

# ***Octave Correction Tool: uma Ferramenta de Correção de Rotinas Computacionais para a Disciplina de Cálculo Numérico***

**Wesley C. Silva<sup>1</sup>, João Augusto G. Castilho<sup>1</sup>, Iara C. R. Silva<sup>2</sup>, Erikson F. Morais<sup>1</sup>**

<sup>1</sup>Departamento de Ciência da Computação – Universidade Tecnológica Federal do Paraná  
Ponta Grossa – PR – Brazil

<sup>2</sup>Departamento de Matemática – Universidade Tecnológica Federal do Paraná  
Ponta Grossa – PR – Brazil

wesley-0512@hotmail.com, joaocastilho@alunos.utfpr.edu.br

{iarasilva,emaorais}@utfpr.edu.br

**Abstract.** *The Octave Correction Tool is a tool for correcting routines implemented in GNU Octave/MATLAB. The weekly labs help students understand the methods presented in class and become familiar with the programming language used in the course. The correction tool presented in this article provides a quick feedback and an accurate correction for students and optimizes the Professor's time providing by automatically spreadsheets and student performance reports.*

**Resumo.** *O Octave Correction Tool é uma ferramenta de correção de rotinas implementadas em GNU Octave/MATLAB. Os laboratórios computacionais semanais de cálculo numérico ajudam os alunos na compreensão dos métodos apresentados em aula e a se familiarizar com a linguagem de programação utilizada no curso. A ferramenta de correção apresentada neste artigo, fornece um feedback rápido e uma correção mais precisa aos alunos e otimiza o tempo do professor fornecendo planilhas com notas e relatórios de desempenhos dos alunos de forma automática.*

## **1. Introdução**

O cálculo numérico ou métodos numéricos é a sinergia do computador eletrônico programável, pela análise matemática e pela oportunidade de resolver problemas grandes e complexos em aplicações. A necessidade de avanços em aplicações, como a previsão balística, o transporte de nêutrons e a dinâmica de fluidos multidimensional e não-estacionária impulsionaram o desenvolvimento do computador e dependeram fortemente dos avanços de métodos numéricos e modelagem matemática [SIAM 2019, Sharma and Gobbert 2010, Pinçon 2007].

Pesquisas nas áreas de métodos numéricos e computação científica contribuíram para o desenvolvimento de muitos pacotes de softwares e linguagens de programação como por exemplo: MATLAB/GNU Octave, Python Numpy, R e Julia [Bryant 2012].

O ensino de cálculo numérico como matéria obrigatória dos currículos de cursos de engenharia, teve início em 1953, na recém-criada Escola de Engenharia de São

Carlos-USP (EES-USP). Na década de 1960, com o advento dos computadores digitais, o cálculo numérico passou a ser uma disciplina existente em praticamente todas as instituições de ensino superior nas áreas de Ciências Exatas e de Tecnologia [Frederico F. Campos 2014]. Os tópicos principais vistos em uma disciplina de cálculo numérico são: zeros de funções, integração de funções contínuas, sistemas lineares, interpolação, ajuste de curvas e resolução aproximada de equações diferenciais.

Apesar de grande parte dos planos de ensino serem 100% teóricos [Borges 2016], é importante que os alunos saibam programar os métodos numéricos para aplicá-los em problemas de ciências e engenharias. Hoje há vários livros didáticos de cálculo numérico que utilizam uma linguagem de programação [Chapra 2013], [Quarteroni and Saleri 2007], [Kiusalaas 2014] e [Grewal 2018]. O software MATLAB é a linguagem mais utilizada na computação numérica [Sharma and Gobbert 2010, Pinçon 2007, Baudin 2016, Machine 2006].

Na Universidade Tecnológica Federal do Paraná - *Campus* Ponta Grossa (UTFPR-PG), a disciplina de cálculo numérico é 50% teórica e 50% prática. A linguagem de programação utilizada para essa disciplina é MATLAB [MathWorks 1994] e GNU Octave [John W. Eaton and Wehbring 1998]. O Gnu Octave é uma ferramenta computacional e uma linguagem de programação 95% compatível com o MATLAB, porém gratuito. Os alunos são incentivados a implementar os métodos vistos nas aulas através de laboratórios semanais e as notas dessas atividades compõem a nota final do aluno. As atividades computacionais são individuais e a correção delas demandam muito tempo, pois o professor precisa executar as rotinas dos alunos uma por uma.

Para otimizar o tempo de correção e para que os alunos tenham um *feedback* mais rápido e preciso em relação ao seu desempenho, foi realizada uma pesquisa das ferramentas de correções de rotinas computacionais existentes. Alguns dos corretores são destinados para competições de programação [Mooshak 2016, PC2 2016] e outros utilizados em disciplinas de programação de computadores como o SuSy [SuSy ], desenvolvido pelo Professor Tomasz Kowaltowski do Instituto de Computação da Universidade Estadual de Campinas, e o BOCA-Lab [Débora Cabral Nazário1 2010] utilizado na disciplina Linguagem de Programação I, no curso de Bacharelado em Ciência da Computação da UDESC. O SuSy aceita rotinas nas linguagens Python e C e o corretor BOCA-Lab limita-se nas as linguagens C, C++ e Java, ambos os corretores não estão disponíveis para outras instituições. Dessa forma, foi desenvolvida uma ferramenta de correção de rotinas implementadas em Gnu Octave/Matlab destinada, a priori, para a disciplina de cálculo numérico.

Esse artigo está estruturado da seguinte maneira: na seção 2 é definido como foi implementada a ferramenta *Octave Correction Tool*; na seção 3 é apresentada algumas limitações dessa plataforma; na seção 4 é explanado o funcionamento do *Octave Correction Tool* e por fim a conclusão.

## **2. Implementação da ferramenta *Octave Correction Tool***

Essa ferramenta foi implementada na linguagem C e executada no sistema Operacional Linux Ubuntu 16.04 LTS, seus módulos são: `manager`, `API.octave` e `linked.list`.

## 2.1. Módulo manager

Esse é o módulo principal do sistema. Ele gerencia toda a ferramenta como: insere novas atividades e testes; cria turmas; define qual API para correção será usada e configura todos os parâmetros para a execução da tarefa do aluno.

Nele foram definidas várias rotinas, as principais são:

**Orquestrador** - Esta rotina é quem gerencia todo processo de correção. Inicialmente ela gera uma fila de alunos que são submetidos a correção. Com isso, todos os alunos, um por vez, são submetidos a uma bateria de testes pré-cadastrados pelo usuário. Para realização dos testes, o orquestrador chama o Gnu Octave para executar a rotina implementada pelo aluno. As saídas (numéricas) são comparadas e com base nas soluções dos testes é atribuída uma nota ao aluno.

**Registra Testes** - A rotina “Registra Testes” é responsável por fazer o registro de um ou mais testes para um exercício já cadastrado. O registro pode ser realizado de duas formas, a manual e a automática que estão especificadas na seção 4.1.

**Executa Correção** - Essa rotina, realiza a comparação entre a saída da tarefa do aluno e o gabarito cadastrado pelo professor. Há uma tolerância pré-estabelecida (um centésimo) em relação ao gabarito gerado pelo professor, caso a solução gerada pelo aluno esteja fora dessa tolerância o mesmo tem um desconto na nota. Por exemplo, suponha que a resposta do gabarito seja 1,3456, e os resultados gerados pelas rotinas do aluno A e B sejam 1,3457 e 1,5143, respectivamente. O aluno A não tem sua nota descontada, pois o valor absoluto da diferença entre os resultados do gabarito e do aluno é menor que um centésimo. Já o aluno B, o valor absoluto da diferença de seu resultado com a do gabarito é de 0,1687, esse valor é subtraído de sua nota. O tempo computacional também é levado em conta na correção, caso ultrapasse um tempo limite estabelecido no gabarito, o aluno também tem sua nota descontada. Por exemplo, o tempo computacional do gabarito foi de três segundos e do aluno foi de quatro segundos, o fator de descréscimo da nota é o valor absoluto da diferença entre o tempo de execução do gabarito e do aluno multiplicado por um centésimo.

**Salva Notas** - A rotina “Salva Notas” é responsável por salvar em uma planilha as notas de uma turma obtidas nos testes.

## 2.2. Módulo API Octave

Esse módulo é responsável por se comunicar com o Gnu Octave para realizar as correções. Para isso, temos duas principais funções:

**Gerador de Query** - A função Gerador de Query configura todos os parâmetros para a comunicação com a linguagem das tarefas, ou seja, ela faz a ponte entre a linguagem em que os alunos implementaram as tarefas e a ferramenta em si.

**Executar Exercício** - A função Executar Exercício é a responsável em fazer a chamada ao Gnu Octave. Para isso, utiliza-se os dados gerados pelo Gerador de Query, que contém o diretório da tarefa do aluno e os parâmetros de entrada da tarefa. A saída da tarefa do aluno é salva e lida pelo *manager*, que é o responsável pelas correções das atividades.

### 2.3. Módulo `linked_list`

O módulo `linked_list` tem a implementação de funções básicas de listas encadeadas. Estas listas são necessárias, pois há um conjunto de dados que possuem tamanho variável, por exemplo, quantidade de alunos das turmas, parâmetros de saídas e entradas das atividades computacionais.

Uma lista encadeada em formato de fila é utilizada para inserir o registro acadêmico (RA) de cada aluno a fim de estabelecer uma ordem no processo de correção. Essa fila é utilizada durante toda execução do sistema desde a execução do processo do aluno até a inserção das notas nas planilhas. Uma outra fila é utilizada para salvar as notas dos alunos em cada um dos testes realizados nas tarefas. Uma lista encadeada em formato de pilha é utilizada para salvar as saídas de cada teste cadastrado e outra para salvar as saídas das rotinas dos alunos.

### 3. Limitações da ferramenta *Octave Correction Tool*

Essa ferramenta foi construída em março de 2019 e possui algumas limitações como:

- (i) o usuário pode criar no máximo 999 testes, isso se deve à padronização utilizada para nomear os arquivos;
- (ii) a rotina do gabarito não pode ultrapassar mais de uma hora de duração em um teste específico;
- (iii) a API `octave` foi construída para corrigir rotinas de métodos numéricos, logo, as saídas permitidas deverão ser numéricas;
- (iv) por enquanto, corrige apenas rotinas implementadas em Gnu Octave e Matlab;
- (v) essa ferramenta está disponível apenas para o sistema operacional Linux.

### 4. Funcionamento da Plataforma *Octave Correction Tool*

A tela inicial do plataforma de correção, chamado de *Octave Correction Tool* pode ser vista na figura 1.



Figura 1. Octave Correction Tool (OC Tool).

O procedimento para cadastrar a(s) turma(s) de cálculo numérico deve(m) ser feita(s) no *OC Tool*, para isso, o usuário deve ir em configurações e escolher a opção cadastrar turma. Após isso o usuário pode cadastrar os exercícios realizados ao longo do curso.

#### 4.1. Cadastrar um exercício

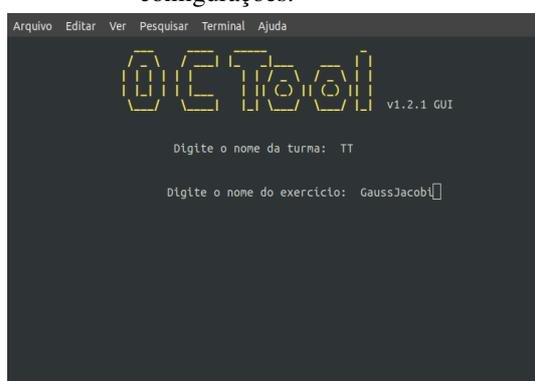
Para cadastrar um exercício, o usuário deve ir em configurações e depois escolher a opção de cadastrar exercício. A figura 2 mostra como isso é feito.



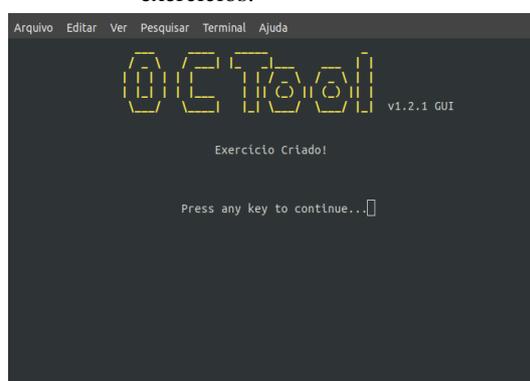
(a) Passo 1: Escolha a opção configurações.



(b) Passo 2: Escolha a opção cadastrar exercícios.



(c) Passo 3: Digite a turma e o nome do exercício.

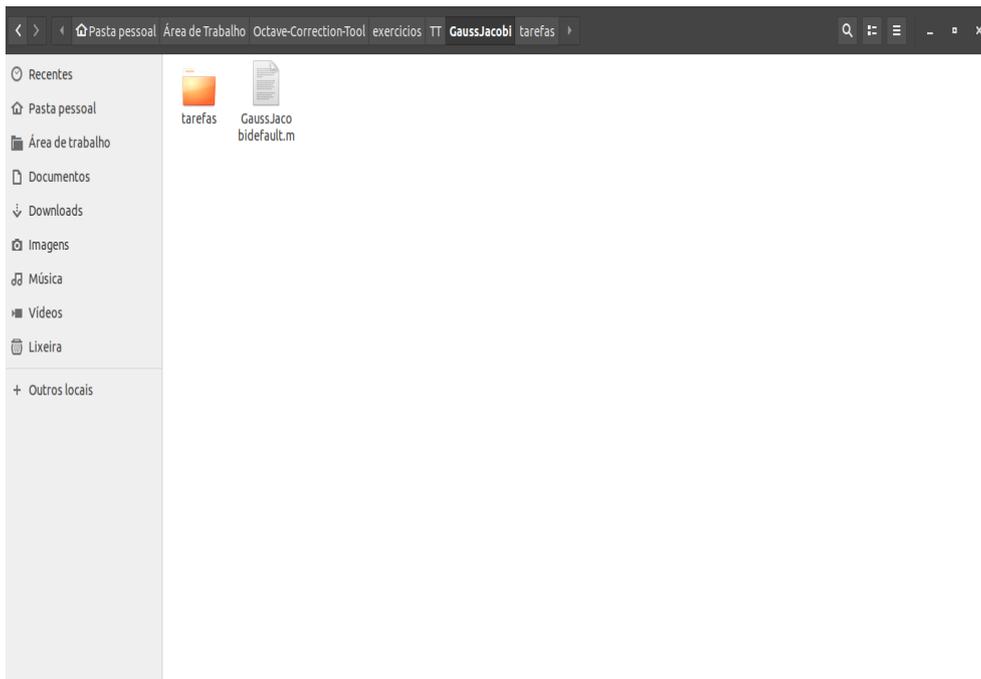


(d) Finalização do cadastro do exercício.

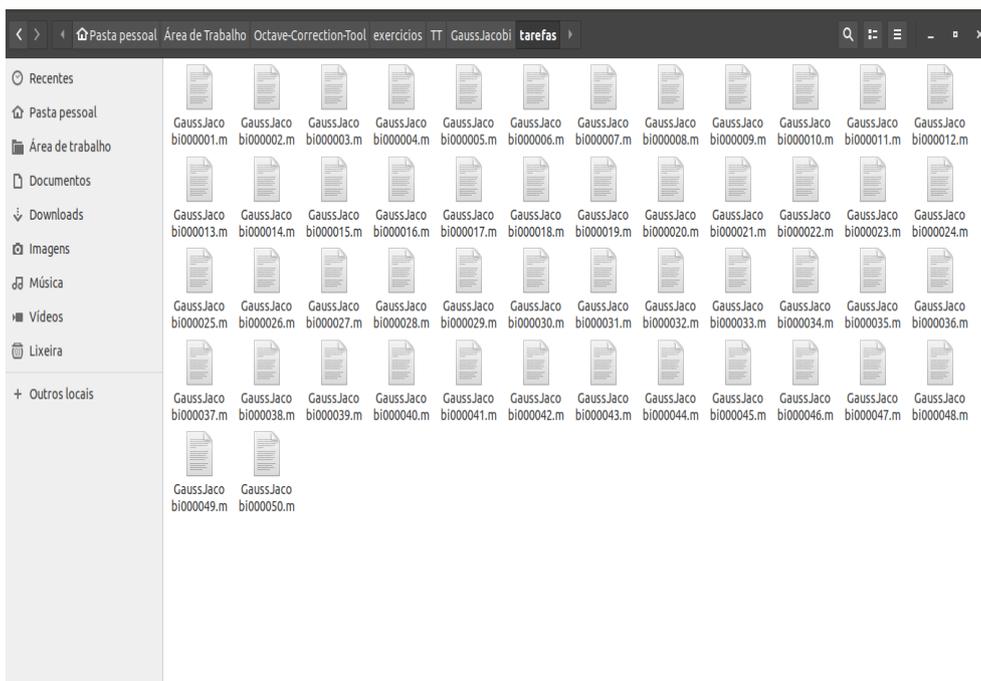
Figura 2. Cadastro do exercício.

Após o cadastro do exercício, a ferramenta cria um diretório com o nome do exercício e nele o gabarito deve ser colocado como mostra a figura 3a. O nome desse gabarito, obrigatoriamente, deve ser o nome do exercício seguida da palavra *default*. Nesse diretório, a ferramenta também cria um outro diretório chamado de tarefas. Neste diretório, são armazenadas<sup>1</sup> as rotinas dos alunos como pode ser visto na figura 3b. Os nomes das rotinas dos alunos devem ser o nome do exercício seguida pelo RA (registro acadêmico) do aluno, caso alguma rotina não siga esse padrão, a ferramenta não consegue encontrar e a rotina não pode ser corrigida.

<sup>1</sup>Esse armazenamento é feito de forma manual. O enunciado da atividade computacional bem como os envios das rotinas computacionais implementadas pelos alunos são feitos pelo Moodle.



**(a)** Passo 1: Diretório do exercício criado pela ferramenta.



**(b)** Passo 2: Diretório tarefas com as rotinas dos alunos.

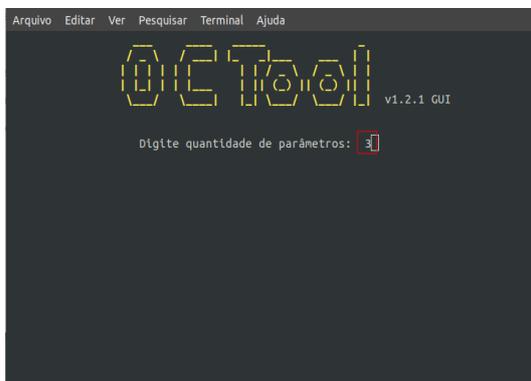
Os testes podem ser cadastrados de duas formas: manual ou automática. A forma manual consiste na inserção, na pasta do exercício, de arquivos de entrada e saída com resultados que serão utilizados a fim de comparação. Para o usuário cadastrar os testes de forma automática ele deve ir em configurações, selecionar cadastrar teste (automático) e seguir os passos descritos na figura 3.



(c) Passo 1: Escolha a turma e o exercício para cadastrar os testes.



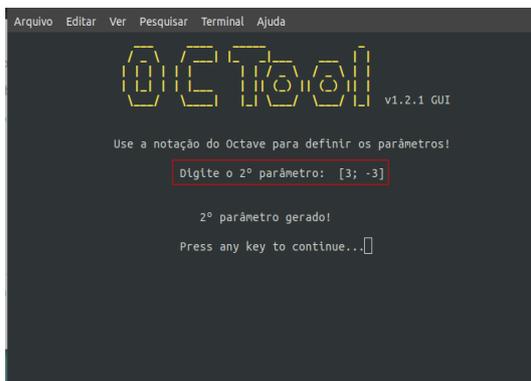
(d) Passo 2: Digite a quantidade de testes.



(e) Passo 3: Digite a quantidade de parâmetros de entrada da *function*.



(f) Passo 4: Digite o primeiro parâmetro.



(g) Passo 5: Digite o segundo parâmetro.



(h) Passo 6: Digite o terceiro parâmetro.

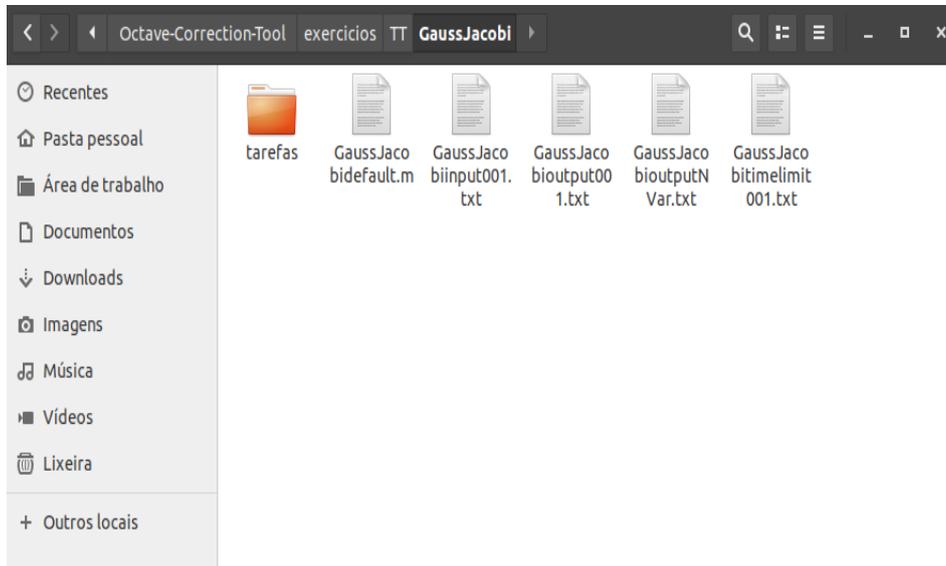
**Figura 3. Cadastro do teste.**

Essa figura acima, exemplifica a inserção de um teste do exercício GaussJacobi<sup>2</sup> que precisa de três parâmetros de entrada.

Após cadastrar o primeiro teste no diretório exercícios, além dos arquivos de *inputs* e *outputs*, com as entradas e saídas que são utilizadas nos testes, há mais outros dois

<sup>2</sup>Nessa tarefa o aluno deve implementar uma rotina que resolve um sistema linear através do método Gauss-Jacobi.

arquivos *\*outputNvar* e *\*timelimit*, com o número de parâmetros de saída e o tempo de execução da rotina do teste, respectivamente, isso pode ser visto na figura 4.

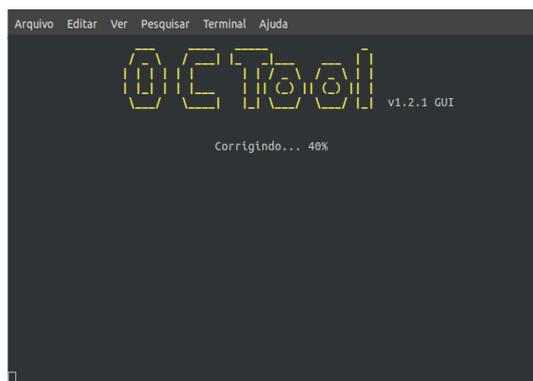


**Figura 4. Arquivos gerados após cadastrar um teste de forma automática.**

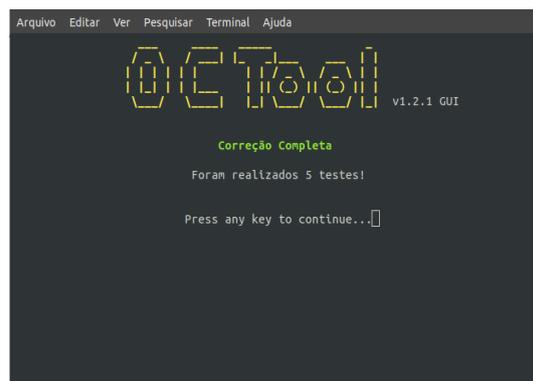
O processo descrito na figura 3 deve se repetir até que todos os testes sejam cadastrados.

#### 4.2. Correção de um exercício

Para a correção do exercício basta o usuário escolher a opção iniciar correção e inserir a turma e o exercício que será corrigido. O processo da correção e finalização do mesmo encontra-se na figura 5.



**(a)** Porcentagem das atividades corrigidas de uma turma.



**(b)** Mensagem de finalização da correção.

**Figura 5. Correção de Exercício.**

Após a correção, a ferramenta cria uma planilha com as notas relativas a cada testes e a média aritmética deles e um relatório com os resultados dos testes de cada aluno, informando o tempo computacional e as saídas geradas de cada testes, assim fica

fácil saber qual foi o erro do aluno. Na figura 6 é apresentada a planilha de notas e o relatório de desempenho gerados após a correção.

1	Nota Parcial - GaussJacobi						
2	RA	Média	Teste 1	Teste 2	Teste 3	Teste 4	Teste 5
3	1	10.00	10.00	10.00	10.00	10.00	10.00
4	2	10.00	10.00	10.00	10.00	10.00	10.00
5	3	10.00	10.00	10.00	10.00	10.00	10.00
6	4	8.33	10.00	10.00	10.00	10.00	1.67
7	5	8.50	10.00	10.00	10.00	10.00	2.50
8	6	6.00	0.00	0.00	0.00	0.00	0.00
9	7	10.00	10.00	10.00	10.00	10.00	10.00
10	8	10.00	10.00	10.00	10.00	10.00	10.00
11	9	10.00	10.00	10.00	10.00	10.00	10.00
12	10	8.50	10.00	10.00	10.00	10.00	2.50
13	11	10.00	10.00	10.00	10.00	10.00	10.00
14	12	10.00	10.00	10.00	10.00	10.00	10.00
15	13	10.00	10.00	10.00	10.00	10.00	10.00
16	14	5.50	10.00	7.50	10.00	0.00	0.00
17	15	10.00	10.00	10.00	10.00	10.00	10.00
18	16	10.00	10.00	10.00	10.00	10.00	10.00
19	17	10.00	10.00	10.00	10.00	10.00	10.00
20	18	10.00	10.00	10.00	10.00	10.00	10.00
21	19	10.00	10.00	10.00	10.00	10.00	10.00
22	20	10.00	10.00	10.00	10.00	10.00	10.00
23	21	10.00	10.00	10.00	10.00	10.00	10.00
24	22	10.00	10.00	10.00	10.00	10.00	10.00
25	23	10.00	10.00	10.00	10.00	10.00	10.00
26	24	10.00	10.00	10.00	10.00	10.00	10.00
27	25	10.00	10.00	10.00	10.00	10.00	10.00
28	26	0.00	0.00	0.00	0.00	0.00	0.00
29	27	8.50	10.00	10.00	10.00	10.00	2.50
30	28	10.00	10.00	10.00	10.00	10.00	10.00
31	29	10.00	10.00	10.00	10.00	10.00	10.00
32	30	10.00	10.00	10.00	10.00	10.00	10.00
33	31	10.00	10.00	10.00	10.00	10.00	10.00
34	32	10.00	10.00	10.00	10.00	10.00	10.00

(a) Planilha gerada pela ferramenta OC Tool.

```

disc (1): 0 | disc(2): 0 | Nota Final: 10

DEFAULT: inf, ALUNO: inf
disc (1): 0 | disc(2): 0 | Nota Final: 10

TESTE 005
O tempo limite é: 1 seg.
DEFAULT: 1000.000000, ALUNO: 1001.000000
disc (1): 5.000000 | disc(2): 0 | Nota Final: 5.000000

DEFAULT: -inf, ALUNO: -inf
disc (1): 0 | disc(2): 0 | Nota Final: 10

DEFAULT: inf, ALUNO: -inf
disc (1): 0 | disc(2): 0 | Nota Final: 0

DEFAULT: inf, ALUNO: -inf
disc (1): 0 | disc(2): 0 | Nota Final: 0

DEFAULT: -inf, ALUNO: inf
disc (1): 0 | disc(2): 0 | Nota Final: 0

DEFAULT: inf, ALUNO: -inf
disc (1): 0 | disc(2): 0 | Nota Final: 0

%
ALUNO: 000006

TESTE 001
O tempo limite é: 1 seg.
DEFAULT: 14.000000, ALUNO: 1.000000
disc (1): 65.000000 | disc(2): 12 | Nota Final: 0.000000

```

(b) Relatório de desempenho dos alunos.

Figura 6. Planilha e relatório gerados pela ferramenta OC Tool .

Na figura 6a pode-se observar que o aluno de RA 000005 não foi bem no teste

5 do exercício, o relatório 6b compara a saídas do gabarito com as do aluno. Observe que algumas saídas obtidas pela rotina do aluno são diferentes comparando com as do gabarito.

As rotinas utilizadas na apresentação do funcionamento da ferramenta *OC Tool* foram extraídas de uma turma real de cálculo numérico, porém com outros registros acadêmicos para manter a privacidade dos alunos.

## 5. Conclusão

A prática de programação em disciplinas de cálculo numérico, faz com que o aluno não fique preso apenas na teoria e possa resolver problemas nas ciências e engenharias através de um algoritmo implementado por ele. Atividades semanais onde o aluno possa implementar os métodos numéricos vistos nas aulas teóricas, fazem com que ele se familiarize com a linguagem utilizada.

A ferramenta *Octave Correction Tool* permite aos professores que façam uma correção em segundos e tenham acessos a planilhas de notas e relatórios de desempenho dos alunos. Já os alunos tem um *feedback* mais rápido, permitindo a eles saber se tiveram ou não êxito na tarefa. Outro ponto positivo da ferramenta é em relação à precisão da correção.

Na UTFPR *Campus* Ponta Grossa, essa ferramenta está sendo utilizada em 50% das turmas de cálculo numérico desde o primeiro semestre de 2019. Logo que termina o prazo de entrega da atividade computacional, em pouca horas, os alunos já têm suas notas disponíveis no Moodle.

Para trabalhos futuros será implementada uma função para a verificação de plágio e serão desenvolvidas API's para outras linguagens de programação, permitindo que o professor de cálculo numérico trabalhe com outras linguagens e que essa ferramenta possa ser utilizada em outras disciplinas como em algoritmos e estrutura de dados. Será realizado também um estudo empírico para avaliar a usabilidade e adoção da ferramenta pelos professores para a verificar os dos pontos positivos e negativos da ferramenta.

## Referências

- Baudin, R. (2016). “run time comparison of matlab, scilab and gnu octave on various benchmark programs”. <http://roland65.free.fr/benchmarks/benchmarks-0.2.pdf>.
- Borges, P. A. P. (2016). Diferenciação de enfoque no ensino de cálculo numérico. In *XII Encontro Nacional de Educação Matemática*.
- Bryant, A. (2012). Matlab, r, and julia: Languages for data analysis. <http://radar.oreilly.com/2012/10/matlab-r-julia-languages-for-data-analysis.html>.
- Chapra, S. C. (2013). *Métodos Numéricos Aplicados com MATLAB para Engenheiros e Cientistas*. AMGH, 3.ed. edition.
- Débora Cabral Nazário<sup>1</sup>, A. d. S. (2010). Boca-lab: Corretor automático de código adaptado ao ensino de linguagem de programação. In *Computer on the Beach: Trilha de Computação aplicada à Educação*, pages 42–46, Florianópolis.

- Frederico F. Campos, F. (2014). *Algoritmos Numéricos*. LTC, 2.ed. edition.
- Grewal, B. S. (2018). *Numerical Methods in Engineering and Science: (C, C++, and MATLAB)*. Mercury Learning and Information, 1.ed. edition.
- John W. Eaton, David Bateman, S. H. and Wehbring, R. (1998). *GNU Octave version 3.8.1 manual: a high-level interactive language for numerical computations*. CreateSpace Independent Publishing Platform. ISBN 1441413006.
- Kiusalaas, J. (2014). *Numerical Methods in Engineering with Python*. Cambridge University Press, 2.ed. edition.
- Machine, W. (2006). Speed comparison of various number crunching packages.
- MathWorks (1994). Matlab documentation. <https://www.mathworks.com/products/matlab.html>. The MathWorks, Natick, MA, USA.
- Mooshak (2016). System for managing programming contests. <https://mooshak.dcc.fc.up.pt/>.
- PC2 (2016). Programming contest control system. <https://pc2.ecs.csus.edu/>.
- Pinçon, B. (2007). Quelques tests de rapidité entre différents logiciels matriciels. <http://cermics.enpc.fr/~jpc/scilab-gtk-tiddly/bench.pdf>.
- Quarteroni, A. and Saleri, F. (2007). *Cálculo Científico com MATLAB e Octave*. Springer, 1.ed. edition.
- Sharma, N. and Gobbert, M. K. (2010). A Comparative Evaluation of Matlab, Octave, FreeMat, and Scilab for Research and Teaching. Technical report, Department of Mathematics and Statistics, University of Maryland, Baltimore County.
- SIAM (2019). The history of numerical analysis and scientific computing. <http://history.siam.org/>. Acessado: 10/06/2019.
- SuSy. “submission and testing system for student programs”. <https://www.ic.unicamp.br/~susy/>.