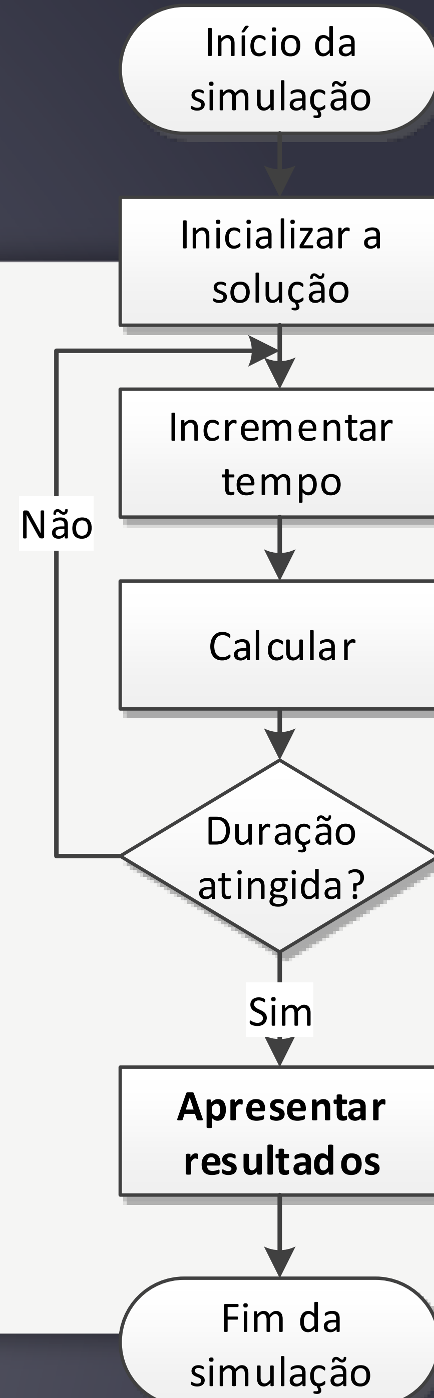
DR / K

```
plot(hipotesesFP, DR/K)
```

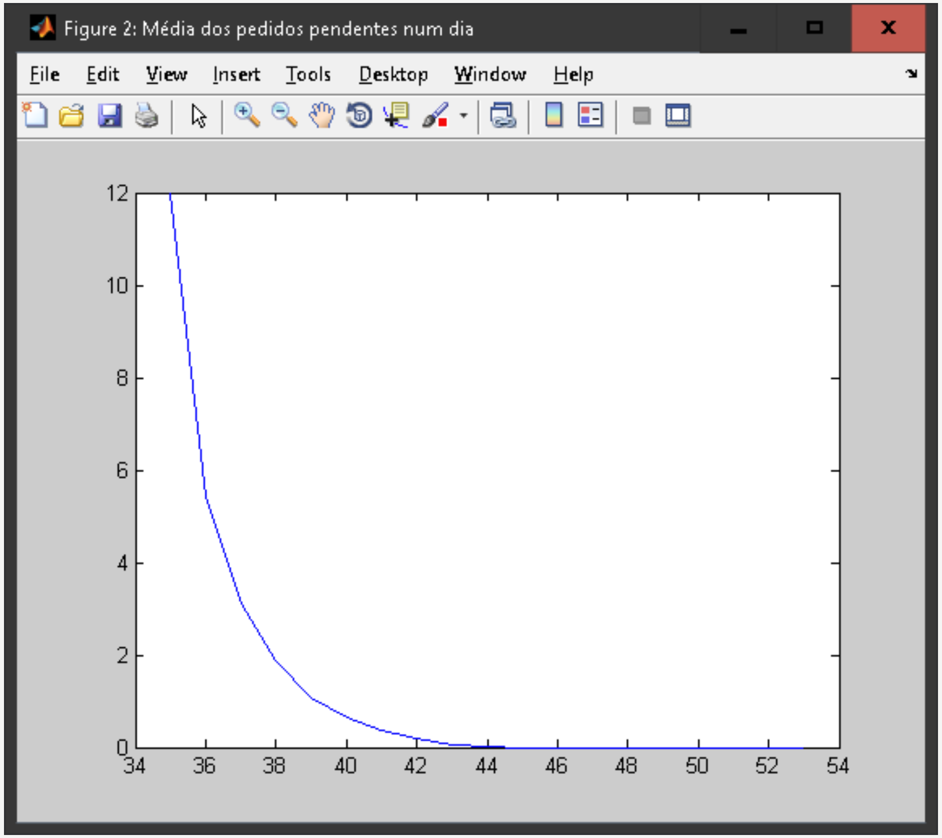


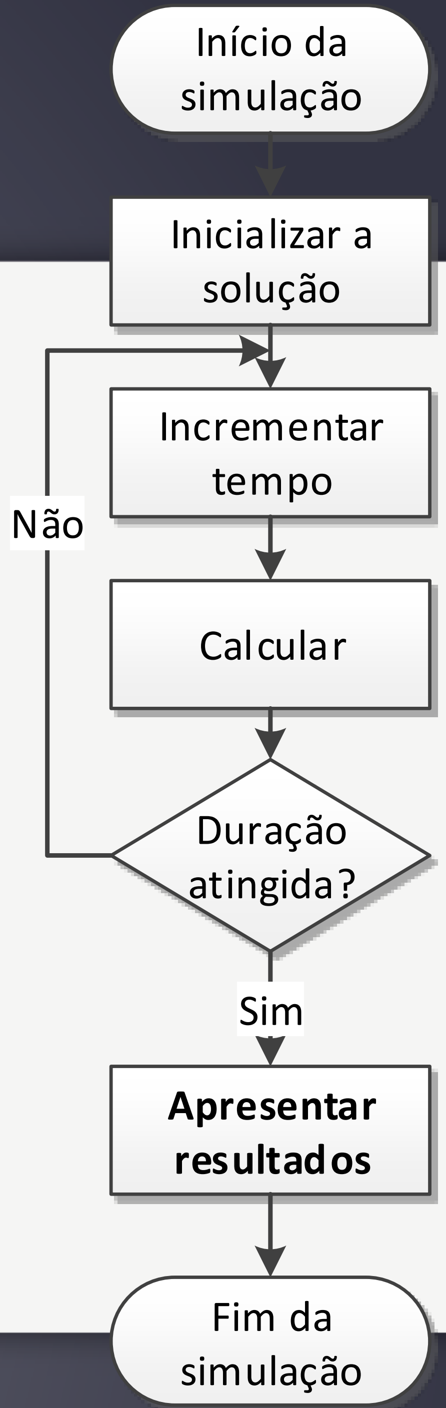
Máxima Flexibilidade



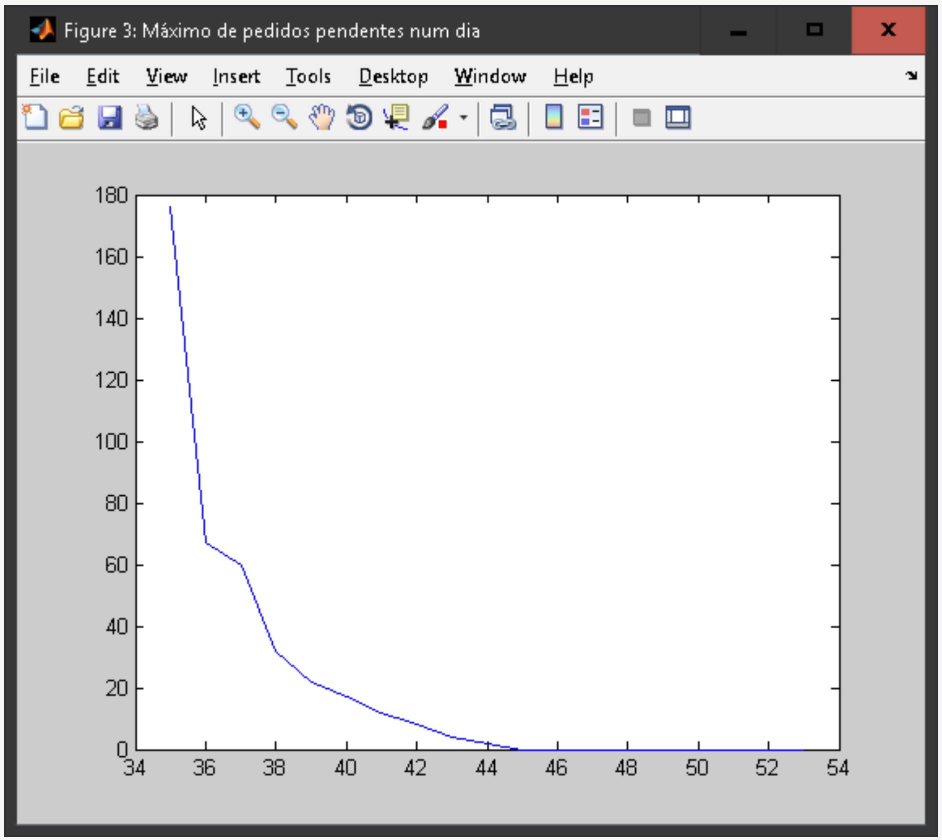


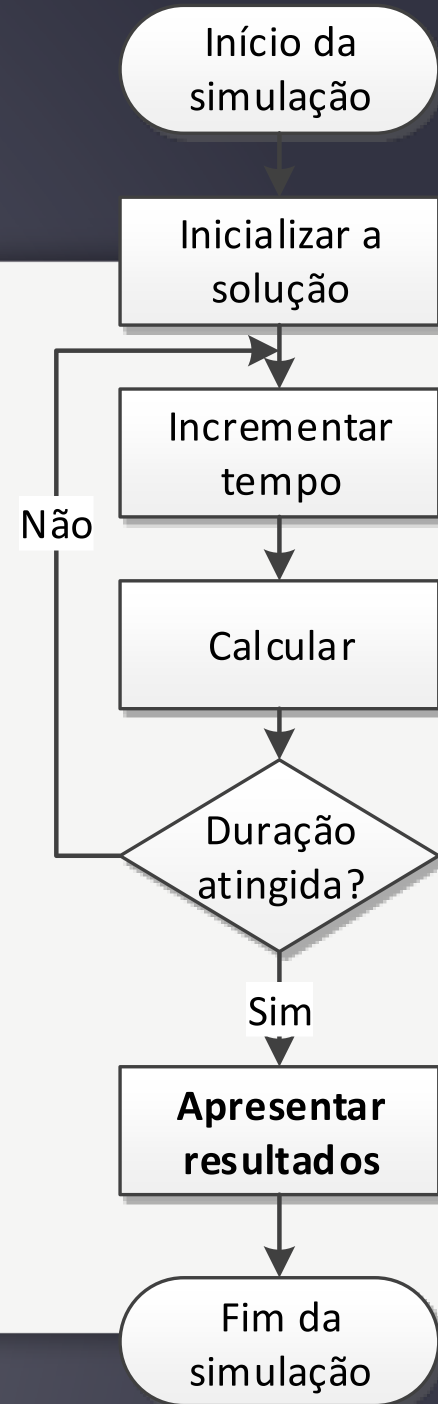
Máxima Flexibilidade



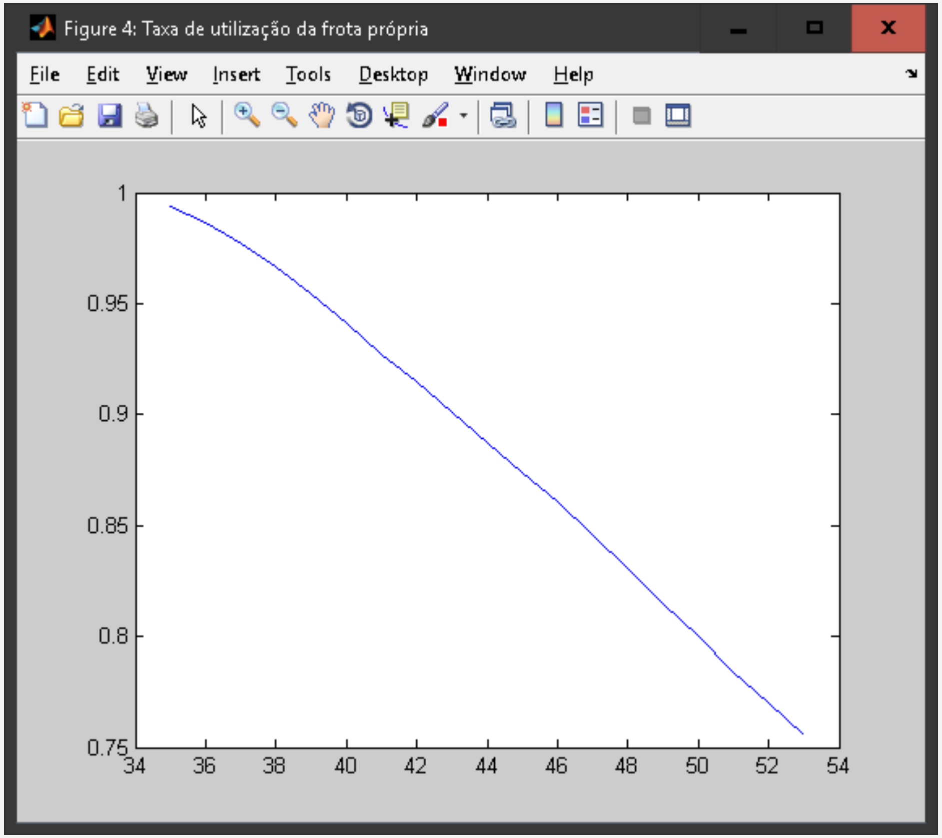


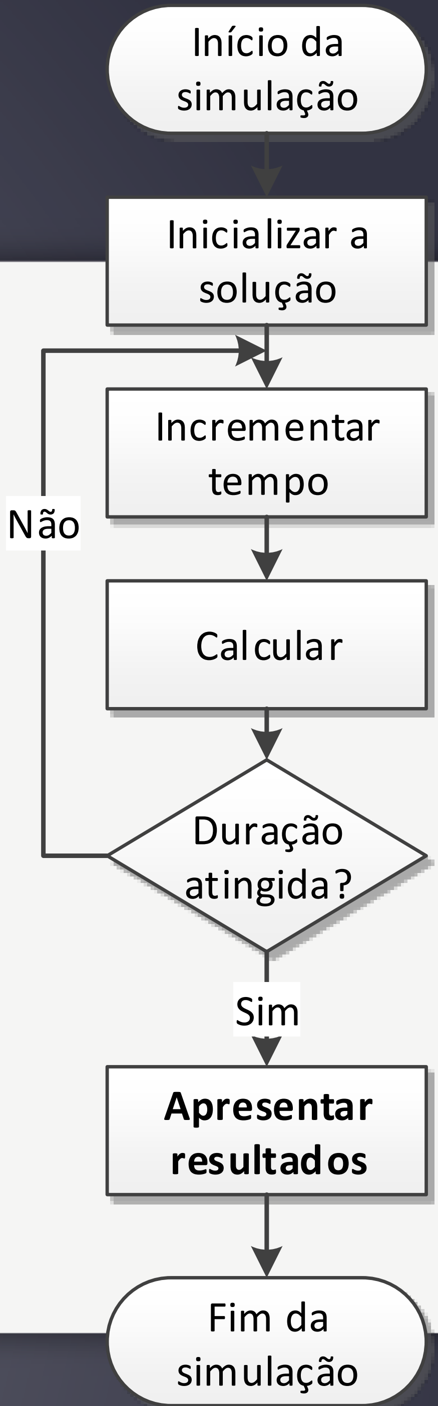
Máxima Flexibilidade



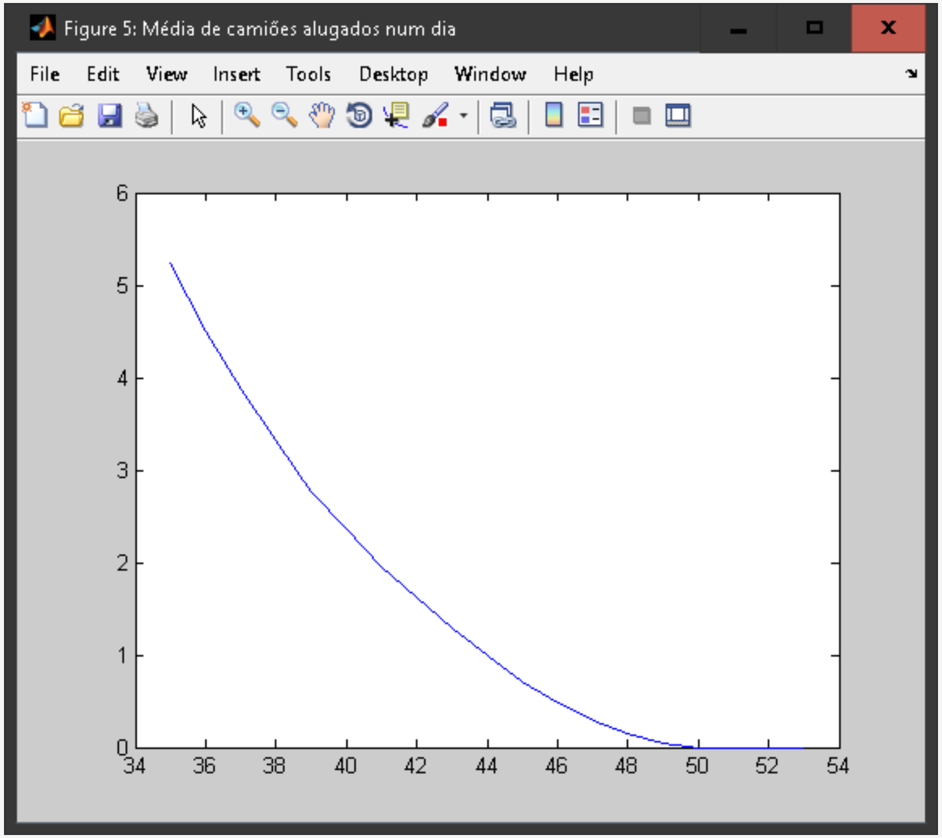


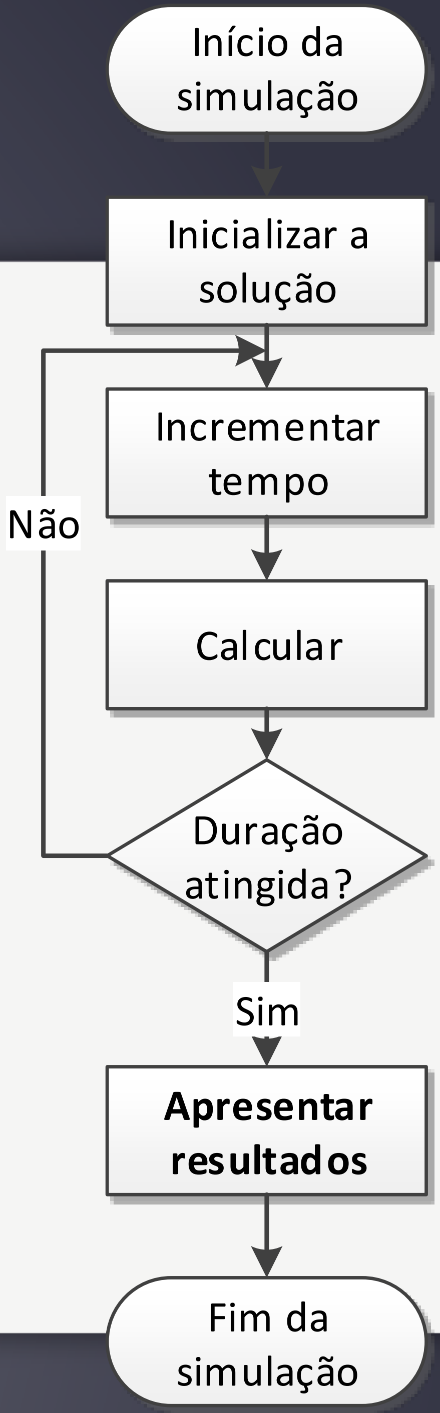
Máxima Flexibilidade



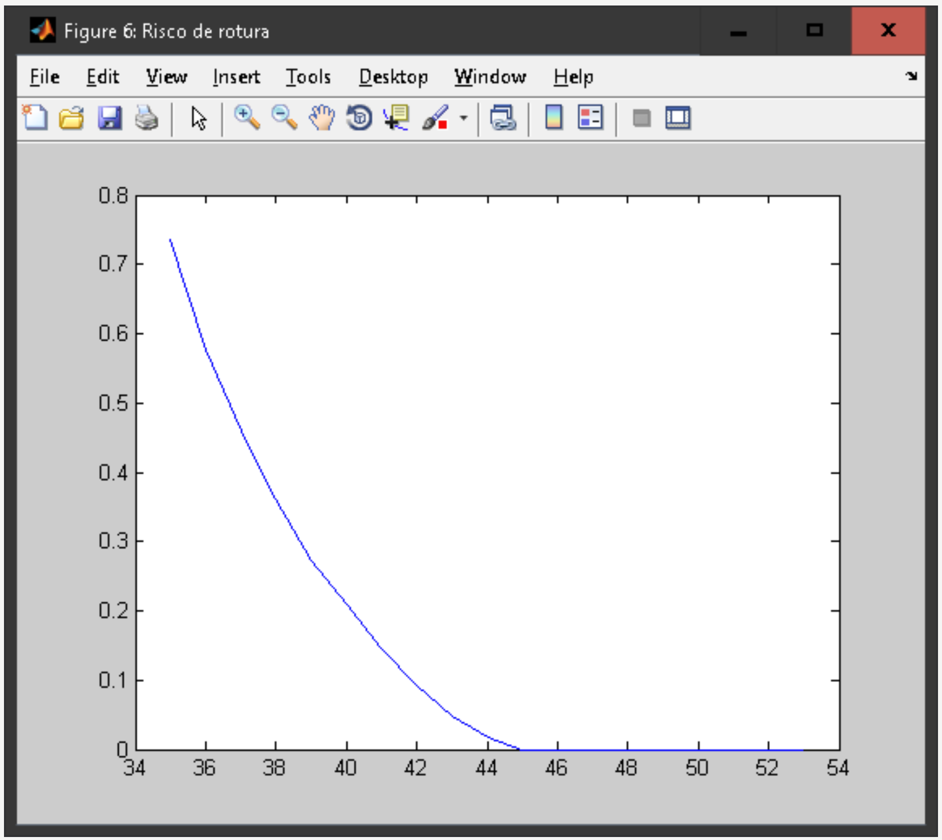


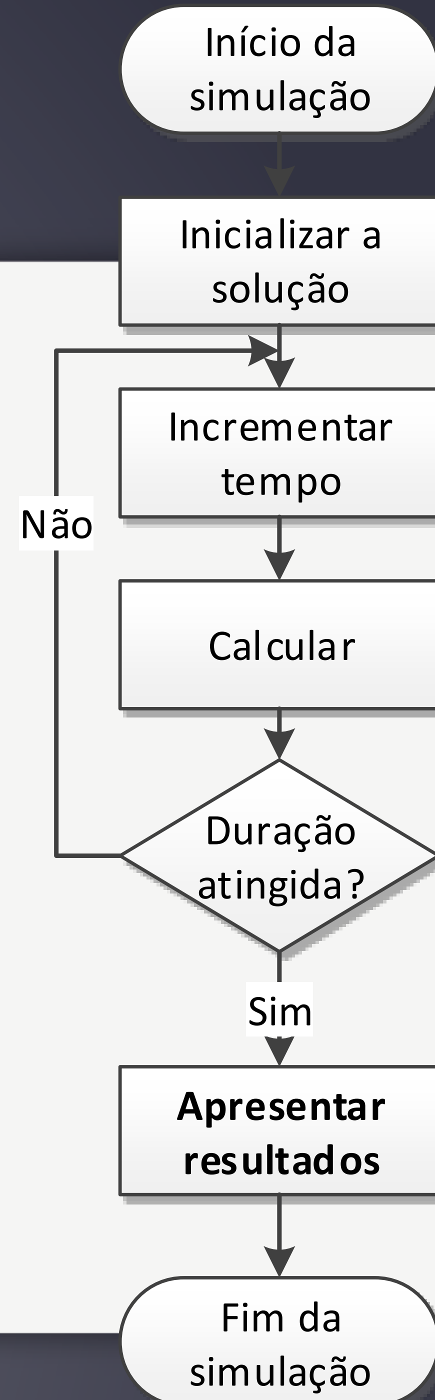
Máxima Flexibilidade





Máxima Flexibilidade





Conclusão

- Resolução robusta do problema (e reutilizável).
- Apenas necessita de nível básico de programação informática.
- Permite aplicar e desenvolver a competência de programação informática.
- Retorno positivo dos alunos.

Obrigado pela atenção!

(francisco.regateiro@tecnico.ulisboa.pt)