



## INTEGRAÇÃO DE OTIMIZADORES A UMA FERRAMENTA DE SIMULAÇÃO

Edson Cordeiro do Valle<sup>1</sup>, Rafael de Pelegrini Soares<sup>1</sup>, Tiago Fiorenzano Finkler<sup>1</sup>

<sup>1</sup>GIMSCOP

Departamento de Engenharia Química, Universidade Federal do Rio Grande do Sul (UFRGS)  
R. Eng. Luis Englert, s/n. Campus Central. CEP: 90040-040 - Porto Alegre - RS - BRASIL,  
E-MAIL: edsoncv@enq.ufrgs.br

### **Palavras Chaves: Otimização, Simuladores de Processos, NLP.**

Em processos de indústrias químicas é necessário, muitas vezes, avaliar o comportamento de unidades em determinados cenários. Tal avaliação pode ser realizada diretamente com testes em planta ou através do uso de simuladores de processos químicos. O uso de simuladores, nestes casos, apresenta vantagens, pois não é necessário o consumo de matérias-primas para a realização de testes além de não comprometer a segurança ou poluir o meio ambiente.

O software EMSO constitui um simulador dinâmico genérico de sistemas, cuja principal aplicação tem sido a modelagem de processos e equipamentos da indústria química. Através da incorporação de um otimizador ao simulador é possível aumentar suas funcionalidades, entre elas estão:

- ? Realizar otimizações estacionárias que, no caso particular de processos químicos, permitirá determinar as condições operacionais ótimas para se atingir determinados objetivos econômicos, de segurança e ambientais.
- ? Realizar a estimação de parâmetros (cinéticos, de eficiência, etc.) dos modelos desenvolvidos, a partir da alimentação no simulador de dados de processos.
- ? Realizar a reconciliação de dados de processos, como vazões, composições e temperaturas.
- ? Realizar síntese de processos (como redes de trocadores de calor e massa), a partir da implementação de métodos de otimização que contemplem variáveis discretas e contínuas.

Neste contexto, o objetivo deste trabalho é a implementação de uma interface de comunicação do simulador EMSO com otimizadores. Inicialmente foi desenvolvida uma linguagem para expressar problemas de otimização. Em seguida, foi desenvolvida uma interface de comunicação do simulador em questão com o otimizador de código aberto *OPT++*. Após a implementação da interface, foram realizados testes, que apresentaram bons resultados relativos aos tempos computacionais para a resolução do problema.

## 1 INTRODUÇÃO

A simulação de processos químicos é, nos dias de hoje, uma ferramenta importante para a avaliação de condições operacionais nas indústrias. Através do uso de tais ferramentas é possível avaliar o comportamento de unidades em determinadas situações. O uso de simuladores nestes casos apresenta vantagens, pois, com estes,

não é necessário o consumo de matérias-primas para as realizações dos testes além de não comprometer a segurança e prejudicar o meio ambiente. O *software* EMSO constitui um simulador dinâmico genérico de sistemas, cuja principal aplicação tem sido a modelagem de processos e equipamentos da indústria química. Por se tratar de um simulador dinâmico, é possível acompanhar as variáveis de processo ao longo do



## Oktober Fórum 2005 – PPGEQ

tempo, sendo possível simular procedimentos de parada e partida de unidades assim como mudanças de pontos de operação. O EMSO possui diversas funcionalidades como a utilização de álgebra esparsa para a resolução de sistemas de equações algébrico-diferenciais, diferenciação automática entre outros.

Além do uso deste *software* para simulação de sistemas este ainda pode ser utilizado para outros fins. O projeto ALSOC (Ambiente Livre para a Simulação, Otimização e Controle) prevê a implementação de rotinas de comunicação do simulador com um otimizador para aumentar o número de funcionalidades deste, após a implementação de tais rotinas será possível:

- Realizar otimizações estacionárias, que no caso particular de processos químicos permitirá determinar as condições operacionais ótimas para se atingir determinados objetivos econômicos de segurança e ambientais.
- Realizar a estimação de parâmetros (cinéticos, de eficiência, etc.) dos modelos desenvolvidos, a partir da alimentação no simulador de dados de processos.
- Realizar a reconciliação de dados (como vazões, composições e temperaturas) de processos.
- Realizar síntese de processos (como redes de trocadores de calor e massa), a partir da implementação de métodos de otimização que contemplem variáveis discretas e contínuas.

## 2 MATERIAIS E MÉTODOS

A implementação da comunicação do otimizador com o simulador constituiu-se de 4 etapas:

- Consolidação da linguagem de otimização.
- Escolha do *solver* de otimização.
- Desenvolvimento e implementação da interface de comunicação entre o simulador e o otimizador.
- Testes.

### 2.1 Consolidação de Linguagem de Otimização

A linguagem de otimização inicialmente proposta foi consolidada após uma avaliação das linguagens

de otimização de softwares comerciais. Para exemplificar a linguagem de otimização foi apresentado o seguinte problema, apresentado em Hock e Schittkowski (1981).

$$\begin{aligned} \min_{x \in \mathbb{R}^4} \quad & x_1 x_4 (x_1 + x_2 + x_3) + x_3 \\ \text{s.t.} \quad & 1 \leq x_i \leq 5, \quad i = 1, \dots, 4 \\ & x_1 x_2 x_3 x_4 - 25 \geq 0 \\ & x_1^2 + x_2^2 + x_3^2 + x_4^2 - 40 = 0. \end{aligned} \quad (1)$$

Tal problema de otimização seria escrito, utilizando a linguagem de otimização do EMSO, da seguinte forma:

```
optimization hs71 |
VARIABLES
x1 as Real(Default=1, Lower=1, Upper=5);
x2 as Real(Default=5, Lower=1, Upper=5);
x3 as Real(Default=5, Lower=1, Upper=5);
x4 as Real(Default=1, Lower=1, Upper=5);

MINIMIZE
x1*x4*(x1+x2+x3) + x3;

EQUATIONS
x1*x2*x3*x4 > 25;

x1*x1 + x2*x2 + x3*x3 + x4*x4 = 40;

OPTIONS
NLPsolver = "opt_ems0";
outputLevel = "all";
relativeAccuracy = 1e-6;
end
```

Através do problema cima, é possível observar que a linguagem apresenta uma forma simples e compacta, facilitando a expressão deste tipo de problema pelo usuário.

### 2.2 Escolha do Solver de Otimização

Após a consolidação da linguagem, foi realizada a revisão bibliográfica dos métodos de otimização (NLP) com e sem restrição. E, seguida foram avaliados *solvers* de otimização comerciais e gratuitos disponíveis. Entre os *solvers* avaliados, foi optado pela utilização do pacote *OPT++* desenvolvido pelo *Sandia National Labs*, escolhido pelo fato de se tratar de *software* livre (Licença Lesser GNU Public Licence ou LGPL) desenvolvido em "C++" (a mesma linguagem utilizada no desenvolvimento do simulador) utilizando conceitos de programação orientada a objetos. Além disso, tal pacote contém métodos de otimização que suportam com restrições não-lineares de igualdade e desigualdade, muito comuns em problemas de engenharia química.



## Oktober Fórum 2005 – PPGEQ

### 2.3 Desenvolvimento e Implementação da Interface de Comunicação entre o Simulador e Otimizador

A implementação da interface de comunicação entre o simulador e otimizador foi realizada de forma a aproveitar as funcionalidades do primeiro em relação ao algoritmo de diferenciação automática (para o cálculo dos gradientes) e às rotinas de álgebra esparsa. Dessa maneira, cada vez que o otimizador necessita avaliar a função objetivo e os resíduos das restrições, assim como seus gradientes, este solicita ao simulador que tais equações sejam avaliadas para o cálculo de uma nova direção de busca ou um novo passo. Tal esquema de comunicação é apresentado na Figura 1.

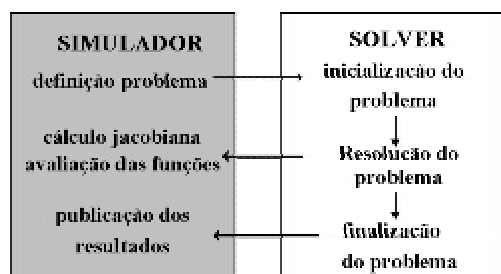


Figura 1: Esquema de comunicação do otimizador com o simulador.

Entre o simulador e o otimizador (*solver*) existem funções que “traduzem” os dados de um aplicativo para o outro para garantir que os mesmos sejam recebidos de forma correta. Este “tradutor” é denominado de interface de comunicação, que é constituída de três funções:

- Função *create*: É responsável pela inicialização dos parâmetros do otimizador como número de variáveis de decisão, função objetivo, restrições de igualdade e desigualdade, etc.
- Função *solve*: É responsável pela resolução do problema em si.
- Função *destroy*: É responsável pela liberação da memória alocada para a resolução do problema assim como a publicação dos resultados.

Como cada otimizador possui funções internas diferentes, as interfaces necessariamente também serão, de acordo com o otimizador utilizado. No caso do otimizador *OPT++*, as funções descritas acima foram implementadas para realizar a comunicação do mesmo com o EMSO.

### 2.4 Testes

Para avaliar a eficácia da implementação, foram realizados testes com problemas de otimização presentes na literatura, estes serão apresentados na seção seguinte.

## 3 RESULTADOS E DISCUSSÕES

Inicialmente foram realizados testes para comparar a eficácia da implementação realizada com problemas executados a partir de código compilado com e avaliação analítica dos gradientes e com código compilado e avaliação dos gradientes através de diferenças finitas. Para fins de comparação foi utilizado o problema 71 proposto por Hock e Schittkowsky (1981), apresentado na Equação 1, onde está presente uma restrição de igualdade linear e uma não-linear. Os resultados são apresentados através da Tabela 1.

Tabela 1: Comparação entre a resolução do problema utilizando a implementação realizada e a resolução com código compilado.

Solver	Tempo (s)*
EMSO e OPT++.	0.006
OPT++ com avaliação dos gradientes de forma analítica.	0.0059
OPT++ com avaliação dos gradientes por diferenças finitas.	0.0062

\*Média de 10 execuções

Através da Tabela 1, é possível observar que a resolução do problema utilizando o EMSO e *OPT++* apresentou resultados semelhantes ao problema resolvido utilizando o *OPT++* com o código compilado, isso indica, inicialmente, que o aproveitamento das rotinas internas do EMSO para a avaliação da função objetivo e restrições foram eficazes. Porém, para a comprovação de tal afirmação, é necessário repetir a mesma avaliação utilizando problemas que envolvam uma quantidade maior de variáveis de decisão e de restrições e, por consequência, realizem uma quantidade maior de avaliações de tais funções (função objetivo e restrições de igualdade e desigualdade). Cabe ressaltar que nos testes acima, todos os aplicativos chegaram a uma mesma solução.



# Oktober Fórum 2005 – PPGEQ

*10 Anos*

## 4 CONCLUSÕES

A implementação realizada apresentou resultados bons para o tipo de problema avaliado, porém, é necessária, a resolução de problemas maiores (mais variáveis de decisão e de restrições). Para fins de avaliação, também é necessário a comparação com aplicativos comerciais.

## 5 AGRADECIMENTOS

Os agradecimentos deste trabalho são destinados à FINEP pelo fomento financeiro e aos autores Rafael de Pelegrini Soares e Tiago Fiorenzani Finkler, integrantes do grupo de otimização do projeto ALSOC.

## REFERÊNCIAS

Hock W, Schittkowski K: Test Examples for Nonlinear Programming Codes, Lecture Notes in Economics and Mathematical Systems, Vol. 187, (1981), Springer.