

## OTIMIZAÇÃO DE UM MOTOR DE FOGUETE DE PROPELENTE SÓLIDO VIA SIMULAÇÕES NUMÉRICAS

Marlon Sproesser Mathias, [marlonsmathias@gmail.com](mailto:marlonsmathias@gmail.com)  
Marcos Vinícius Fernandes Ribeiro, [ribeiro.mvf@gmail.com](mailto:ribeiro.mvf@gmail.com)  
Paulo Celso Greco Júnior, [pgreco@sc.usp.br](mailto:pgreco@sc.usp.br)

Escola de Engenharia de São Carlos - USP, Avenida João Dagnone, 1100 - CEP 13563-120 - São Carlos, SP

**Resumo:** Este trabalho consiste na simulação da operação de um motor de foguete a propelente sólido para otimização de performance. A otimização da performance de um motor foguete é de fundamental importância para o sucesso de um veículo espacial devido à grande quantidade de propelente envolvida, que pode superar 80% do peso total de lançamento. A partir de uma configuração inicial, definida pelo operador, o programa calcula toda a operação do motor, modificando os parâmetros do bocal até que se atinja uma performance ótima, sem que se ultrapassem os limites de segurança de operação ou limitações impostas pelo projeto. São utilizadas discretizações temporais para que se obtenha as características do motor durante todo o seu funcionamento. O motor é, então, otimizado para diferentes instantes e simulado novamente a cada variação, até que se encontre o que fornece a melhor performance. Inicialmente o programa foi executado em Visual Basic, sendo posteriormente escrito em uma combinação de programas nas linguagens Python e Fortran, reduzindo o custo computacional das simulações, além de permitir o uso da tecnologia Multithreading. Por fim, pode-se obter um bocal otimizado para qualquer altitude de operação, em qualquer faixa de empuxo e dimensão, permitindo o uso do programa tanto para pequenos motores, como para grandes veículos lançadores.

**Palavras-chave:** Otimização, simulação computacional, propelente sólido, bocal, multithreading

### 1. INTRODUÇÃO

Um motor de foguete que queima combustível sólido apresenta vantagens e desvantagens. A principal vantagem é sua simplicidade, pois ele é composto basicamente por um sistema de ignição, uma câmara de combustão e um bocal. Após iniciada, não há como parar a reação até que todo o combustível seja queimado. O propelente é armazenado dentro da própria câmara de combustão, na forma de um ou mais grãos.

A taxa de queima depende da superfície exposta do combustível e da pressão interna da câmara. Durante o funcionamento do motor, os grãos vão sendo consumidos e sua geometria é alterada, o que, por sua vez, altera a taxa de queima e o volume livre dentro da câmara.

Devido a essa variação nas características do motor, é impossível projetar um bocal sólido que seja ideal durante todo o seu funcionamento. Existe a possibilidade de se usar bocais de geometria variável, que se ajustam às dimensões ideais a cada instante, mas estes são muito caros, complexos e sujeitos a falhar, além de representarem um aumento no peso do foguete, o que diminui sua carga útil.

Este trabalho tem como objetivo desenvolver um método para encontrar um bocal que seja adequado a todas as fases do funcionamento do motor. As equações disponíveis possibilitam a obtenção de um bocal ótimo para um único instante do funcionamento. São realizadas simulações do motor com seu bocal otimizado para diversos instantes e é escolhida a configuração que forneceu o melhor desempenho dentro dos limites estabelecidos pelas outras áreas do projeto, sendo a pressão interna do motor a principal limitação.

Cada simulação é feita por meio de uma discretização temporal, as variáveis relevantes são calculadas para cada instante usando-se como base o instante anterior, o processo é repetido até que a queima se conclua e o motor pare de produzir empuxo. Os dados de cada simulação são armazenados pelo programa de otimização e usados para a escolha das próximas configurações a serem testadas.

Por fim, a partir das dimensões da câmara de combustão e dos grãos, das características do combustível, da pressão externa e da máxima pressão interna admissível, o programa retorna os raios da garganta e da saída do melhor bocal para essa determinada missão, além de um gráfico de empuxo e pressão versus tempo para o motor.

## 2. METODOLOGIA

Após o usuário inserir os dados do motor e dar o comando para iniciar o processo de otimização, o programa calcula uma geometria inicial para o bocal, que será refinada até que se chegue à performance desejada. São escolhidos alguns instantes para os quais o bocal será otimizado, variando desde o início da queima até sua finalização. O funcionamento motor é, então, simulado com esses bocais e os intervalos com melhor desempenho são selecionados. Novos instantes são escolhidos dentro desses intervalos e o processo é repetido, até que se chegue à precisão desejada.

Inicialmente foi feita uma planilha eletrônica no programa *Microsoft Excel 2010* e uma série de scripts em *Visual Basic for Applications* para testar a metodologia usada, uma vez que assim é possível acompanhar cada etapa dos cálculos a fim de conferir os resultados. Com a planilha pronta e seu resultado coerente com o esperado, foi iniciado o desenvolvimento do programa final, com linguagens de programação que acelerem sua execução e reduzam o custo computacional da otimização, possibilitando o teste de mais configurações e tornando o resultado final mais preciso.

Para o desenvolvimento do programa final são usados o sistema operacional *Ubuntu 10.10 "Lucid Lynx" 64-bit*, a linguagem *Fortran 95* com o compilador *GNU Fortran 4.4* e a linguagem *Python 2.6* com o módulo *EasyGUI 0.95*.

O programa de otimização se divide em quatro partes: simulação de queima, otimização para um determinado instante, escolha do melhor instante para o qual o motor deve ser otimizado e interface gráfica para operação do programa.

### 2.1. Simulação de Queima

Esta seção foi escrita em *Fortran 95*, sua alta velocidade de execução se tornou um fator determinante em sua escolha, pois em cada simulação podem ser necessárias milhares de iterações.

Um arquivo de entrada com todos os dados do modelo a ser simulado é lido e o modelo é preparado.

A velocidade de queima do grão é uma função da pressão interna da câmara, dada pela equação 1 (Nakka, 1984):

$$r = aP^n \quad (1)$$

Onde  $r$  é a taxa de queima,  $P$  é a pressão interna e  $a$  e  $n$  são constantes determinadas empiricamente que dependem do combustível usado. Com a taxa de queima definida, pode-se calcular quanto do grão será consumido a cada intervalo de tempo e qual será sua geometria após esse intervalo. Medindo-se a diferença de volume causada no grão, e sabendo sua densidade, calcula-se quanto gás foi liberado na câmara nesse instante. Com essa liberação de gás, a pressão interna começa a subir e parte do gás começa a sair pelo bocal.

Neste modelo todos escoamentos são considerados isentrópicos.

No bocal convergente-divergente usado, o gás, em fluxo subsônico, é acelerado em uma seção convergente até a velocidade sônica, que é atingida na garganta, a seção mais estreita, e então continua a acelerar em uma seção divergente, já em regime supersônico, como mostra a equação (Anderson, 2003):

$$\left(\frac{A}{A^*}\right)^2 = \frac{1}{M^2} \left[ \frac{2}{k+1} \left( 1 + \frac{k-1}{2} * M^2 \right) \right]^{\frac{k+1}{k-1}} \quad (2)$$

Na qual  $A$  é a área de uma dada seção,  $A^*$  é a área da garganta,  $M$  é o número de Mach e  $k$  é o coeficiente de expansão adiabática.

A pressão em um dado ponto do bocal, em função do número de Mach é dada por:

$$\frac{P}{P_0} = \left( 1 + \frac{k-1}{2} * M^2 \right)^{\frac{-k}{k-1}} \quad (3)$$

Onde  $P_0$  é a pressão quando  $M=0$ . Isso mostra que para que se tenha um gradiente de pressão favorável na parte divergente do bocal, ou seja,  $P^* > P_{externa}$ , é preciso de uma pressão na câmara maior que uma dada pressão crítica:

$$P_{crítica} = P_{externa} * \left( 1 + \frac{k-1}{2} \right)^{\frac{-k}{k-1}} \quad (4)$$

A partir dessa pressão, o bocal está engasgado, ou seja, o escoamento na garganta ocorre na velocidade do som. Com o bocal engasgado, a taxa de saída de gás é dada por:

$$Q = P * A^* * \sqrt{\left( \frac{M_{Mol}}{R * T_0} * k \right) * \left( \frac{2}{k+1} \right)^{\frac{k+1}{2*(k-1)}}} \quad (5)$$

Onde  $R$  é a constante universal dos gases,  $T_0$  é a temperatura da queima e  $M_{Mol}$  é a massa molar do gás. Com a quantidade de gás que é gerada na câmara e a quantidade que sai a cada intervalo de tempo, é possível calcular a quantidade acumulada e a pressão interna durante todo o funcionamento.

O próximo passo é calcular o empuxo gerado. Com o número de Mach na saída, calculado a partir de sua área, a temperatura do gás nesse ponto é dada por:

$$T_{saída} = T_0 * \left(1 + \frac{k-1}{2} * M_{saída}^2\right)^{-1} \quad (6)$$

Com a temperatura definida, calcula-se a velocidade do som nesse ponto e, consequentemente, a velocidade de exaustão do gás.

O empuxo em um determinado momento é dado por:

$$F = (Q_{sai} * V_{saída} + A_{saída} * (P_{saída} - P_{externa})) * E \quad (7)$$

O empuxo máximo é obtido quando  $P_{saída} = P_{externa}$ . Também foi levado em conta um fator de eficiência do bocal que, em bocais cônicos, tem um valor próximo a 0,8.

Integrando-se o empuxo ao longo do tempo de funcionamento, obtém-se o impulso gerado pelo motor.

O programa retorna o impulso gerado e a pressão máxima na câmara. Assim como um gráfico de empuxo e pressão ao longo do tempo.

## 2.2. Otimização Para Um Determinado Instante

Esta seção, assim como as próximas, é escrita em *Python 2.6*, linguagem escolhida por ser bastante flexível e fácil de usar. Mesmo que seja de execução mais lenta, ela será responsável por apenas uma pequena parte do tempo final de execução, uma vez que a maior parte do tempo é tomada pela simulação.

O bocal é considerado ótimo para um determinado tempo quando dois fatos ocorrem. O primeiro é que a quantidade de gás gerada pela queima é igual à quantidade expelida pelo bocal. O segundo é que a pressão do gás na saída do bocal é igual à pressão externa, o que significa que o máximo empuxo é gerado.

A partir de uma dimensão inicial dada ao bocal, a simulação é realizada até o instante para o qual o bocal será otimizado. A pressão interna e a taxa de queima do combustível nesse momento são, então usadas para que se calcule o próximo bocal a ser testado. Ao se mudar a garganta do bocal, as características do motor naquele instante também serão alteradas, o que gera novas dimensões a serem usadas, tornando esse um processo iterativo.

Para acelerar a convergência do processo, após a segunda iteração, é usado um método de Newton-Raphson modificado. No método original, para se encontrar o valor de  $x$  para o qual  $f(x)=0$  é usada a seguinte iteração (Burden, 2004):

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} \quad (8)$$

Neste caso,  $f(x)$  é determinado como a diferença entre o raio da garganta atual e o raio da garganta calculado para a próxima iteração. E uma aproximação de  $f'(x)$  é calculada da forma:

$$f'(x_i) := \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}} \quad (9)$$

As equações usadas são as inversas das encontradas no programa da simulação. Para se calcular o raio da garganta, usa-se a equação 5, para se calcular o número de Mach na saída, usa-se a equação 3 de modo que a pressão na saída se iguale à pressão externa, então é usada a equação 2 para se calcular o raio da saída.

Quando as iterações convergem para um valor dentro de uma margem de erro, a simulação é rodada até o fim da queima com as dimensões encontradas e a performance do motor, armazenada.

## 2.3. Escolha Do Instante Para o Qual o Motor Será Otimizado

Esta seção é escrita em *Python 2.6*, devido à mesma razão que a anterior. Primeiramente, é calculada uma geometria inicial para o bocal, usando como base a pressão máxima que a câmara pode atingir e a superfície exposta do grão no instante inicial.

Então, o programa otimiza o bocal para diversos instantes, a fim de encontrar o máximo valor de  $t$  para o qual o bocal pode ser otimizado. Com o valor máximo encontrado, são testados valores intermediários, a cada iteração o intervalo de busca é refinado para valores próximos da melhor performance encontrada que não ultrapasse os limites do projeto.

Para medir-se a performance de uma configuração, é usado o impulso gerado, quanto maior o impulso, melhor a configuração. O limitante usado é a pressão máxima atingida na câmara, que não deve ultrapassar um valor pré-estabelecido. Podem ser escolhidos outros meios para medir-se a performance e incluir outras limitações ao projeto se for necessário à missão, por exemplo, um empuxo máximo que não deve ser ultrapassado no caso de foguetes que contenham uma carga frágil. Nesse caso, a configuração é considerada ótima quando gera o máximo de impulso possível, respeitando-se o limite da pressão máxima interna da câmara.

Observou-se que dois tipos de otimização podem ocorrer. Em um deles o motor é otimizado para o instante que gera o máximo impulso possível, sem que a pressão cause problemas. No outro, o limitante é a pressão atingida, nesse caso existem instantes que gerariam um impulso maior, mas estes ultrapassam a pressão máxima. Se ocorrer o segundo caso, o programa refina a busca em dois intervalos, logo antes e logo após a pressão máxima ser atingida.

Quando o intervalo de busca se torna menor que a margem de erro estabelecida, o programa retorna a geometria do bocal que gerou o melhor resultado e uma tabela com valores do empuxo e da pressão interna em função do tempo.

Este programa se beneficia da tecnologia de *Multithreading*, disponível em grande parte dos computadores modernos, possibilitando que múltiplas configurações sejam testadas concomitantemente.

## 2.4. Interface Gráfica

A interface gráfica é escrita em *Python 2.6* com o módulo *EasyGUI 0.95*. A facilidade no desenvolvimento de uma interface gráfica simples se tornou um fator determinante em sua escolha. A interface possui campos de entrada para as variáveis envolvidas, divididos em quatro categorias, propriedades do grão, propriedades da câmara de combustão, dados do combustível, e dados da simulação.

Nas características do grão são entradas suas dimensões, a quantidade de grãos e quais de suas superfícies estão inibidas, isto é, protegidas contra a queima.

Nas características da câmara são entradas suas dimensões, a pressão máxima que deve ser suportada durante o funcionamento e o coeficiente de eficiência do bocal.

Nos dados do combustível são entradas a densidade do grão, a proporção de propelente no grão, os coeficientes de expansão adiabática do gás gerado, tanto bifásico quanto da mistura, a massa molar do gás, a temperatura e a eficiência da queima e os coeficientes  $a$  e  $n$  da queima, que expressam sua taxa em função da pressão.

Nos dados da simulação são entradas o tamanho do intervalo de discretização temporal, a precisão desejada na simulação e o intervalo de tempo entre cada entrada na tabela de empuxo e pressão em função do tempo.

Os dados são salvos em um arquivo de entrada que é usado pelo programa que faz a otimização.

## 2.5. Funcionamento Geral do Programa

Quando os dados da otimização são inseridos na interface gráfica, é gerado um arquivo de texto que contém todas as informações necessárias para a simulação. Esse arquivo é usado como entrada para o programa principal, que escolhe o melhor instante para o qual o bocal deve ser otimizado. O programa, então calcula uma geometria inicial que será usada como base durante a otimização.

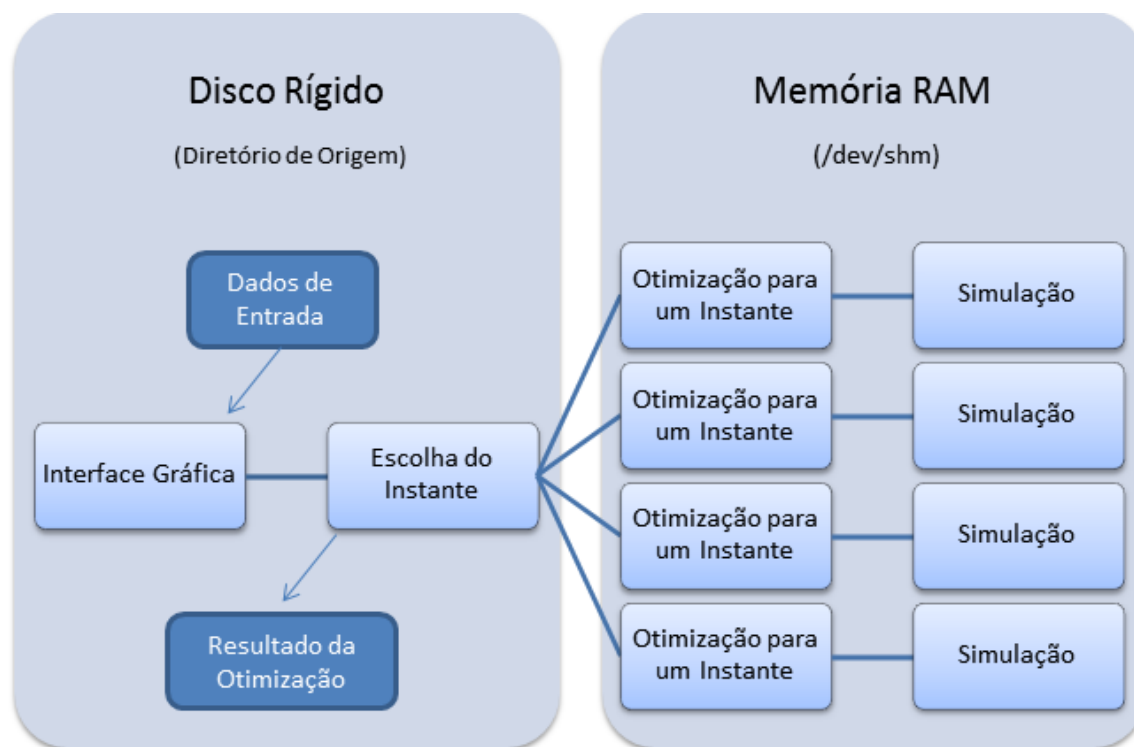
É criado um diretório temporário na *RAM* do computador, contendo um subdiretório para cada núcleo disponível para o processamento. Cada subdiretório possui uma cópia do programa que otimiza o bocal para um determinado instante e uma cópia do programa que realiza a simulação. Para isso é usado o diretório */dev/shm*, que conta com o sistema de arquivos *tmpfs* que armazena suas informações na memória virtual do sistema, aumentando a velocidade de leitura e gravação.

É, então, iniciado o processo iterativo para a escolha da melhor geometria para o bocal. Arquivos de entrada já contendo a geometria inicial do bocal e um valor para o instante ótimo são colocados em cada subdiretório e a otimização para cada instante é realizada.

Ao fim de cada otimização para um único instante, as informações são retornadas ao programa principal, que escolhe os próximos instantes a serem testados.

Quando todo o processo é finalizado, os resultados são salvos em dois arquivos. Um, mostrando todos os instantes para os quais o bocal foi otimizado e suas características, o outro, mostrando as características do instante que foi escolhido como o melhor e o resultado completo da simulação realizada.

A figura 1 mostra um esquema do funcionamento do programa em um computador com quatro núcleos disponíveis.



**Figura 1. Esquema da organização do programa em um computador com quatro núcleos**

### 3. RESULTADOS

O programa foi executado para um motor com uma câmara de combustão cilíndrica de 43cm de comprimento, e 2,2cm de raio, que suporta uma pressão interna máxima de 4MPa. Na câmara foram colocados quatro grãos cilíndricos de 9cm de comprimento e 2,2cm de raio, com furos cilíndricos de 1 cm de raio, inibidos na superfície externa. A eficiência considerada para o bocal foi de 80%, típica de bocais cônicos.

Os grãos são compostos de KNSU, uma mistura de nitrato de potássio com açúcar, cujas características são descritas na tabela 1.

**Tabela 1. Dados do combustível KNSU e do gás produzido**

Densidade teórica [Kg/m <sup>3</sup> ]	1889
Razão de densidade composta pelo combustível	0.92
Temperatura de queima [K]	1720
Eficiência considerada da queima	0.92
Coefficiente $a$ [MPa <sup>-n</sup> .m/s]	8.263
Coefficiente $n$ (para P em MPa)	0.319
Massa molar do gás [g/Mol]	41.98
$k$ bifásico do gás	1.044
$k$ da mistura do gás	1.133

As simulações foram rodadas com intervalos de discretização temporal de 0,0005s e a precisão buscada foi de 0,001s. Os dados da simulação foram salvos a cada 100 iterações.

O processo de otimização retornou que o melhor bocal para esse caso é otimizado para um tempo de 0,011s após o início da queima, o raio da garganta é 8,32mm e o raio da saída é 16,5mm.

Nessa configuração o motor gerou 990N.s de impulso e funcionou durante 1,05s, atingindo uma pressão máxima de 3,95MPa na câmara de combustão.

#### 3.1. Configurações Testadas

A figura 2 mostra os valores tentados para o tempo ótimo e os resultados obtidos. Observa-se que quanto maior a pressão suportada pelo motor, maior pode ser o impulso gerado por ele.

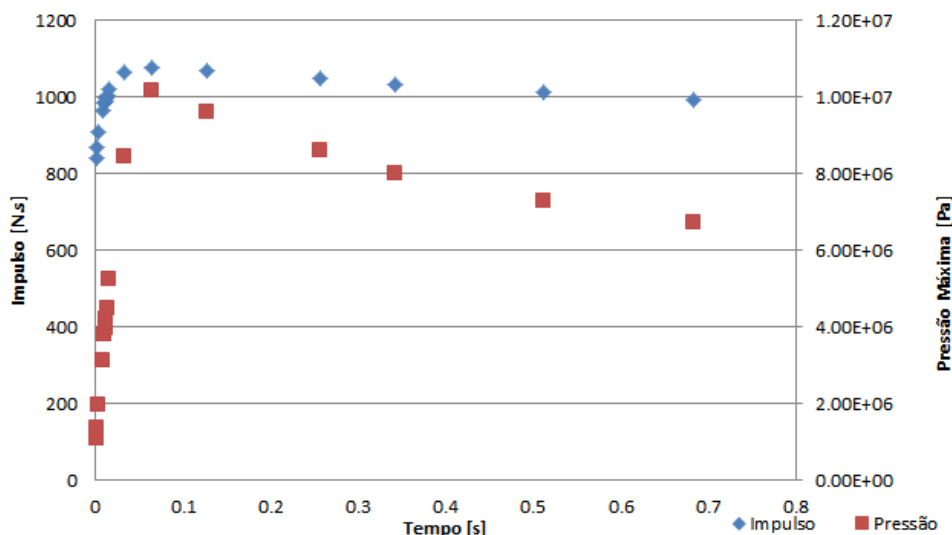


Figura 2. Impulso e pressão máxima a cada instante para o qual o bocal foi otimizado

O programa concentrou as tentativas próximas ao ponto onde a pressão máxima é atingida, até encontrar o maior impulso possível dada essa limitação, o que pode ser observado na figura 3, que mostra essa região com mais detalhes.

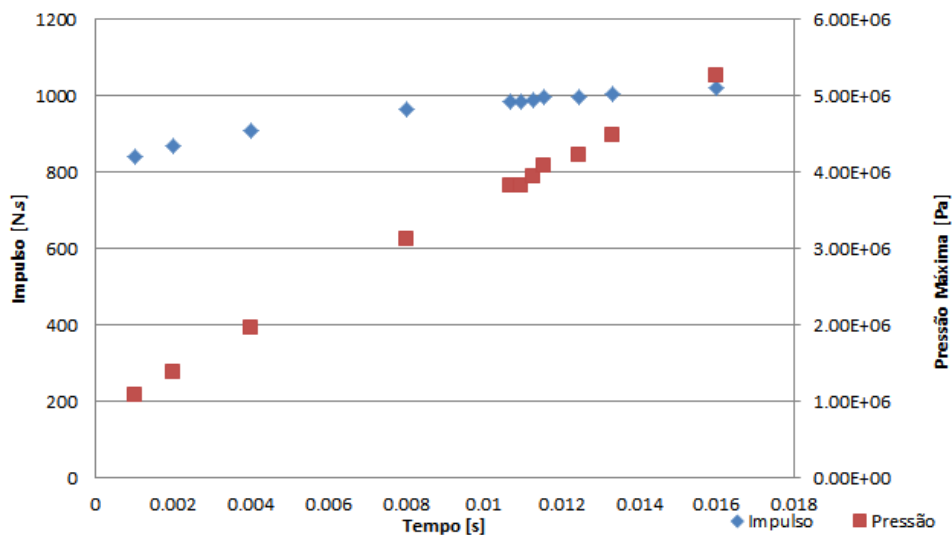


Figura 3. Impulso e pressão máxima para os instantes próximos ao limite do motor

### 3.2. Configuração Ótima

Na configuração ótima encontrada, o empuxo iniciou próximo a 600N e aumentou durante a queima para um máximo de 1310N. Esse comportamento se deve ao aumento da superfície exposta do grão ao longo do tempo, o que amplia a taxa de liberação de gás pelo motor.

O mesmo ocorre com a pressão interna do motor. Após um rápido aumento inicial até cerca de 2,2MPa nos primeiros cinco centésimos de segundo, a pressão passa a aumentar lentamente até chegar próxima aos 4MPa suportados pela câmara ao final da queima.

A figura 4 mostra os gráficos completos do empuxo e da pressão interna ao longo do funcionamento. Para a obtenção dos gráficos com uma maior resolução, o intervalo de gravação dos dados foi alterado para uma vez a cada cinco iterações.

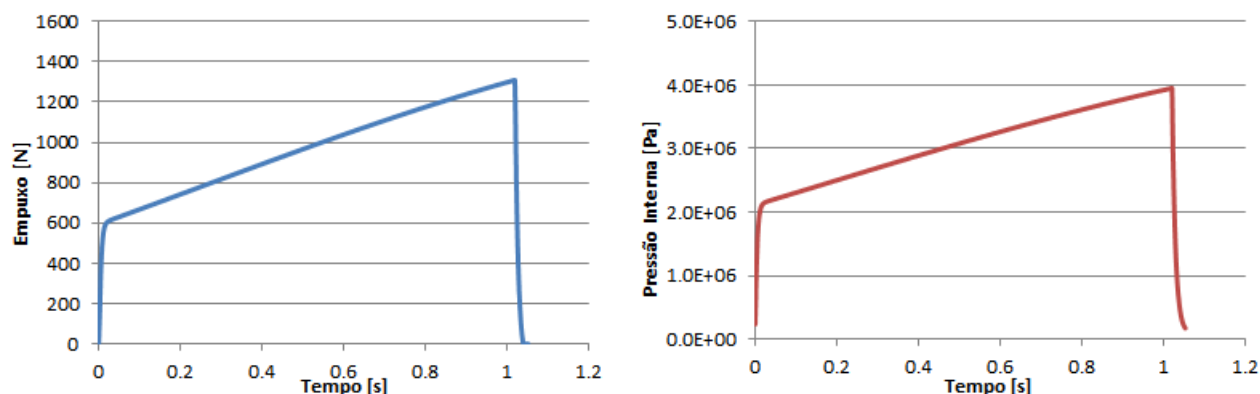


Figura 4. Empuxo e pressão interna ao longo do funcionamento do motor

### 3.3. Custo Computacional da Otimização

A otimização foi realizada no sistema operacional *Ubuntu 10.10 "Lucid Lynx" 64-bit* em um computador com processador *AMD Turion II M500* e 4Gb de *RAM DDR2-800 Dual-Channel*.

O processo foi repetido cinco vezes e levou um tempo médio de 2s com um desvio padrão de 0,1s. Uma melhora considerável quando comparado ao programa inicial desenvolvido no *Microsoft Excel 2010* com *Visual Basic for Applications* que levou, em média, 55s no mesmo computador, mas com o sistema operacional *Windows 7 Home Premium 64-bit*.

## 4. CONCLUSÕES

Foi possível o desenvolvimento de um programa para calcular um bocal ótimo dadas as características de um motor de foguete que tem um custo computacional baixo o suficiente para ser usado em estágios iniciais do projeto, possibilitando o teste de diversas configurações para a determinação da que melhor se adequa à missão. O programa é, também, flexível o suficiente para a introdução de novas medidas para a performance de um motor e novas limitações de projeto que devam ser respeitadas.

Com a determinação do melhor instante para o qual o motor deve ser otimizado, se torna possível o uso de programas mais complexos e mais precisos, mas também mais custosos, para a determinação da forma ideal para o bocal, já que são conhecidas as características que ele deve apresentar.

Para a validação do programa desenvolvido, pode ser usada uma célula de carga para medir-se o empuxo do motor na configuração gerada.

Também é possível o uso dos dados de empuxo do motor para se calcular com mais precisão a performance do foguete durante o lançamento.

## 5. AGRADECIMENTOS

Os autores gostariam de agradecer aos membros Grupo Topus de Pesquisas Aeroespaciais da Escola de Engenharia de São Carlos da USP, que possibilitaram e apoiaram a realização deste trabalho.

## 6. REFERÊNCIA

- Nakka, R.A. , 1984, "Solid Propellant Rocket Motor Design and Testing", University of Manitoba  
 Anderson Jr. , J.D. , 2003, "Modern Compressible Flow: With Historical Perspective", University of Maryland  
 Burden, R.L., Faires, J.D., 2004, "Numerical Analysis"  
 Nakka, R.A., 2011, Richard Nakka's Experimental Rocketry Web Site <<http://www.nakka-rocketry.net/>>  
 Python Software Foundation, 2011, Python v2.6.7 Documentation <<http://docs.python.org/release/2.6.7/>>  
 Free Software Foundation, 2008, The GNU Fortran Compiler <<http://gcc.gnu.org/onlinedocs/gcc-4.4.6/gfortran/>>  
 EasyGUI, 2010, EasyGUI 0.95 Documentation <<http://easygui.sourceforge.net/>>

## 7. DIREITOS AUTORAIS

Os autores são os únicos responsáveis pelo conteúdo do material impresso incluído no seu trabalho.

## SOLID FUEL ROCKET ENGINE OPTIMIZATION WITH NUMERICAL SIMULATION

Marlon Sproesser Mathias, marlonsmathias@gmail.com

Marcos Vinícius Fernandes Ribeiro, ribeiro.mvf@gmail.com

Paulo Celso Greco Júnior, pgreco@sc.usp.br

Escola de Engenharia de São Carlos - USP, Avenida João Dagnone, 1100 - CEP 13563-120 - São Carlos, SP

**Abstract:** *This paper consists in the simulation of a solid propellant rocket engine aiming an optimal performance. Optimizing a rocket engine is critical for the success of a space vehicle due to the large amount of fuel involved, which might exceed 80% of total launch weight. From an initial configuration defined by the user, the program changes the nozzle parameters until optimal performance is achieved, while respecting safety and project limitations. Temporal discretization is used in order to compute the engine data through all of its functioning. The engine is, then, optimized for different instants of the operation and simulated again, with every change, until optimum performance is found. Initially, the program was developed in Visual Basic, then was written as a combination of programs in Python and Fortran programming languages, reducing the computational cost of simulations and allowing use of multithreading technology. Finally, the optimum nozzle for every operation altitude, in any thrust range and at any scale can be obtained in order to allow the program's use for small engines, large launcher vehicles or any mission in between.*

**Keywords:** *Optimization, computational simulation, solid fuel, nozzle, multithreading*