



Universidade Estadual de Santa Cruz – UESC

Relatório de Implementações dos Métodos da Disciplina Análise Numérica

**Relatório de implementações
realizadas por Igor Alex Farias Dos
Santos De Almeida**

**Disciplina Analise Numérica.
Curso Ciência da Computação
Semestre 2023.1**

**Professor Gesil Sampaio
Amarante II**

Ilhéus – BA2022

ÍNDICE

- 1- Método de Euler
- 2- Método de Heun
- 3- Método Runge-Kutta de 4ª Ordem
- 4- Método de Ralston
- 5- Método de “Shooting”
- 6- Considerações Finais

Linguagem(ns) Escolhida(s) e justificativas

Optei pela Linguagem Python como nossa escolha de programação devido à sua facilidade de aprendizado. Além disso, ela gera um código altamente legível e possui uma vasta quantidade de bibliotecas disponíveis para diversas soluções.

Método de Euler

Estratégia de Implementação:

Durante a leitura do arquivo, o intervalo é armazenado em uma lista, enquanto as demais entradas são atribuídas a variáveis separadas. O cálculo é realizado para cada ponto no intervalo, com espaçamento determinado pelo valor de "h" informado.

Os cálculos dos valores nos pontos são realizados em três métodos no algoritmo. Primeiramente, o método "passos()" determina quantos pontos serão calculados com base no intervalo e no valor de "h". Em seguida, o método "calc2()" recebe a função e duas variáveis como entrada, retornando o valor da função nesses pontos. Para isso, a função "eval()" é utilizada, portanto, todas as funções devem ser escritas de acordo com a sintaxe do Python, de modo a serem interpretadas corretamente.

Por fim, o método "euler()" contém o cálculo do método em si e retorna os pontos arredondados em três casas decimais, utilizando a função "round()".

Estrutura dos Arquivos de Entrada/Saída:

No arquivo de entrada, são fornecidas as seguintes informações, separadas por quebras de linha ('\n'): o intervalo, o ponto inicial 'y0', a função e o valor de 'h'. Cada informação é inserida em uma linha separada, e cada conjunto de informações é separado por uma quebra de linha ('\n').

No arquivo de saída, os intervalos de pontos são apresentados indicando sua ordem em forma de lista. Cada caso é separado por uma quebra de linha ('\n').

Problemas teste:

O primeiro problema a ser testado está na página 423 do livro "Cálculo Numérico" de Neide Franco, na questão de número 12.3.

A entrada para este problema foi :

0

50

0

$(200-2y)/(200-x)^1$

E o resultado encontrado foi:

['0: 0.9', '1: 1.8', '2: 2.7', '3: 3.6', '4: 4.5', '5: 5.4', '6: 6.3', '7: 7.2', '8: 8.1', '9: 9.0',
'10: 9.9', '11: 10.8', '12: 10.743', '13: 10.685', '14: 10.628', '15: 10.57', '16:
10.513', '17: 10.455', '18: 10.398', '19: 10.34', '20: 10.283', '21: 10.226', '22:
10.168', '23: 10.111', '24: 10.053', '25: 9.996', '26: 10.973', '27: 10.909', '28:
10.846', '29: 10.783', '30: 10.719', '31: 10.656', '32: 10.592', '33: 10.529', '34:
10.465', '35: 10.402', '36: 10.339', '37: 10.275', '38: 10.212', '39: 10.148', '40:
10.085', '41: 10.021', '42: 9.958', '43: 11.041', '44: 10.97', '45: 10.899', '46:
10.829', '47: 10.758', '48: 10.687', '49: 10.616']

O segundo problema a ser testado está na página 425 do livro "Cálculo Numérico" de

Neide Franco, na questão de número 12.10.

Este problema teve como entrada:

0

20

1

$0.075 \cdot y$

0.5

E o resultado foi:

['0: 1.038', '1: 1.076', '2: 1.117', '3: 1.159', '4: 1.202', '5: 1.247', '6: 1.294', '7: 1.342', '8: 1.393', '9: 1.445', '10: 1.499', '11: 1.555', '12: 1.614', '13: 1.674', '14: 1.737', '15: 1.802', '16: 1.87', '17: 1.94', '18: 2.013', '19: 2.088', '20: 2.166', '21: 2.248', '22: 2.332', '23: 2.419', '24: 2.51', '25: 2.604', '26: 2.702', '27: 2.803', '28: 2.908', '29: 3.017', '30: 3.131', '31: 3.248', '32: 3.37', '33: 3.496', '34: 3.627', '35: 3.763', '36: 3.904', '37: 4.051', '38: 4.203', '39: 4.36']

O terceiro problema a ser testado está na página 426 do livro "Cálculo Numérico" de Neide Franco, na questão de número 12.16.

Este problema traz duas funções como entrada, sendo elas :

0

2

1

$0.5 \cdot y$

0.01

0

2

1

$0.38 \cdot y$

0.01

E os resultados respectivamente foram:

['0: 1.005', '1: 1.01', '2: 1.015', '3: 1.02', '4: 1.025', '5: 1.03', '6: 1.036', '7: 1.041',
'8: 1.046', '9: 1.051', '10: 1.056', '11: 1.062', '12: 1.067', '13: 1.072', '14: 1.078',
'15: 1.083', '16: 1.088', '17: 1.094', '18: 1.099', '19: 1.105', '20: 1.11', '21: 1.116',
'22: 1.122', '23: 1.127', '24: 1.133', '25: 1.138', '26: 1.144', '27: 1.15', '28: 1.156',
'29: 1.161', '30: 1.167', '31: 1.173', '32: 1.179', '33: 1.185', '34: 1.191', '35:
1.197', '36: 1.203', '37: 1.209', '38: 1.215', '39: 1.221', '40: 1.227', '41: 1.233',
'42: 1.239', '43: 1.245', '44: 1.252', '45: 1.258', '46: 1.264', '47: 1.27', '48: 1.277',
'49: 1.283', '50: 1.29', '51: 1.296', '52: 1.303', '53: 1.309', '54: 1.316', '55: 1.322',
'56: 1.329', '57: 1.335', '58: 1.342', '59: 1.349', '60: 1.356', '61: 1.362', '62:
1.369', '63: 1.376', '64: 1.383', '65: 1.39', '66: 1.397', '67: 1.404', '68: 1.411', '69:
1.418', '70: 1.425', '71: 1.432', '72: 1.439', '73: 1.446', '74: 1.454', '75: 1.461',
'76: 1.468', '77: 1.476', '78: 1.483', '79: 1.49', '80: 1.498', '81: 1.505', '82: 1.513',
'83: 1.52', '84: 1.528', '85: 1.536', '86: 1.543', '87: 1.551', '88: 1.559', '89: 1.567',
'90: 1.574', '91: 1.582', '92: 1.59', '93: 1.598', '94: 1.606', '95: 1.614', '96: 1.622',
'97: 1.63', '98: 1.638', '99: 1.647', '100: 1.655', '101: 1.663', '102: 1.671', '103:
1.68', '104: 1.688', '105: 1.697', '106: 1.705', '107: 1.714', '108: 1.722', '109:
1.731', '110: 1.74', '111: 1.748', '112: 1.757', '113: 1.766', '114: 1.775', '115:
1.783', '116: 1.792', '117: 1.801', '118: 1.81', '119: 1.819', '120: 1.828', '121:
1.838', '122: 1.847', '123: 1.856', '124: 1.865', '125: 1.875', '126: 1.884', '127:
1.893', '128: 1.903', '129: 1.912', '130: 1.922', '131: 1.932', '132: 1.941', '133:
1.951', '134: 1.961', '135: 1.971', '136: 1.98', '137: 1.99', '138: 2.0', '139: 2.01',
'140: 2.02', '141: 2.03', '142: 2.041', '143: 2.051', '144: 2.061', '145: 2.071', '146:
2.082', '147: 2.092', '148: 2.103', '149: 2.113', '150: 2.124', '151: 2.134', '152:
2.145', '153: 2.156', '154: 2.166', '155: 2.177', '156: 2.188', '157: 2.199', '158:
2.21', '159: 2.221', '160: 2.232', '161: 2.243', '162: 2.255', '163: 2.266', '164:
2.277', '165: 2.289', '166: 2.3', '167: 2.312', '168: 2.323', '169: 2.335', '170:
2.346', '171: 2.358', '172: 2.37', '173: 2.382', '174: 2.394', '175: 2.406', '176:
2.418', '177: 2.43', '178: 2.442', '179: 2.454', '180: 2.466', '181: 2.479', '182:
2.491', '183: 2.504', '184: 2.516', '185: 2.529', '186: 2.541', '187: 2.554', '188:

2.567', '189: 2.58', '190: 2.592', '191: 2.605', '192: 2.618', '193: 2.632', '194: 2.645', '195: 2.658', '196: 2.671', '197: 2.685', '198: 2.698', '199: 2.712']

['0: 1.004', '1: 1.008', '2: 1.011', '3: 1.015', '4: 1.019', '5: 1.023', '6: 1.027', '7: 1.031', '8: 1.035', '9: 1.039', '10: 1.043', '11: 1.047', '12: 1.051', '13: 1.055', '14: 1.059', '15: 1.063', '16: 1.067', '17: 1.071', '18: 1.075', '19: 1.079', '20: 1.083', '21: 1.087', '22: 1.091', '23: 1.095', '24: 1.099', '25: 1.104', '26: 1.108', '27: 1.112', '28: 1.116', '29: 1.121', '30: 1.125', '31: 1.129', '32: 1.133', '33: 1.138', '34: 1.142', '35: 1.146', '36: 1.151', '37: 1.155', '38: 1.159', '39: 1.164', '40: 1.168', '41: 1.173', '42: 1.177', '43: 1.182', '44: 1.186', '45: 1.191', '46: 1.195', '47: 1.2', '48: 1.204', '49: 1.209', '50: 1.213', '51: 1.218', '52: 1.223', '53: 1.227', '54: 1.232', '55: 1.237', '56: 1.241', '57: 1.246', '58: 1.251', '59: 1.256', '60: 1.26', '61: 1.265', '62: 1.27', '63: 1.275', '64: 1.28', '65: 1.284', '66: 1.289', '67: 1.294', '68: 1.299', '69: 1.304', '70: 1.309', '71: 1.314', '72: 1.319', '73: 1.324', '74: 1.329', '75: 1.334', '76: 1.339', '77: 1.344', '78: 1.349', '79: 1.354', '80: 1.36', '81: 1.365', '82: 1.37', '83: 1.375', '84: 1.38', '85: 1.386', '86: 1.391', '87: 1.396', '88: 1.402', '89: 1.407', '90: 1.412', '91: 1.418', '92: 1.423', '93: 1.428', '94: 1.434', '95: 1.439', '96: 1.445', '97: 1.45', '98: 1.456', '99: 1.461', '100: 1.467', '101: 1.472', '102: 1.478', '103: 1.484', '104: 1.489', '105: 1.495', '106: 1.501', '107: 1.506', '108: 1.512', '109: 1.518', '110: 1.523', '111: 1.529', '112: 1.535', '113: 1.541', '114: 1.547', '115: 1.553', '116: 1.559', '117: 1.564', '118: 1.57', '119: 1.576', '120: 1.582', '121: 1.588', '122: 1.594', '123: 1.6', '124: 1.607', '125: 1.613', '126: 1.619', '127: 1.625', '128: 1.631', '129: 1.637', '130: 1.644', '131: 1.65', '132: 1.656', '133: 1.662', '134: 1.669', '135: 1.675', '136: 1.681', '137: 1.688', '138: 1.694', '139: 1.701', '140: 1.707', '141: 1.714', '142: 1.72', '143: 1.727', '144: 1.733', '145: 1.74', '146: 1.746', '147: 1.753', '148: 1.76', '149: 1.766', '150: 1.773', '151: 1.78', '152: 1.787', '153: 1.793', '154: 1.8', '155: 1.807', '156: 1.814', '157: 1.821', '158: 1.828', '159: 1.835', '160: 1.842', '161: 1.849', '162: 1.856',

```
'163: 1.863', '164: 1.87', '165: 1.877', '166: 1.884', '167: 1.891', '168: 1.898',  
'169: 1.906', '170: 1.913', '171: 1.92', '172: 1.927', '173: 1.935', '174: 1.942',  
'175: 1.949', '176: 1.957', '177: 1.964', '178: 1.972', '179: 1.979', '180: 1.987',  
'181: 1.994', '182: 2.002', '183: 2.009', '184: 2.017', '185: 2.025', '186: 2.032',  
'187: 2.04', '188: 2.048', '189: 2.056', '190: 2.064', '191: 2.071', '192: 2.079',  
'193: 2.087', '194: 2.095', '195: 2.103', '196: 2.111', '197: 2.119', '198: 2.127',  
'199: 2.135']
```

Dificuldades enfrentadas:

Devido à simplicidade do método, a implementação não foi desafiadora.

Método de Heun

Estratégia de Implementação:

Durante a leitura do arquivo, o intervalo é armazenado em uma lista, enquanto as demais entradas são atribuídas a variáveis individuais. Em seguida, o cálculo é realizado para cada ponto no intervalo, com espaçamento determinado pelo valor de "h" informado.

Os cálculos dos valores nos pontos são feitos separadamente em três métodos no algoritmo. Primeiro, o método "passos()" determina a quantidade de pontos que serão calculados com base no intervalo e no valor de "h". Em seguida, o método "calc2()" é utilizado para retornar o valor da função nos pontos desejados, dado a função e duas variáveis. Para isso, a função "eval()" é utilizada, portanto, todas as funções devem ser escritas de acordo com a sintaxe do Python, para que sejam interpretadas corretamente.

Por fim, a função "heun()" contém o cálculo do método em si e retorna os pontos com o arredondamento em três casas decimais, utilizando a função "round()".

Estrutura dos Arquivos de Entrada/Saída:

No arquivo de entrada, são fornecidas as seguintes informações, separadas por quebras de linha ('\n'): o intervalo, o ponto inicial 'y0', a função e o valor de 'h'. Cada informação é inserida em uma linha separada, e cada conjunto de informações é separado por uma quebra de linha ('\n').

No arquivo de saída, os intervalos de pontos são apresentados indicando sua ordem em forma de lista. Cada caso é separado por uma quebra de linha ('\n').

Problemas teste:

O primeiro problema a ser testado encontra-se na página 423 do livro "Cálculo Numérico" de Neide Franco, na questão de número 12.3.

A entrada para este problema foi:

0

50

0

$(200-2y)/(200-x)^1$

E o resultado encontrado foi:

['0: 0.9', '1: 1.8', '2: 2.7', '3: 3.6', '4: 4.5', '5: 5.4', '6: 6.3', '7: 7.2', '8: 8.1', '9: 9.0',
'10: 9.9', '11: 10.321', '12: 10.266', '13: 10.211', '14: 10.157', '15: 10.102', '16:
10.047', '17: 10.486', '18: 10.429', '19: 10.371', '20: 10.314', '21: 10.256', '22:
10.198', '23: 10.141', '24: 10.083', '25: 10.025', '26: 10.488', '27: 10.427', '28:
10.367', '29: 10.306', '30: 10.246', '31: 10.185', '32: 10.124', '33: 10.064', '34:
10.003', '35: 10.491', '36: 10.427', '37: 10.363', '38: 10.299', '39: 10.235', '40:
10.171', '41: 10.107', '42: 10.043', '43: 10.556', '44: 10.489', '45: 10.421', '46:
10.353', '47: 10.286', '48: 10.218', '49: 10.15']

O segundo problema a ser testado está na página 425 do livro "Cálculo Numérico" de Neide Franco, na questão de número 12.10.

Este problema teve como entrada:

0

20

1

$0.075*y$

0.5

E o resultado foi:

['0: 1.038', '1: 1.078', '2: 1.119', '3: 1.162', '4: 1.206', '5: 1.252', '6: 1.3', '7: 1.35',

'8: 1.401', '9: 1.455', '10: 1.51', '11: 1.568', '12: 1.628', '13: 1.69', '14: 1.755', '15: 1.822', '16: 1.891', '17: 1.964', '18: 2.039', '19: 2.117', '20: 2.198', '21: 2.281', '22: 2.369', '23: 2.459', '24: 2.553', '25: 2.651', '26: 2.752', '27: 2.857', '28: 2.966', '29: 3.079', '30: 3.197', '31: 3.319', '32: 3.446', '33: 3.578', '34: 3.714', '35: 3.856', '36: 4.004', '37: 4.157', '38: 4.315', '39: 4.48']

O terceiro problema a ser testado está na página 426 do livro "Cálculo Numérico" de Neide Franco, na questão de número 12.16.

Este problema traz duas funções como entrada :

0
2
1
0.5*y
0.01

0
2
1
0.38*y
0.01

E os resultados respectivamente foram:

['0: 1.005', '1: 1.01', '2: 1.015', '3: 1.02', '4: 1.025', '5: 1.03', '6: 1.036', '7: 1.041', '8: 1.046', '9: 1.051', '10: 1.057', '11: 1.062', '12: 1.067', '13: 1.073', '14: 1.078', '15: 1.083', '16: 1.089', '17: 1.094', '18: 1.1', '19: 1.105', '20: 1.111', '21: 1.116', '22: 1.122', '23: 1.127', '24: 1.133', '25: 1.139', '26: 1.145', '27: 1.15', '28: 1.156', '29: 1.162', '30: 1.168', '31: 1.174', '32: 1.179', '33: 1.185', '34: 1.191', '35: 1.197', '36: 1.203', '37: 1.209', '38: 1.215', '39: 1.221', '40: 1.228', '41: 1.234', '42: 1.24', '43: 1.246', '44: 1.252', '45: 1.259', '46: 1.265', '47: 1.271', '48: 1.278', '49: 1.284', '50: 1.29', '51: 1.297', '52: 1.303', '53: 1.31', '54: 1.317', '55: 1.323', '56: 1.33', '57: 1.336', '58: 1.343', '59: 1.35', '60: 1.357', '61: 1.363', '62: 1.37', '63: 1.377', '64: 1.384', '65: 1.391', '66: 1.398', '67: 1.405', '68: 1.412', '69: 1.419', '70: 1.426', '71: 1.433', '72: 1.441', '73: 1.448', '74: 1.455', '75: 1.462',

'76: 1.47', '77: 1.477', '78: 1.484', '79: 1.492', '80: 1.499', '81: 1.507', '82: 1.514',
'83: 1.522', '84: 1.53', '85: 1.537', '86: 1.545', '87: 1.553', '88: 1.56', '89: 1.568',
'90: 1.576', '91: 1.584', '92: 1.592', '93: 1.6', '94: 1.608', '95: 1.616', '96: 1.624',
'97: 1.632', '98: 1.64', '99: 1.649', '100: 1.657', '101: 1.665', '102: 1.674', '103:
1.682', '104: 1.69', '105: 1.699', '106: 1.707', '107: 1.716', '108: 1.725', '109:
1.733', '110: 1.742', '111: 1.751', '112: 1.759', '113: 1.768', '114: 1.777', '115:
1.786', '116: 1.795', '117: 1.804', '118: 1.813', '119: 1.822', '120: 1.831', '121:
1.84', '122: 1.85', '123: 1.859', '124: 1.868', '125: 1.878', '126: 1.887', '127:
1.896', '128: 1.906', '129: 1.916', '130: 1.925', '131: 1.935', '132: 1.944', '133:
1.954', '134: 1.964', '135: 1.974', '136: 1.984', '137: 1.994', '138: 2.004', '139:
2.014', '140: 2.024', '141: 2.034', '142: 2.044', '143: 2.054', '144: 2.065', '145:
2.075', '146: 2.085', '147: 2.096', '148: 2.106', '149: 2.117', '150: 2.128', '151:
2.138', '152: 2.149', '153: 2.16', '154: 2.171', '155: 2.181', '156: 2.192', '157:
2.203', '158: 2.214', '159: 2.226', '160: 2.237', '161: 2.248', '162: 2.259', '163:
2.27', '164: 2.282', '165: 2.293', '166: 2.305', '167: 2.316', '168: 2.328', '169:
2.34', '170: 2.351', '171: 2.363', '172: 2.375', '173: 2.387', '174: 2.399', '175:
2.411', '176: 2.423', '177: 2.435', '178: 2.447', '179: 2.46', '180: 2.472', '181:
2.484', '182: 2.497', '183: 2.509', '184: 2.522', '185: 2.534', '186: 2.547', '187:
2.56', '188: 2.573', '189: 2.586', '190: 2.599', '191: 2.612', '192: 2.625', '193:
2.638', '194: 2.651', '195: 2.664', '196: 2.678', '197: 2.691', '198: 2.705', '199:
2.718']

['0: 1.004', '1: 1.008', '2: 1.011', '3: 1.015', '4: 1.019', '5: 1.023', '6: 1.027', '7:
1.031', '8: 1.035', '9: 1.039', '10: 1.043', '11: 1.047', '12: 1.051', '13: 1.055', '14:
1.059', '15: 1.063', '16: 1.067', '17: 1.071', '18: 1.075', '19: 1.079', '20: 1.083',
'21: 1.087', '22: 1.091', '23: 1.095', '24: 1.1', '25: 1.104', '26: 1.108', '27: 1.112',
'28: 1.117', '29: 1.121', '30: 1.125', '31: 1.129', '32: 1.134', '33: 1.138', '34: 1.142',
'35: 1.147', '36: 1.151', '37: 1.155', '38: 1.16', '39: 1.164', '40: 1.169', '41: 1.173',

'42: 1.178', '43: 1.182', '44: 1.186', '45: 1.191', '46: 1.196', '47: 1.2', '48: 1.205',
'49: 1.209', '50: 1.214', '51: 1.218', '52: 1.223', '53: 1.228', '54: 1.232', '55:
1.237', '56: 1.242', '57: 1.247', '58: 1.251', '59: 1.256', '60: 1.261', '61: 1.266',
'62: 1.27', '63: 1.275', '64: 1.28', '65: 1.285', '66: 1.29', '67: 1.295', '68: 1.3', '69:
1.305', '70: 1.31', '71: 1.315', '72: 1.32', '73: 1.325', '74: 1.33', '75: 1.335', '76:
1.34', '77: 1.345', '78: 1.35', '79: 1.355', '80: 1.36', '81: 1.366', '82: 1.371', '83:
1.376', '84: 1.381', '85: 1.387', '86: 1.392', '87: 1.397', '88: 1.402', '89: 1.408',
'90: 1.413', '91: 1.418', '92: 1.424', '93: 1.429', '94: 1.435', '95: 1.44', '96: 1.446',
'97: 1.451', '98: 1.457', '99: 1.462', '100: 1.468', '101: 1.473', '102: 1.479', '103:
1.485', '104: 1.49', '105: 1.496', '106: 1.502', '107: 1.507', '108: 1.513', '109:
1.519', '110: 1.525', '111: 1.531', '112: 1.536', '113: 1.542', '114: 1.548', '115:
1.554', '116: 1.56', '117: 1.566', '118: 1.572', '119: 1.578', '120: 1.584', '121:
1.59', '122: 1.596', '123: 1.602', '124: 1.608', '125: 1.614', '126: 1.62', '127:
1.626', '128: 1.633', '129: 1.639', '130: 1.645', '131: 1.651', '132: 1.658', '133:
1.664', '134: 1.67', '135: 1.677', '136: 1.683', '137: 1.689', '138: 1.696', '139:
1.702', '140: 1.709', '141: 1.715', '142: 1.722', '143: 1.728', '144: 1.735', '145:

1.742', '146: 1.748', '147: 1.755', '148: 1.762', '149: 1.768', '150: 1.775', '151:
1.782', '152: 1.789', '153: 1.795', '154: 1.802', '155: 1.809', '156: 1.816', '157:
1.823', '158: 1.83', '159: 1.837', '160: 1.844', '161: 1.851', '162: 1.858', '163:
1.865', '164: 1.872', '165: 1.879', '166: 1.886', '167: 1.893', '168: 1.901', '169:
1.908', '170: 1.915', '171: 1.922', '172: 1.93', '173: 1.937', '174: 1.944', '175:
1.952', '176: 1.959', '177: 1.967', '178: 1.974', '179: 1.982', '180: 1.989', '181:
1.997', '182: 2.005', '183: 2.012', '184: 2.02', '185: 2.027', '186: 2.035', '187:
2.043', '188: 2.051', '189: 2.059', '190: 2.066', '191: 2.074', '192: 2.082', '193:
2.09', '194: 2.098', '195: 2.106', '196: 2.114', '197: 2.122', '198: 2.13', '199: 2.138']

Dificultades enfrentadas:

Devido à simplicidade do método, a implementação não foi difícil. Além disso, foi utilizada a implementação do método de Euler como base para facilitar o processo.

Método Runge-Kutta de 4ª Ordem

Estratégia de Implementação:

Durante a leitura do arquivo, o intervalo é armazenado em uma lista, enquanto as demais entradas são atribuídas a variáveis individuais. Em seguida, o cálculo é realizado para cada ponto no intervalo, com espaçamento determinado pelo valor de "h" informado.

Os cálculos dos valores nos pontos são feitos separadamente em três métodos no algoritmo. O método "passos()" determina a quantidade de pontos que serão calculados com base no intervalo e no valor de "h". O método "calc2()" é utilizado para retornar o valor da função nos pontos desejados, dado a função e duas variáveis. Para isso, a função "eval()" é utilizada, portanto, todas as funções devem ser escritas de acordo com a sintaxe do Python, para que sejam interpretadas corretamente.

Por fim, a função "rungekutta()" contém o cálculo do método em si e retorna os pontos com o arredondamento em três casas decimais, utilizando a função "round()".

Estrutura dos Arquivos de Entrada/Saída:

No arquivo de entrada, as informações são fornecidas da seguinte forma: o intervalo é separado por quebras de linha ('\n') das outras informações. Em seguida, é fornecido o ponto inicial 'y0', a função e o valor de 'h'. Cada informação é inserida em uma linha separada, e cada conjunto de informações é separado por uma quebra de linha ('\n').

No arquivo de saída, cada intervalo de pontos é apresentado indicando sua ordem em forma de lista. Cada caso é separado por uma quebra de linha ('\n').

Problemas teste:

O primeiro problema a ser testado encontra-se na página 423 do livro "Cálculo Numérico" de Neide Franco, na questão de número 12.3.

A entrada para este problema foi:

0

50

0

$$(200-2y)/(200-x)^1$$

E o resultado encontrado foi:

['0: 0.9', '1: 1.8', '2: 2.7', '3: 3.6', '4: 4.5', '5: 5.4', '6: 6.3', '7: 7.2', '8: 8.1', '9: 9.0',
'10: 9.9', '11: 10.323', '12: 10.268', '13: 10.213', '14: 10.158', '15: 10.103', '16:
10.048', '17: 10.158', '18: 10.103', '19: 10.047', '20: 10.158', '21: 10.102', '22:
10.045', '23: 10.159', '24: 10.101', '25: 10.043', '26: 10.159', '27: 10.1', '28:
10.042', '29: 10.159', '30: 10.099', '31: 10.04', '32: 10.16', '33: 10.099', '34:
10.038', '35: 10.16', '36: 10.098', '37: 10.036', '38: 10.161', '39: 10.097', '40:
10.034', '41: 10.161', '42: 10.097', '43: 10.032', '44: 10.162', '45: 10.096', '46:
10.031', '47: 10.555', '48: 10.485', '49: 10.416']

O segundo problema a ser testado está na página 425 do livro "Cálculo Numérico" de Neide Franco, na questão de número 12.10.

Este problema teve como entrada:

0

20

1

$$0.075*y$$

0.5

E o resultado foi:

['0: 1.038', '1: 1.078', '2: 1.119', '3: 1.162', '4: 1.206', '5: 1.252', '6: 1.3', '7: 1.35',

'8: 1.401', '9: 1.455', '10: 1.511', '11: 1.568', '12: 1.628', '13: 1.69', '14: 1.755',
'15: 1.822', '16: 1.892', '17: 1.964', '18: 2.039', '19: 2.117', '20: 2.198', '21: 2.282',
'22: 2.369', '23: 2.46', '24: 2.554', '25: 2.651', '26: 2.752', '27: 2.858', '28: 2.967',
'29: 3.08', '30: 3.198', '31: 3.32', '32: 3.447', '33: 3.579', '34: 3.715', '35: 3.857',
'36: 4.005', '37: 4.158', '38: 4.317', '39: 4.482']

O terceiro problema a ser testado está na página 426 do livro "Cálculo Numérico" de Neide Franco, na questão de número 12.16.

Este problema envolve o uso de duas funções como entrada :

0
2
1
0.5*y
0.01

0
2
1
0.38*y
0.01

E os resultados respectivamente foram:

['0: 1.005', '1: 1.01', '2: 1.015', '3: 1.02', '4: 1.025', '5: 1.03', '6: 1.036', '7: 1.041',
'8: 1.046', '9: 1.051', '10: 1.057', '11: 1.062', '12: 1.067', '13: 1.073', '14: 1.078',
'15: 1.083', '16: 1.089', '17: 1.094', '18: 1.1', '19: 1.105', '20: 1.111', '21: 1.116',
'22: 1.122', '23: 1.127', '24: 1.133', '25: 1.139', '26: 1.145', '27: 1.15', '28: 1.156',
'29: 1.162', '30: 1.168', '31: 1.174', '32: 1.179', '33: 1.185', '34: 1.191', '35:
1.197', '36: 1.203', '37: 1.209', '38: 1.215', '39: 1.221', '40: 1.228', '41: 1.234',
'42: 1.24', '43: 1.246', '44: 1.252', '45: 1.259', '46: 1.265', '47: 1.271', '48: 1.278',
'49: 1.284', '50: 1.29', '51: 1.297', '52: 1.303', '53: 1.31', '54: 1.317', '55: 1.323',
'56: 1.33', '57: 1.336', '58: 1.343', '59: 1.35', '60: 1.357', '61: 1.363', '62: 1.37',
'63: 1.377', '64: 1.384', '65: 1.391', '66: 1.398', '67: 1.405', '68: 1.412', '69:
1.419', '70: 1.426', '71: 1.433', '72: 1.441', '73: 1.448', '74: 1.455', '75: 1.462',

'76: 1.47', '77: 1.477', '78: 1.484', '79: 1.492', '80: 1.499', '81: 1.507', '82: 1.514',
'83: 1.522', '84: 1.53', '85: 1.537', '86: 1.545', '87: 1.553', '88: 1.56', '89: 1.568',
'90: 1.576', '91: 1.584', '92: 1.592', '93: 1.6', '94: 1.608', '95: 1.616', '96: 1.624',
'97: 1.632', '98: 1.64', '99: 1.649', '100: 1.657', '101: 1.665', '102: 1.674', '103:
1.682', '104: 1.69', '105: 1.699', '106: 1.707', '107: 1.716', '108: 1.725', '109:
1.733', '110: 1.742', '111: 1.751', '112: 1.759', '113: 1.768', '114: 1.777', '115:
1.786', '116: 1.795', '117: 1.804', '118: 1.813', '119: 1.822', '120: 1.831', '121:
1.84', '122: 1.85', '123: 1.859', '124: 1.868', '125: 1.878', '126: 1.887', '127:
1.896', '128: 1.906', '129: 1.916', '130: 1.925', '131: 1.935', '132: 1.944', '133:
1.954', '134: 1.964', '135: 1.974', '136: 1.984', '137: 1.994', '138: 2.004', '139:
2.014', '140: 2.024', '141: 2.034', '142: 2.044', '143: 2.054', '144: 2.065', '145:
2.075', '146: 2.085', '147: 2.096', '148: 2.106', '149: 2.117', '150: 2.128', '151:
2.138', '152: 2.149', '153: 2.16', '154: 2.171', '155: 2.181', '156: 2.192', '157:
2.203', '158: 2.214', '159: 2.226', '160: 2.237', '161: 2.248', '162: 2.259', '163:
2.27', '164: 2.282', '165: 2.293', '166: 2.305', '167: 2.316', '168: 2.328', '169:
2.34', '170: 2.351', '171: 2.363', '172: 2.375', '173: 2.387', '174: 2.399', '175:
2.411', '176: 2.423', '177: 2.435', '178: 2.447', '179: 2.46', '180: 2.472', '181:
2.484', '182: 2.497', '183: 2.509', '184: 2.522', '185: 2.535', '186: 2.547', '187:
2.56', '188: 2.573', '189: 2.586', '190: 2.599', '191: 2.612', '192: 2.625', '193:
2.638', '194: 2.651', '195: 2.664', '196: 2.678', '197: 2.691', '198: 2.705', '199:
2.718']

['0: 1.004', '1: 1.008', '2: 1.011', '3: 1.015', '4: 1.019', '5: 1.023', '6: 1.027', '7:
1.031', '8: 1.035', '9: 1.039', '10: 1.043', '11: 1.047', '12: 1.051', '13: 1.055', '14:
1.059', '15: 1.063', '16: 1.067', '17: 1.071', '18: 1.075', '19: 1.079', '20: 1.083',
'21: 1.087', '22: 1.091', '23: 1.095', '24: 1.1', '25: 1.104', '26: 1.108', '27: 1.112',
'28: 1.117', '29: 1.121', '30: 1.125', '31: 1.129', '32: 1.134', '33: 1.138', '34: 1.142',
'35: 1.147', '36: 1.151', '37: 1.155', '38: 1.16', '39: 1.164', '40: 1.169', '41: 1.173',

'42: 1.178', '43: 1.182', '44: 1.186', '45: 1.191', '46: 1.196', '47: 1.2', '48: 1.205',
'49: 1.209', '50: 1.214', '51: 1.218', '52: 1.223', '53: 1.228', '54: 1.232', '55:
1.237', '56: 1.242', '57: 1.247', '58: 1.251', '59: 1.256', '60: 1.261', '61: 1.266',
'62: 1.27', '63: 1.275', '64: 1.28', '65: 1.285', '66: 1.29', '67: 1.295', '68: 1.3', '69:
1.305', '70: 1.31', '71: 1.315', '72: 1.32', '73: 1.325', '74: 1.33', '75: 1.335', '76:
1.34', '77: 1.345', '78: 1.35', '79: 1.355', '80: 1.36', '81: 1.366', '82: 1.371', '83:
1.376', '84: 1.381', '85: 1.387', '86: 1.392', '87: 1.397', '88: 1.402', '89: 1.408',
'90: 1.413', '91: 1.419', '92: 1.424', '93: 1.429', '94: 1.435', '95: 1.44', '96: 1.446',
'97: 1.451', '98: 1.457', '99: 1.462', '100: 1.468', '101: 1.473', '102: 1.479', '103:
1.485', '104: 1.49', '105: 1.496', '106: 1.502', '107: 1.507', '108: 1.513', '109:
1.519', '110: 1.525', '111: 1.531', '112: 1.536', '113: 1.542', '114: 1.548', '115:
1.554', '116: 1.56', '117: 1.566', '118: 1.572', '119: 1.578', '120: 1.584', '121:
1.59', '122: 1.596', '123: 1.602', '124: 1.608', '125: 1.614', '126: 1.62', '127:
1.626', '128: 1.633', '129: 1.639', '130: 1.645', '131: 1.651', '132: 1.658', '133:
1.664', '134: 1.67', '135: 1.677', '136: 1.683', '137: 1.689', '138: 1.696', '139:
1.702', '140: 1.709', '141: 1.715', '142: 1.722', '143: 1.728', '144: 1.735', '145:
1.742', '146: 1.748', '147: 1.755', '148: 1.762', '149: 1.768', '150: 1.775', '151:
1.782', '152: 1.789', '153: 1.795', '154: 1.802', '155: 1.809', '156: 1.816', '157:
1.823', '158: 1.83', '159: 1.837', '160: 1.844', '161: 1.851', '162: 1.858', '163:
1.865', '164: 1.872', '165: 1.879', '166: 1.886', '167: 1.893', '168: 1.901', '169:
1.908', '170: 1.915', '171: 1.922', '172: 1.93', '173: 1.937', '174: 1.944', '175:
1.952', '176: 1.959', '177: 1.967', '178: 1.974', '179: 1.982', '180: 1.989', '181:
1.997', '182: 2.005', '183: 2.012', '184: 2.02', '185: 2.027', '186: 2.035', '187:
2.043', '188: 2.051', '189: 2.059', '190: 2.066', '191: 2.074', '192: 2.082', '193:
2.09', '194: 2.098', '195: 2.106', '196: 2.114', '197: 2.122', '198: 2.13', '199: 2.138']

Dificuldades enfrentadas:

Por se tratar de um método simples, a implementação não foi complexa e também foi utilizada a implementação do método de Euler como base.

Método de Ralston

Estratégia de Implementação:

Durante a leitura do arquivo, o intervalo é armazenado em uma lista e as outras informações são atribuídas a variáveis específicas. Em seguida, é realizado o cálculo do resultado para cada ponto no intervalo, espaçado de acordo com o valor de "h" fornecido.

Os cálculos dos valores nos pontos são realizados em três métodos distintos no algoritmo. O primeiro método, chamado de "passos()", determina a quantidade de pontos que serão calculados com base no intervalo e no valor de "h". O segundo método, chamado "calc2()", recebe como entrada a função e duas variáveis e retorna o valor da função nos pontos especificados. Para realizar essa avaliação, utiliza-se a função "eval()", sendo necessário que todas as funções sejam escritas seguindo a sintaxe do Python para serem interpretadas corretamente.

Por fim, o método "rungekutta()" é responsável por efetuar o cálculo do método de Runge-Kutta em si, retornando os pontos resultantes. Os valores são arredondados com três casas decimais utilizando a função "round()".

Estrutura dos Arquivos de Entrada/Saída:

No arquivo de entrada, as informações são fornecidas separadamente por '\n'. Estas informações incluem o intervalo, o ponto inicial 'y0', a função e o valor de 'h'. Cada informação é inserida em uma linha e cada conjunto de informações é separado por '\n'.

No arquivo de saída, os intervalos de pontos são listados indicando sua ordem em forma de lista. Cada caso é separado por '\n'.

Problemas teste:

O problema inicial a ser testado está localizado na página 423 do livro "Cálculo Numérico" de Neide Franco, na questão número 12.3.

A entrada fornecida para este problema foi:

0

50

0

$(200-2y)/(200-x)^1$

E o resultado encontrado foi:

['0: 1.2', '1: 2.398', '2: 3.594', '3: 4.788', '4: 5.98', '5: 7.17', '6: 8.358', '7: 9.543',
'8: 10.099', '9: 10.028', '10: 10.593', '11: 10.518', '12: 10.444', '13: 10.369', '14:
10.295', '15: 10.221', '16: 10.146', '17: 10.073', '18: 9.999', '19: 10.584', '20:
10.506', '21: 10.428', '22: 10.35', '23: 10.272', '24: 10.194', '25: 10.116', '26:
10.039', '27: 10.659', '28: 10.576', '29: 10.494', '30: 10.412', '31: 10.329', '32:
10.247', '33: 10.166', '34: 10.084', '35: 10.002', '36: 10.657', '37: 10.57', '38:
10.483', '39: 10.396', '40: 10.31', '41: 10.223', '42: 10.137', '43: 10.051', '44:
10.739', '45: 10.647', '46: 10.555', '47: 10.463', '48: 10.371', '49: 10.279']

O segundo problema a ser testado está localizado na página 425 do livro "Cálculo Numérico" de Neide Franco, na questão número 12.10.

Este problema teve a seguinte entrada:

0

20

1

$0.075 \cdot y$

0.5

E o resultado foi:

['0: 1.051', '1: 1.104', '2: 1.161', '3: 1.22', '4: 1.282', '5: 1.347', '6: 1.416', '7:
1.488', '8: 1.564', '9: 1.643', '10: 1.727', '11: 1.815', '12: 1.908', '13: 2.005', '14:
2.107', '15: 2.214', '16: 2.327', '17: 2.446', '18: 2.57', '19: 2.701', '20: 2.839', '21:
2.983', '22: 3.135', '23: 3.295', '24: 3.463', '25: 3.639', '26: 3.825', '27: 4.019',
'28: 4.224', '29: 4.439', '30: 4.665', '31: 4.903', '32: 5.153', '33: 5.415', '34:
5.691', '35: 5.981', '36: 6.286', '37: 6.606', '38: 6.942', '39: 7.296']

O terceiro problema a ser testado está localizado na página 426 do livro "Cálculo Numérico" de Neide Franco, na questão número 12.16.

Este problema envolve duas funções e as seguintes entradas:

0
2
1
0.5*y
0.01

0
2
1
0.38*y
0.01

E os resultados respectivamente foram:

['0: 1.007', '1: 1.013', '2: 1.02', '3: 1.027', '4: 1.034', '5: 1.041', '6: 1.048', '7: 1.055', '8: 1.062', '9: 1.069', '10: 1.076', '11: 1.083', '12: 1.09', '13: 1.098', '14: 1.105', '15: 1.112', '16: 1.12', '17: 1.127', '18: 1.135', '19: 1.143', '20: 1.15', '21: 1.158', '22: 1.166', '23: 1.173', '24: 1.181', '25: 1.189', '26: 1.197', '27: 1.205', '28: 1.213', '29: 1.221', '30: 1.229', '31: 1.238', '32: 1.246', '33: 1.254', '34: 1.263', '35: 1.271', '36: 1.279', '37: 1.288', '38: 1.297', '39: 1.305', '40: 1.314', '41: 1.323', '42: 1.332', '43: 1.341', '44: 1.35', '45: 1.359', '46: 1.368', '47: 1.377', '48: 1.386', '49: 1.395', '50: 1.405', '51: 1.414', '52: 1.423', '53: 1.433', '54: 1.442', '55: 1.452', '56: 1.462', '57: 1.472', '58: 1.481', '59: 1.491', '60: 1.501', '61: 1.511', '62: 1.521', '63: 1.532', '64: 1.542', '65: 1.552', '66: 1.563', '67: 1.573', '68: 1.583', '69: 1.594', '70: 1.605', '71: 1.615', '72: 1.626', '73: 1.637', '74: 1.648', '75: 1.659', '76: 1.67', '77: 1.681', '78: 1.693', '79: 1.704', '80: 1.715', '81: 1.727', '82: 1.738', '83: 1.75', '84: 1.762', '85: 1.773', '86: 1.785', '87: 1.797', '88: 1.809', '89: 1.821', '90: 1.833', '91: 1.846', '92: 1.858', '93: 1.87', '94: 1.883', '95: 1.895', '96: 1.908', '97: 1.921', '98: 1.934', '99: 1.947', '100: 1.96', '101: 1.973',

'102: 1.986', '103: 1.999', '104: 2.013', '105: 2.026', '106: 2.04', '107: 2.053',
'108: 2.067', '109: 2.081', '110: 2.095', '111: 2.109', '112: 2.123', '113: 2.137',
'114: 2.151', '115: 2.166', '116: 2.18', '117: 2.195', '118: 2.209', '119: 2.224', '120:
2.239', '121: 2.254', '122: 2.269', '123: 2.284', '124: 2.299', '125: 2.315', '126:
2.33', '127: 2.346', '128: 2.361', '129: 2.377', '130: 2.393', '131: 2.409', '132:
2.425', '133: 2.441', '134: 2.458', '135: 2.474', '136: 2.491', '137: 2.507', '138:
2.524', '139: 2.541', '140: 2.558', '141: 2.575', '142: 2.592', '143: 2.61', '144:
2.627', '145: 2.645', '146: 2.662', '147: 2.68', '148: 2.698', '149: 2.716', '150:
2.734', '151: 2.752', '152: 2.771', '153: 2.789', '154: 2.808', '155: 2.827', '156:
2.846', '157: 2.865', '158: 2.884', '159: 2.903', '160: 2.922', '161: 2.942', '162:
2.962', '163: 2.981', '164: 3.001', '165: 3.021', '166: 3.042', '167: 3.062', '168:
3.082', '169: 3.103', '170: 3.124', '171: 3.145', '172: 3.166', '173: 3.187', '174:
3.208', '175: 3.23', '176: 3.251', '177: 3.273', '178: 3.295', '179: 3.317', '180:
3.339', '181: 3.361', '182: 3.384', '183: 3.406', '184: 3.429', '185: 3.452', '186:
3.475', '187: 3.498', '188: 3.522', '189: 3.545', '190: 3.569', '191: 3.593', '192:
3.617', '193: 3.641', '194: 3.665', '195: 3.69', '196: 3.714', '197: 3.739', '198:
3.764', '199: 3.789']

['0: 1.005', '1: 1.01', '2: 1.015', '3: 1.02', '4: 1.026', '5: 1.031', '6: 1.036', '7: 1.041',
'8: 1.047', '9: 1.052', '10: 1.057', '11: 1.063', '12: 1.068', '13: 1.073', '14: 1.079',
'15: 1.084', '16: 1.09', '17: 1.095', '18: 1.101', '19: 1.107', '20: 1.112', '21: 1.118',
'22: 1.124', '23: 1.129', '24: 1.135', '25: 1.141', '26: 1.146', '27: 1.152', '28:
1.158', '29: 1.164', '30: 1.17', '31: 1.176', '32: 1.182', '33: 1.188', '34: 1.194', '35:
1.2', '36: 1.206', '37: 1.212', '38: 1.218', '39: 1.225', '40: 1.231', '41: 1.237', '42:
1.243', '43: 1.25', '44: 1.256', '45: 1.262', '46: 1.269', '47: 1.275', '48: 1.282', '49:
1.288', '50: 1.295', '51: 1.301', '52: 1.308', '53: 1.314', '54: 1.321', '55: 1.328',
'56: 1.335', '57: 1.341', '58: 1.348', '59: 1.355', '60: 1.362', '61: 1.369', '62:
1.376', '63: 1.383', '64: 1.39', '65: 1.397', '66: 1.404', '67: 1.411', '68: 1.418', '69:

1.425', '70: 1.433', '71: 1.44', '72: 1.447', '73: 1.455', '74: 1.462', '75: 1.469', '76: 1.477', '77: 1.484', '78: 1.492', '79: 1.499', '80: 1.507', '81: 1.515', '82: 1.522', '83: 1.53', '84: 1.538', '85: 1.546', '86: 1.554', '87: 1.561', '88: 1.569', '89: 1.577', '90: 1.585', '91: 1.593', '92: 1.601', '93: 1.61', '94: 1.618', '95: 1.626', '96: 1.634', '97: 1.642', '98: 1.651', '99: 1.659', '100: 1.668', '101: 1.676', '102: 1.685', '103: 1.693', '104: 1.702', '105: 1.71', '106: 1.719', '107: 1.728', '108: 1.737', '109: 1.745', '110: 1.754', '111: 1.763', '112: 1.772', '113: 1.781', '114: 1.79', '115: 1.799', '116: 1.808', '117: 1.818', '118: 1.827', '119: 1.836', '120: 1.845', '121: 1.855', '122: 1.864', '123: 1.874', '124: 1.883', '125: 1.893', '126: 1.902', '127: 1.912', '128: 1.922', '129: 1.931', '130: 1.941', '131: 1.951', '132: 1.961', '133: 1.971', '134: 1.981', '135: 1.991', '136: 2.001', '137: 2.011', '138: 2.021', '139: 2.032', '140: 2.042', '141: 2.052', '142: 2.063', '143: 2.073', '144: 2.084', '145: 2.094', '146: 2.105', '147: 2.116', '148: 2.126', '149: 2.137', '150: 2.148', '151: 2.159', '152: 2.17', '153: 2.181', '154: 2.192', '155: 2.203', '156: 2.214', '157: 2.226', '158: 2.237', '159: 2.248', '160: 2.26', '161: 2.271', '162: 2.283', '163: 2.294', '164: 2.306', '165: 2.318', '166: 2.329', '167: 2.341', '168: 2.353', '169: 2.365', '170: 2.377', '171: 2.389', '172: 2.401', '173: 2.413', '174: 2.426', '175: 2.438', '176: 2.45', '177: 2.463', '178: 2.475', '179: 2.488', '180: 2.5', '181: 2.513', '182: 2.526', '183: 2.539', '184: 2.552', '185: 2.565', '186: 2.578', '187: 2.591', '188: 2.604', '189: 2.617', '190: 2.63', '191: 2.644', '192: 2.657', '193: 2.671', '194: 2.684', '195: 2.698', '196: 2.711', '197: 2.725', '198: 2.739', '199: 2.753']

Dificuldades enfrentadas:

Devido a ser uma implementação do método Runge-Kutta de 2º ordem, com apenas a modificação de $a_2 = \frac{2}{3}$, a implementação do método não apresentou dificuldades.

Método de “Shooting”

Estratégia de Implementação:

Durante a leitura do arquivo, o intervalo é armazenado em uma lista, enquanto as outras entradas são atribuídas a variáveis separadas. Em seguida, o resultado é

calculado para cada ponto dentro do intervalo, espaçado de acordo com o valor de "h" informado.

O método implementado é destinado a resolver Equações Diferenciais Ordinárias de segunda ordem, utilizando como auxílio o método de Runge-Kutta de quarta ordem.

Para calcular os valores nos pontos desejados, o algoritmo é dividido em quatro funções. A primeira função, "passos()", determina quantos pontos serão calculados com base no intervalo e no valor de "h" informados. Em seguida, temos a função "calc3()", que recebe a função desejada e três variáveis como parâmetros, e retorna o valor da função nos pontos especificados. Essa função utiliza a função "eval()" para avaliar a função desejada, portanto, todas as funções devem ser escritas em conformidade com a sintaxe do Python para que a função "eval()" possa ser utilizada corretamente.

A terceira função, "rungekutta()", é responsável por calcular os valores para cada ponto, começando de duas posições diferentes. Por fim, a função "shooting()" realiza o cálculo do método em si, utilizando os resultados obtidos na função "rungekutta()" como auxílio.

Estrutura dos Arquivos de Entrada/Saída:

No arquivo de entrada, as informações são fornecidas separadamente por '\n'. Essas informações incluem o intervalo, que é separado por '\n' das outras informações. Além disso, são fornecidos o ponto inicial 'y0', o segundo ponto 'y1' e as duas funções derivadas para o método implementado de segunda ordem. As duas funções devem ser informadas separadamente por '\n', sem a necessidade de escrever "y'" ou "z'". Apenas a função em si deve ser fornecida. Por fim, é necessário informar o valor de 'h'. Cada informação é inserida em uma linha e cada conjunto de informações é separado por '\n'.

No arquivo de saída, cada intervalo de pontos é listado indicando sua ordem em forma de lista. Cada caso é separado por '\n'.

Problemas teste:

O problema que será testado encontra-se no slide da aula.

A entrada fornecida é:
0

10
40
200
z
 $0.01 \cdot (200 - z)$
0.01
E o resultado:

['0: 40.0', '1: 40.006', '2: 40.013', '3: 40.019', '4: 40.026', '5: 40.034', '6: 40.042',
'7: 40.05', '8: 40.058', '9: 40.067', '10: 40.076', '11: 40.085', '12: 40.095', '13:
40.105', '14: 40.115', '15: 40.126', '16: 40.137', '17: 40.148', '18: 40.159', '19:
40.171', '20: 40.183', '21: 40.196', '22: 40.209', '23: 40.222', '24: 40.235', '25:
40.249', '26: 40.263', '27: 40.278', '28: 40.292', '29: 40.307', '30: 40.323', '31:
40.339', '32: 40.355', '33: 40.371', '34: 40.388', '35: 40.405', '36: 40.422', '37:
40.439', '38: 40.457', '39: 40.476', '40: 40.494', '41: 40.513', '42: 40.532', '43:
40.552', '44: 40.572', '45: 40.592', '46: 40.612', '47: 40.633', '48: 40.654', '49:
40.675', '50: 40.697', '51: 40.719', '52: 40.742', '53: 40.764', '54: 40.787', '55:
40.811', '56: 40.834', '57: 40.858', '58: 40.882', '59: 40.907', '60: 40.932', '61:
40.957', '62: 40.983', '63: 41.009', '64: 41.035', '65: 41.061', '66: 41.088', '67:
41.115', '68: 41.143', '69: 41.17', '70: 41.199', '71: 41.227', '72: 41.256', '73:
41.285', '74: 41.314', '75: 41.344', '76: 41.374', '77: 41.404', '78: 41.435', '79:
41.465', '80: 41.497', '81: 41.528', '82: 41.56', '83: 41.592', '84: 41.625', '85:
41.658', '86: 41.691', '87: 41.724', '88: 41.758', '89: 41.792', '90: 41.827', '91:
41.861', '92: 41.896', '93: 41.932', '94: 41.967', '95: 42.003', '96: 42.04', '97:
42.076', '98: 42.113', '99: 42.15', '100: 42.188', '101: 42.226', '102: 42.264', '103:
42.303', '104: 42.341', '105: 42.381', '106: 42.42', '107: 42.46', '108: 42.5', '109:
42.54', '110: 42.581', '111: 42.622', '112: 42.663', '113: 42.705', '114: 42.747',
'115: 42.789', '116: 42.832', '117: 42.875', '118: 42.918', '119: 42.962', '120:
43.006', '121: 43.05', '122: 43.094', '123: 43.139', '124: 43.184', '125: 43.23',
'126: 43.276', '127: 43.322', '128: 43.368', '129: 43.415', '130: 43.462', '131:
43.509', '132: 43.557', '133: 43.605', '134: 43.653', '135: 43.702', '136: 43.751',
'137: 43.8', '138: 43.849', '139: 43.899', '140: 43.949', '141: 44.0', '142: 44.051',

'143: 44.102', '144: 44.153', '145: 44.205', '146: 44.257', '147: 44.309', '148: 44.362', '149: 44.415', '150: 44.468', '151: 44.522', '152: 44.576', '153: 44.63', '154: 44.685', '155: 44.74', '156: 44.795', '157: 44.85', '158: 44.906', '159: 44.962', '160: 45.019', '161: 45.076', '162: 45.133', '163: 45.19', '164: 45.248', '165: 45.306', '166: 45.364', '167: 45.423', '168: 45.482', '169: 45.541', '170: 45.601', '171: 45.661', '172: 45.721', '173: 45.781', '174: 45.842', '175: 45.904', '176: 45.965', '177: 46.027', '178: 46.089', '179: 46.151', '180: 46.214', '181: 46.277', '182: 46.34', '183: 46.404', '184: 46.468', '185: 46.532', '186: 46.597', '187: 46.662', '188: 46.727', '189: 46.793', '190: 46.859', '191: 46.925', '192: 46.991', '193: 47.058', '194: 47.125', '195: 47.193', '196: 47.26', '197: 47.328', '198: 47.397', '199: 47.466', '200: 47.535', '201: 47.604', '202: 47.673', '203: 47.743', '204: 47.814', '205: 47.884', '206: 47.955', '207: 48.026', '208: 48.098', '209: 48.17', '210: 48.242', '211: 48.314', '212: 48.387', '213: 48.46', '214: 48.533', '215: 48.607', '216: 48.681', '217: 48.755', '218: 48.83', '219: 48.905', '220: 48.98', '221: 49.056', '222: 49.132', '223: 49.208', '224: 49.284', '225: 49.361', '226: 49.438', '227: 49.516', '228: 49.593', '229: 49.671', '230: 49.75', '231: 49.828', '232: 49.907', '233: 49.987', '234: 50.066', '235: 50.146', '236: 50.226', '237: 50.307', '238: 50.388', '239: 50.469', '240: 50.551', '241: 50.632', '242: 50.714', '243: 50.797', '244: 50.88', '245: 50.963', '246: 51.046', '247: 51.13', '248: 51.214', '249: 51.298', '250: 51.382', '251: 51.467', '252: 51.553', '253: 51.638', '254: 51.724', '255: 51.81', '256: 51.897', '257: 51.983', '258: 52.07', '259: 52.158', '260: 52.246', '261: 52.334', '262: 52.422', '263: 52.51', '264: 52.599', '265: 52.689', '266: 52.778', '267: 52.868', '268: 52.958', '269: 53.049', '270: 53.14', '271: 53.231', '272: 53.322', '273: 53.414', '274: 53.506', '275: 53.598', '276: 53.691', '277: 53.784', '278: 53.877', '279: 53.971', '280: 54.065', '281: 54.159', '282: 54.254', '283: 54.348', '284: 54.444', '285: 54.539', '286: 54.635', '287: 54.731', '288: 54.827', '289: 54.924', '290: 55.021', '291: 55.118', '292: 55.216', '293: 55.314', '294: 55.412', '295: 55.511', '296: 55.61', '297: 55.709', '298: 55.808', '299: 55.908', '300: 56.008', '301: 56.109', '302:

56.209', '303: 56.31', '304: 56.412', '305: 56.513', '306: 56.615', '307: 56.718',
'308: 56.82', '309: 56.923', '310: 57.026', '311: 57.13', '312: 57.234', '313:
57.338', '314: 57.442', '315: 57.547', '316: 57.652', '317: 57.758', '318: 57.863',
'319: 57.969', '320: 58.076', '321: 58.182', '322: 58.289', '323: 58.396', '324:
58.504', '325: 58.612', '326: 58.72', '327: 58.828', '328: 58.937', '329: 59.046',
'330: 59.156', '331: 59.265', '332: 59.375', '333: 59.486', '334: 59.596', '335:
59.707', '336: 59.818', '337: 59.93', '338: 60.042', '339: 60.154', '340: 60.266',
'341: 60.379', '342: 60.492', '343: 60.606', '344: 60.719', '345: 60.833', '346:
60.948', '347: 61.062', '348: 61.177', '349: 61.293', '350: 61.408', '351: 61.524',
'352: 61.64', '353: 61.757', '354: 61.873', '355: 61.99', '356: 62.108', '357:
62.226', '358: 62.344', '359: 62.462', '360: 62.581', '361: 62.7', '362: 62.819',
'363: 62.938', '364: 63.058', '365: 63.178', '366: 63.299', '367: 63.42', '368:
63.541', '369: 63.662', '370: 63.784', '371: 63.906', '372: 64.028', '373: 64.151',
'374: 64.274', '375: 64.397', '376: 64.521', '377: 64.644', '378: 64.769', '379:
64.893', '380: 65.018', '381: 65.143', '382: 65.268', '383: 65.394', '384: 65.52',
'385: 65.646', '386: 65.773', '387: 65.9', '388: 66.027', '389: 66.155', '390:
66.283', '391: 66.411', '392: 66.539', '393: 66.668', '394: 66.797', '395: 66.926',
'396: 67.056', '397: 67.186', '398: 67.316', '399: 67.447', '400: 67.578', '401:
67.709', '402: 67.841', '403: 67.973', '404: 68.105', '405: 68.237', '406: 68.37',
'407: 68.503', '408: 68.636', '409: 68.77', '410: 68.904', '411: 69.038', '412:
69.173', '413: 69.308', '414: 69.443', '415: 69.579', '416: 69.714', '417: 69.851',
'418: 69.987', '419: 70.124', '420: 70.261', '421: 70.398', '422: 70.536', '423:
70.674', '424: 70.812', '425: 70.951', '426: 71.089', '427: 71.229', '428: 71.368',
'429: 71.508', '430: 71.648', '431: 71.788', '432: 71.929', '433: 72.07', '434:
72.212', '435: 72.353', '436: 72.495', '437: 72.637', '438: 72.78', '439: 72.923',
'440: 73.066', '441: 73.209', '442: 73.353', '443: 73.497', '444: 73.642', '445:
73.786', '446: 73.931', '447: 74.077', '448: 74.222', '449: 74.368', '450: 74.514',
'451: 74.661', '452: 74.808', '453: 74.955', '454: 75.102', '455: 75.25', '456:
75.398', '457: 75.546', '458: 75.695', '459: 75.844', '460: 75.993', '461: 76.143',

'462: 76.293', '463: 76.443', '464: 76.593', '465: 76.744', '466: 76.895', '467: 77.046', '468: 77.198', '469: 77.35', '470: 77.502', '471: 77.655', '472: 77.808', '473: 77.961', '474: 78.115', '475: 78.269', '476: 78.423', '477: 78.577', '478: 78.732', '479: 78.887', '480: 79.042', '481: 79.198', '482: 79.354', '483: 79.51', '484: 79.667', '485: 79.824', '486: 79.981', '487: 80.138', '488: 80.296', '489: 80.454', '490: 80.612', '491: 80.771', '492: 80.93', '493: 81.089', '494: 81.249', '495: 81.409', '496: 81.569', '497: 81.73', '498: 81.89', '499: 82.052', '500: 82.213', '501: 82.375', '502: 82.537', '503: 82.699', '504: 82.862', '505: 83.025', '506: 83.188', '507: 83.351', '508: 83.515', '509: 83.679', '510: 83.844', '511: 84.009', '512: 84.174', '513: 84.339', '514: 84.505', '515: 84.671', '516: 84.837', '517: 85.004', '518: 85.17', '519: 85.338', '520: 85.505', '521: 85.673', '522: 85.841', '523: 86.009', '524: 86.178', '525: 86.347', '526: 86.516', '527: 86.686', '528: 86.856', '529: 87.026', '530: 87.197', '531: 87.367', '532: 87.539', '533: 87.71', '534: 87.882', '535: 88.054', '536: 88.226', '537: 88.399', '538: 88.572', '539: 88.745', '540: 88.918', '541: 89.092', '542: 89.266', '543: 89.441', '544: 89.616', '545: 89.791', '546: 89.966', '547: 90.142', '548: 90.318', '549: 90.494', '550: 90.67', '551: 90.847', '552: 91.024', '553: 91.202', '554: 91.38', '555: 91.558', '556: 91.736', '557: 91.915', '558: 92.094', '559: 92.273', '560: 92.453', '561: 92.632', '562: 92.813', '563: 92.993', '564: 93.174', '565: 93.355', '566: 93.536', '567: 93.718', '568: 93.9', '569: 94.082', '570: 94.265', '571: 94.448', '572: 94.631', '573: 94.815', '574: 94.998', '575: 95.182', '576: 95.367', '577: 95.552', '578: 95.737', '579: 95.922', '580: 96.107', '581: 96.293', '582: 96.48', '583: 96.666', '584: 96.853', '585: 97.04', '586: 97.227', '587: 97.415', '588: 97.603', '589: 97.791', '590: 97.98', '591: 98.169', '592: 98.358', '593: 98.548', '594: 98.738', '595: 98.928', '596: 99.118', '597: 99.309', '598: 99.5', '599: 99.691', '600: 99.883', '601: 100.075', '602: 100.267', '603: 100.459', '604: 100.652', '605: 100.845', '606: 101.039', '607: 101.232', '608: 101.426', '609: 101.621', '610: 101.815', '611: 102.01', '612: 102.206', '613: 102.401', '614: 102.597', '615: 102.793', '616: 102.989', '617: 103.186', '618: 103.383', '619:

103.581', '620: 103.778', '621: 103.976', '622: 104.174', '623: 104.373', '624: 104.572', '625: 104.771', '626: 104.97', '627: 105.17', '628: 105.37', '629: 105.57', '630: 105.771', '631: 105.972', '632: 106.173', '633: 106.375', '634: 106.576', '635: 106.778', '636: 106.981', '637: 107.184', '638: 107.387', '639: 107.59', '640: 107.794', '641: 107.997', '642: 108.202', '643: 108.406', '644: 108.611', '645: 108.816', '646: 109.022', '647: 109.227', '648: 109.433', '649: 109.64', '650: 109.846', '651: 110.053', '652: 110.26', '653: 110.468', '654: 110.676', '655: 110.884', '656: 111.092', '657: 111.301', '658: 111.51', '659: 111.719', '660: 111.929', '661: 112.139', '662: 112.349', '663: 112.559', '664: 112.77', '665: 112.981', '666: 113.192', '667: 113.404', '668: 113.616', '669: 113.828', '670: 114.041', '671: 114.254', '672: 114.467', '673: 114.681', '674: 114.894', '675: 115.108', '676: 115.323', '677: 115.537', '678: 115.752', '679: 115.968', '680: 116.183', '681: 116.399', '682: 116.615', '683: 116.832', '684: 117.048', '685: 117.265', '686: 117.483', '687: 117.7', '688: 117.918', '689: 118.137', '690: 118.355', '691: 118.574', '692: 118.793', '693: 119.013', '694: 119.232', '695: 119.452', '696: 119.673', '697: 119.893', '698: 120.114', '699: 120.335', '700: 120.557', '701: 120.779', '702: 121.001', '703: 121.223', '704: 121.446', '705: 121.669', '706: 121.892', '707: 122.116', '708: 122.34', '709: 122.564', '710: 122.788', '711: 123.013', '712: 123.238', '713: 123.464', '714: 123.689', '715: 123.915', '716: 124.142', '717: 124.368', '718: 124.595', '719: 124.822', '720: 125.05', '721: 125.277', '722: 125.506', '723: 125.734', '724: 125.963', '725: 126.191', '726: 126.421', '727: 126.65', '728: 126.88', '729: 127.11', '730: 127.341', '731: 127.571', '732: 127.802', '733: 128.034', '734: 128.265', '735: 128.497', '736: 128.729', '737: 128.962', '738: 129.195', '739: 129.428', '740: 129.661', '741: 129.895', '742: 130.129', '743: 130.363', '744: 130.598', '745: 130.833', '746: 131.068', '747: 131.303', '748: 131.539', '749: 131.775', '750: 132.011', '751: 132.248', '752: 132.485', '753: 132.722', '754: 132.96', '755: 133.198', '756: 133.436', '757: 133.674', '758: 133.913', '759: 134.152', '760: 134.391', '761: 134.631', '762: 134.871', '763: 135.111', '764:

135.351', '765: 135.592', '766: 135.833', '767: 136.075', '768: 136.316', '769:
136.558', '770: 136.801', '771: 137.043', '772: 137.286', '773: 137.529', '774:
137.773', '775: 138.016', '776: 138.26', '777: 138.505', '778: 138.749', '779:
138.994', '780: 139.24', '781: 139.485', '782: 139.731', '783: 139.977', '784:
140.223', '785: 140.47', '786: 140.717', '787: 140.964', '788: 141.212', '789:
141.46', '790: 141.708', '791: 141.956', '792: 142.205', '793: 142.454', '794:
142.704', '795: 142.953', '796: 143.203', '797: 143.453', '798: 143.704', '799:
143.955', '800: 144.206', '801: 144.457', '802: 144.709', '803: 144.961', '804:
145.213', '805: 145.466', '806: 145.719', '807: 145.972', '808: 146.225', '809:
146.479', '810: 146.733', '811: 146.988', '812: 147.242', '813: 147.497', '814:
147.752', '815: 148.008', '816: 148.264', '817: 148.52', '818: 148.776', '819:
149.033', '820: 149.29', '821: 149.547', '822: 149.805', '823: 150.063', '824:
150.321', '825: 150.579', '826: 150.838', '827: 151.097', '828: 151.357', '829:
151.616', '830: 151.876', '831: 152.136', '832: 152.397', '833: 152.658', '834:
152.919', '835: 153.18', '836: 153.442', '837: 153.704', '838: 153.966', '839:
154.229', '840: 154.492', '841: 154.755', '842: 155.018', '843: 155.282', '844:
155.546', '845: 155.81', '846: 156.075', '847: 156.34', '848: 156.605', '849:
156.871', '850: 157.137', '851: 157.403', '852: 157.669', '853: 157.936', '854:
158.203', '855: 158.47', '856: 158.737', '857: 159.005', '858: 159.274', '859:
159.542', '860: 159.811', '861: 160.08', '862: 160.349', '863: 160.619', '864:
160.889', '865: 161.159', '866: 161.429', '867: 161.7', '868: 161.971', '869:
162.242', '870: 162.514', '871: 162.786', '872: 163.058', '873: 163.331', '874:
163.604', '875: 163.877', '876: 164.15', '877: 164.424', '878: 164.698', '879:
164.972', '880: 165.247', '881: 165.522', '882: 165.797', '883: 166.072', '884:
166.348', '885: 166.624', '886: 166.9', '887: 167.177', '888: 167.454', '889:

167.731', '890: 168.009', '891: 168.286', '892: 168.565', '893: 168.843', '894:
169.122', '895: 169.401', '896: 169.68', '897: 169.959', '898: 170.239', '899:
170.519', '900: 170.8', '901: 171.08', '902: 171.361', '903: 171.643', '904:
171.924', '905: 172.206', '906: 172.488', '907: 172.771', '908: 173.054', '909:

173.337', '910: 173.62', '911: 173.904', '912: 174.188', '913: 174.472', '914:
174.756', '915: 175.041', '916: 175.326', '917: 175.612', '918: 175.897', '919:
176.183', '920: 176.469', '921: 176.756', '922: 177.043', '923: 177.33', '924:
177.617', '925: 177.905', '926: 178.193', '927: 178.481', '928: 178.77', '929:
179.059', '930: 179.348', '931: 179.637', '932: 179.927', '933: 180.217', '934:
180.507', '935: 180.798', '936: 181.089', '937: 181.38', '938: 181.672', '939:
181.963', '940: 182.255', '941: 182.548', '942: 182.84', '943: 183.133', '944:
183.427', '945: 183.72', '946: 184.014', '947: 184.308', '948: 184.602', '949:
184.897', '950: 185.192', '951: 185.487', '952: 185.783', '953: 186.079', '954:
186.375', '955: 186.671', '956: 186.968', '957: 187.265', '958: 187.562', '959:
187.86', '960: 188.158', '961: 188.456', '962: 188.754', '963: 189.053', '964:
189.352', '965: 189.651', '966: 189.951', '967: 190.251', '968: 190.551', '969:
190.851', '970: 191.152', '971: 191.453', '972: 191.755', '973: 192.056', '974:
192.358', '975: 192.66', '976: 192.963', '977: 193.266', '978: 193.569', '979:
193.872', '980: 194.176', '981: 194.48', '982: 194.784', '983: 195.088', '984:
195.393', '985: 195.698', '986: 196.004', '987: 196.309', '988: 196.615', '989:
196.922', '990: 197.228', '991: 197.535', '992: 197.842', '993: 198.149', '994:
198.457', '995: 198.765', '996: 199.073', '997: 199.382', '998: 199.691', '999:
200.0']

Dificuldades enfrentadas:

Nenhuma dificuldade encontrada.

Considerações Finais

Neste relatório, os métodos utilizados possuem abordagens muito semelhantes, o que reduz a dificuldade de implementação.

