

### 3 Formulação Matemática

Este capítulo contém a formulação matemática da técnica chamada *In Situ Adaptive Tabulation* – ISAT. Nele são apresentados os fundamentos do algoritmo original; a modificação proposta na forma como o algoritmo realiza a tabulação; as definições auxiliares necessárias ao entendimento dos resultados e a caracterização do ganho com respeito ao tempo gasto no processamento do algoritmo, utilizando a modificação proposta.

#### 3.1. Fundamentos da Técnica Original

A Tabulação adaptativa *in situ* – ISAT (*In Situ Adaptive Tabulation*) é uma técnica desenvolvida por Pope (1997) e implementada na PUC-Rio por Cunha (2010) para a redução de custos de problemas de alta dimensionalidade. Segundo Pope (1997), esta técnica mostrou-se 1000 vezes mais rápida do que a integração numérica direta das equações governantes (massa e energia). O aspecto mais importante é que não há restrições sobre o sistema físico para o qual a técnica pode ser aplicada, portanto, sua utilização é independente de hipóteses a respeito da estrutura da chama em simulações de combustão.

Considerando um escoamento reativo com  $n_s$  espécies químicas, o vetor de composição,  $\varphi$ , que representa o estado termoquímico deste escoamento, para uma determinada pressão  $p$  e entalpia específica  $h$ , é dado por:

$$\varphi \equiv \{h, p, Y_1, \dots, Y_{n_s}\}, \quad (3.1)$$

onde  $Y_i$  é a fração de massa da espécie  $i$  ( $i=1, \dots, n_s$ ).

O fluido no domínio de solução, reator, é representado por  $n_p$  partículas, cada qual com seu respectivo vetor  $\varphi$ , cujas composições evoluem de acordo com a reação e a mistura. A equação diferencial que descreve a evolução de um

conjunto de propriedades  $\varphi$  pode ser escrita, para cada partícula dentro do reator, do seguinte modo:

$$\frac{d\varphi(t)}{dt} = \mathbf{\Gamma}(\varphi, t) + \mathbf{S}(\varphi, t), \quad (3.2)$$

sendo  $\mathbf{\Gamma}$  a taxa de variação associada aos processos de transporte e  $\mathbf{S}$ , aquela associada às reações químicas, dada pela lei de Arrhenius.

O vetor de reação  $\mathbf{S}(\varphi, t)$  para cada partícula é dado pela taxa de produção das espécies:

$$\mathbf{S}(\varphi, t) = \left( 0, 0, \frac{\dot{\omega}_1 W_1}{\rho}, \dots, \frac{\dot{\omega}_{n_s} W_{n_s}}{\rho} \right), \quad (3.3)$$

onde, para cada espécie química,  $\dot{\omega}$  é a taxa de reação química líquida,  $W$  é a massa molar e  $\rho$  é densidade.

Para o caso de um modelo de reator do tipo PMSR (*Pairwise Mixing Stirred Reactor*), como usado pelo algoritmo ISAT, onde as partículas dentro do reator são organizadas aleatoriamente em pares  $(j1, j2)$ , o vetor de mistura,  $\mathbf{\Gamma}(t)$ , é dado por:

$$\begin{aligned} \mathbf{\Gamma}^{(j1)}(t) &= -\frac{\varphi^{(j1)} - \varphi^{(j2)}}{\tau_m}, \\ \mathbf{\Gamma}^{(j2)}(t) &= -\frac{\varphi^{(j2)} - \varphi^{(j1)}}{\tau_m}, \end{aligned} \quad (3.4)$$

onde  $(j=1, \dots, n_p)$  e  $\tau_m$  é o tempo característico de mistura

Nesse modelo de reator, a cada passo de tempo, acontecem os três eventos que o descrevem: entrada, saída e emparelhamento das partículas. Desse modo, a cada passo de tempo, os pares de partículas evoluem segundo:

$$\begin{aligned}\frac{d\varphi^{(j1)}}{dt} &= -\frac{\varphi^{(j1)} - \varphi^{(j2)}}{\tau_m} + \mathbf{S}(\varphi^{(j1)}, t), \\ \frac{d\varphi^{(j2)}}{dt} &= -\frac{\varphi^{(j2)} - \varphi^{(j1)}}{\tau_m} + \mathbf{S}(\varphi^{(j2)}, t),\end{aligned}\quad (3.5)$$

onde o índice  $j$  indica as partículas dentro do reator, ( $j=1, \dots, n_p$ ).

A integração no tempo da Eq.(3.2) a partir de uma composição inicial,  $\varphi_0$ , e um tempo inicial,  $t_0$ , fornece o mapa de reação para cada partícula dentro do reator:

$$\mathbf{R}(\varphi_0, t) \equiv \varphi(t) = \varphi_0 + \int_{t_0}^t \mathbf{\Gamma}(\varphi, t') dt' + \int_{t_0}^t \mathbf{S}(\varphi, t') dt', \quad (3.6)$$

Os dois termos do lado direito da Eq.(3.2) são tratados em passos de tempo diferentes. No passo de tempo de reação, a reação química evolui em uma entalpia e pressão fixas de acordo com a EDO:

$$\frac{d\varphi(t)}{dt} = \mathbf{S}(\varphi, t), \quad (3.7)$$

Dada uma composição de consulta  $\varphi^q$ , o ISAT é empregado na integração no tempo da Eq.(3.7), para cada partícula, a partir de uma composição inicial  $\varphi_0$ , a fim de reduzir o custo computacional associado à solução, a qual é uma aproximação linear para o mapa de reação,  $\mathbf{R}^l(\varphi^q, t)$ .

$$\mathbf{R}^l(\varphi^q, t) \equiv \mathbf{R}(\varphi_0, t) + \mathbf{A}(\varphi_0, t)(\varphi^q - \varphi_0), \quad (3.8)$$

onde a matriz gradiente  $\mathbf{A}$  é representada por:

$$A_{ij}(\varphi_0, t) \equiv \frac{\partial R_i}{\partial \varphi_{0j}}(\varphi_0, t), \quad (3.9)$$

O erro local para a aproximação do mapa de reação, feita pelo ISAT, é definido como norma euclidiana da diferença entre o valor real, calculado através da integração direta da equação, e o valor linearizado:

$$\varepsilon \equiv \|\mathbf{R}(\varphi^q, t) - \mathbf{R}^l(\varphi^q, t)\|, \quad (3.10)$$

De uma forma geral, a técnica funciona como segue: Os valores da composição inicial, o seu valor integrado no tempo e a matriz jacobiana correspondente são armazenados nas folhas de uma árvore binária, esta é a tabela ISAT, a qual é percorrida sempre que há uma busca pelos valores da integração das equações governantes. Nessa busca, a solução tabulada (armazenada em uma folha) é recuperada e usada, caso o erro da tabulação esteja dentro de uma tolerância, para uma extrapolação linear a partir das soluções vizinhas usando a matriz de sensibilidade. Esta aproximação tem seu erro controlado e possui uma acurácia de segunda ordem no tempo.

No instante inicial, a tabela ISAT encontra-se vazia e recebe como entrada, a tolerância de erro,  $\varepsilon_{tol}$ , o passo de tempo,  $\Delta t$ , e a composição a ser consultada,  $\varphi^q$ . Para o primeiro passo de tempo da reação, uma integração direta (DI) da equação diferencial ordinária que governa a evolução da composição de cada partícula no reator, Eq.(3.7), é feita a partir da composição inicial e fornece o valor exato do mapa de reação,  $\mathbf{R}(\varphi^q, t)$ .

Os valores obtidos são armazenados sequencialmente na tabela ISAT para uso futuro. Esta tabela, constituída por uma árvore binária de busca, é construída em tempo real conforme a simulação acontece. A primeira folha nesta árvore binária é, então, criada e contém as seguintes informações:

- composição inicial:  $\varphi_0 \equiv (h, p, Y_1, \dots, Y_{ns})$
- mapeamento da reação:  $\mathbf{R}(\varphi_0, t)$
- matriz de sensibilidade (gradiente):  $\mathbf{A}(\varphi_0, t)$
- matriz do elipsoide de acurácia:  $\mathbf{L}$

O elipsoide de acurácia (EOA) é o espaço elíptico em torno da composição inicial  $\varphi_0$ , onde a aproximação linear para o mapa da reação é possivelmente acurada para a tolerância especificada. O elipsoide é recalculado toda vez que tem seu tamanho aumentado, o que será explicado mais adiante.

O processo, que se repete para cada passo de tempo fornecido ao ISAT, é representado através de um fluxograma na Figura 3-1. A discussão a seguir acompanha as diferentes etapas deste fluxograma.

O próximo mapa de reação é calculado do seguinte modo: dada uma composição consultada,  $\varphi^q$  a árvore binária de busca é percorrida até encontrar uma composição inicial, que é próxima àquela consultada. Caso a aproximação feita para o ponto consultado esteja dentro do EOA, ou seja,  $\varepsilon \leq \varepsilon_{tol}$ , Eq.(3.10), admite-se que esta é suficientemente acurada, de forma a matriz gradiente é utilizada para realizar uma extrapolação linear, que fornece o mapa de reação,  $\mathbf{R}^l(\varphi^q, t)$  [Eq.(3.8)]; esta etapa é chamada de *recuperação*.

Caso contrário, a aproximação feita para o ponto consultado esteja fora do EOA, uma integração direta [Eq.(3.7)], que fornece  $\mathbf{R}$ , é realizada a partir do ponto consultado e a Eq.(3.10) é usada para calcular o erro,  $\varepsilon$ . Se este erro for menor do que a tolerância especificada,  $\varepsilon_{tol}$ , significa que  $\mathbf{R}^l$  é suficientemente acurado e então o EOA é aumentado, ainda com seu centro em  $\varphi_0$ , de forma a englobar tanto o ponto consultado,  $\varphi^q$ , quanto o elipsoide original; esta etapa é chamada de *crescimento*.

Se o erro for maior que a tolerância especificada, significa que  $\mathbf{R}^l$  não é acurado e o mapa de reação a ser usado deve ser  $\mathbf{R}$ . Caso o número máximo permitido de folhas na árvore binária não tenha sido atingido, uma nova folha é adicionada com as informações relativas ao ponto consultado, e o mapa de reação correspondente a esse ponto é retornado; esta etapa é chamada de *adição*.

Caso o erro local seja maior que a tolerância e o número máximo permitido de folhas na árvore tenha sido atingido, o valor do mapa de reação  $\mathbf{R}$ , obtido por integração direta, é retornado diretamente, sem adição de uma nova folha; esta etapa é chamada de *avaliação direta*.

A árvore binária com as informações do ISAT é composta de folhas e nós, cada nó está associado a um plano de corte definido por um vetor e um escalar, os quais definem o critério de busca da árvore. Quando uma nova folha é adicionada à árvore, a folha original se torna um nó com 2 folhas-filho, uma das quais é a folha original e a outra é aquela adicionada.

Nos instantes iniciais da simulação, espera-se que mais etapas de adições e crescimentos sejam realizadas. Depois, à medida que a tabela é preenchida, as etapas de recuperação se tornam mais frequentes que as demais. É isso que garante a eficiência do algoritmo (ver seção 3.2.3) quando comparada com DI, considerando que as adições são três ordens de grandeza e os crescimentos, duas

ordens de grandeza mais demorados do que as recuperações, como será apresentado na seção 3.3.3.

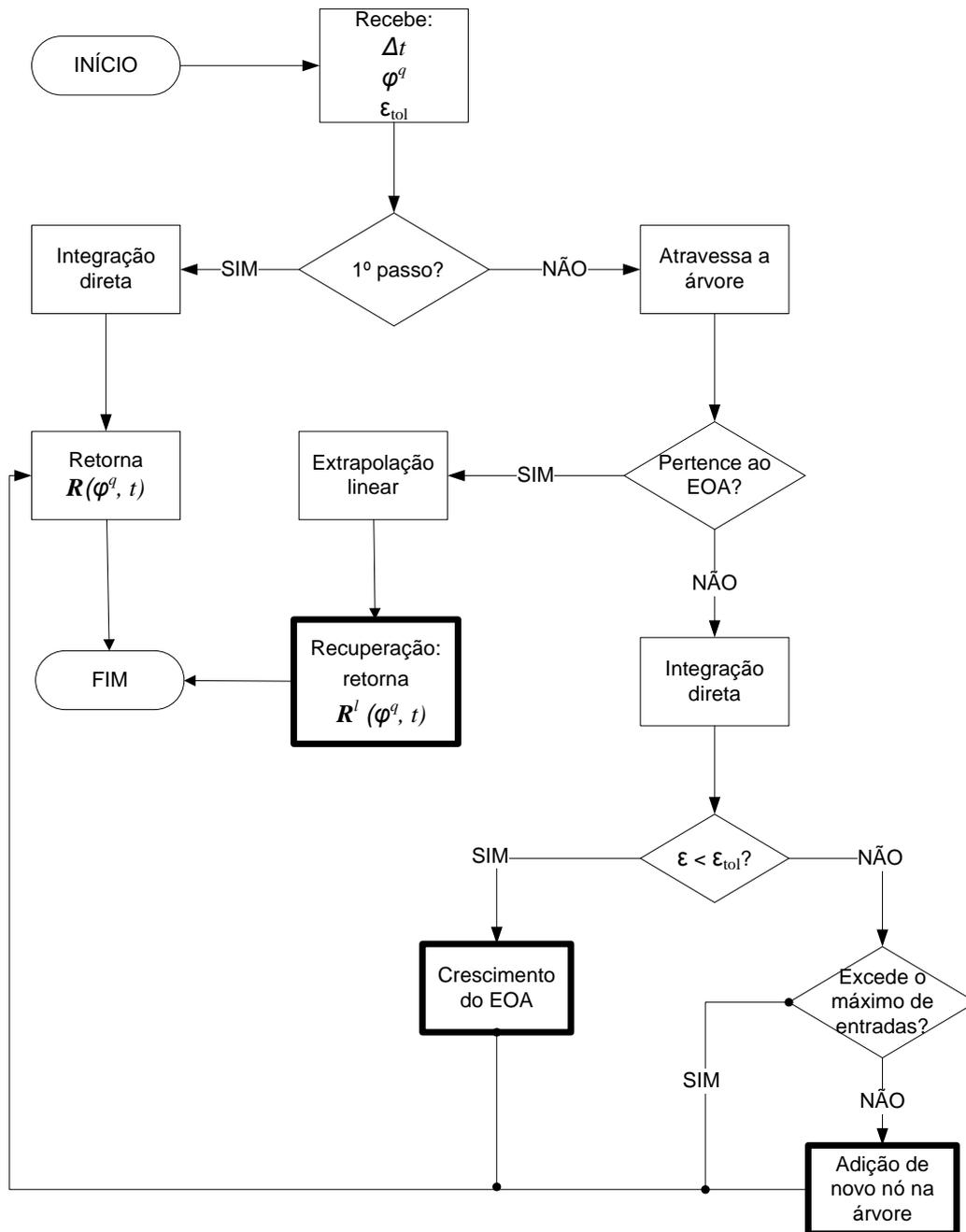


Figura 3-1. Fluxograma do algoritmo ISAT (adaptado de (Cunha, 2010)).

### 3.2. Definições Auxiliares

Nessa seção, apresentam-se as definições auxiliares. Para caracterizar as respostas do algoritmo ISAT, é importante que algumas definições sejam introduzidas.

O tempo adimensional é dado por:

$$\tau^* \equiv \frac{t}{\tau_r}, \quad (3.11)$$

onde  $\tau_r = \frac{\rho V}{\dot{m}}$  é o tempo de residência no reator, sendo  $V$  seu volume e  $\dot{m}$ , a vazão mássica e  $\rho$  a densidade da mistura.

A temperatura reduzida é dada por:

$$T^* \equiv \frac{T - T_{in}}{T_{eq} - T_{in}}, \quad (3.12)$$

onde  $T_{eq}$  corresponde à temperatura de equilíbrio associada à composição de entrada do reator e  $T_{in}$ , à temperatura dos gases de entrada no reator.

### 3.2.1. Medidas de Erro

Considerando  $\psi$  uma propriedade qualquer, o erro médio relativo dessa propriedade, em um intervalo de tempo  $\Delta\tau$ , é definido como:

$$\langle \varepsilon_{r,\psi} \rangle_M \equiv \frac{1}{\Delta\tau} \int_t^{t+\Delta\tau} \varepsilon_{r,\psi}(t') dt', \quad (3.13)$$

onde  $\varepsilon_{r,\psi}$  é o erro local relativo, definido como:

$$\varepsilon_{r,\psi} \equiv \frac{|\psi_{DI}(t) - \psi_{ISAT}(t)|}{\psi_{DI}(t)}, \quad (3.14)$$

e o erro global é definido como:

$$\varepsilon_g \equiv \frac{1}{\Delta\tau} \int_t^{t+\Delta\tau} \frac{\|\langle \varphi \rangle(t')_{DI} - \langle \varphi \rangle(t')_{ISAT}\|}{\|\langle \varphi \rangle(t')_{DI}\|} dt', \quad (3.15)$$

onde  $\langle \varphi \rangle$  denota o vetor de composição média, DI denota o cálculo usando integração direta e ISAT denota o uso do algoritmo.

Nos cálculos onde o ISAT usa a nova estratégia, o erro é calculado a partir do instante de tempo em que começa a tabular, ou seja, a partir de  $t_T$  (ver seção 3.3).

### 3.2.2. Medidas de Ganhos Com a Nova Estratégia

No presente trabalho, os resultados,  $r$ , obtidos com a nova estratégia (exposta na seção 3.3) são comparados com aqueles oriundos da versão original da técnica e os ganhos,  $g$ , são definidos como:

$$g = \frac{r_o - r_m}{r_o} \times 100\%, \quad (3.16)$$

onde  $r_o$  refere-se ao resultado utilizando a técnica original e  $r_m$ , ao resultado da técnica modificada.

A Eq. (3.16) pode compreender valores positivos ou negativos pertencentes ao intervalo que vai de -100% a +100%. Valores positivos indicam que a formulação modificada apresenta melhores resultados que a formulação original.

### 3.2.3. Considerações Quanto ao Custo Computacional

Para que o ISAT seja mais eficiente do que a avaliação direta das equações, é necessário que o tempo total gasto no uso da técnica ISAT seja menor, pois apenas desse modo seu uso é vantajoso. Considerando  $\tau G \approx \tau DI$  e  $\tau R \ll \tau DI$ , a condição necessária para a eficiência do ISAT, segundo Cunha (2010), é a seguinte:

$$\frac{nR}{nA} > \frac{\tau A}{\tau DI} - 1, \quad (3.17)$$

onde  $nR$  e  $nA$  são o número de recuperações e o número adições realizadas na árvore binária, respectivamente;  $\tau G, \tau DI, \tau R$  e  $\tau A$  são os tempos gastos com crescimento, integração direta, recuperação e adição, respectivamente.

Considerando que, para o caso da mistura de CO/O<sub>2</sub>, o tempo gasto na adição e na avaliação direta valem, aproximadamente, 10<sup>3</sup> μs e 10<sup>2</sup> μs, tem-se que o número de recuperações deve exceder o número de adições por um fator de 9, para garantir a eficiência da técnica

Outra questão bastante importante é o custo computacional em termos de uso de memória. A grande demanda de memória no uso do algoritmo deve-se principalmente à árvore binária de busca. O cálculo da memória gasta é feito como se segue:

Considerando o operador # (.) como o espaço ocupado por (.), tem-se que, para a biblioteca matemática utilizada, *Gnu Scientific Library* (GSL), o espaço ocupado por um vetor e uma matriz é dado como:

$$\begin{aligned} \#(\text{vetor}) &= 5 \times \#(\text{inteiro}) + n_{\varphi} \times \#(\text{double}), \\ \#(\text{matriz}) &= 6 \times \#(\text{inteiro}) + n_{\varphi}^2 \times \#(\text{double}), \end{aligned} \quad (3.18)$$

onde  $n_{\varphi}$  = dimensão do vetor de variáveis.

No entanto, pode-se considerar que o espaço ocupado por um inteiro e um *double* é dado por:

$$\begin{aligned} \#(\text{inteiro}) &= 4 \text{ bytes}, \\ \#(\text{double}) &= 8 \text{ bytes}. \end{aligned} \quad (3.19)$$

Na árvore binária, que é composta de nós e folhas, cada nó guarda 2 elementos e cada folha guarda 4 elementos. Esses dois componentes ocupam o seguinte espaço:

$$\begin{aligned} \#(\text{nó}) &= \#(\text{vetor}) + \#(\text{double}), \\ \#(\text{folha}) &= 2 \times \#(\text{vetor}) + 2 \times \#(\text{matriz}). \end{aligned} \quad (3.20)$$

Considerando  $n_F$  como o número de folhas e  $n_N$  como o número nós contidos na árvore binária de busca, a qual é referenciada como *abb*, tem-se que o espaço ocupado pela mesma é dado pela seguinte equação:

$$\#(\text{abb}) = n_F \times \#(\text{folha}) + n_N \times \#(\text{nó}). \quad (3.21)$$

Substituindo (3.18), (3.19) e (3.20) em (3.21):

$$\begin{aligned} \#(abb) = n_F \times [2 \times \#(vetor) + 2 \times \#(matriz)] \\ + n_N \times [\#(vetor) + \#(double)]. \end{aligned} \quad (3.22)$$

Logo, a memória gasta com a tabulação ISAT é dada pela equação final:

$$\begin{aligned} \#(abb) = n_F \times [88 + 16(n_\varphi + n_\varphi^2)] + \\ n_N \times [28 + 8n_\varphi]. \end{aligned} \quad (3.23)$$

### 3.3. ISAT Modificado

Esta seção apresenta o algoritmo modificado por uma nova estratégia de uso da tabulação. Como será discutida em detalhes, mais adiante, essa nova estratégia consiste em atrasar o início do processo de tabulação adaptativa durante o cálculo.

#### 3.3.1. Modificações no Código ISAT Existente

A primeira versão do código ISAT foi desenvolvida por (Cunha, 2010), este código da PUC-Rio, possui diversas bibliotecas usadas para implementar o algoritmo apresentado na seção 3.1. Tais bibliotecas possuem funções termoquímicas, funções que operam a árvore binária de busca, funções que operam o elipsoide de acurácia, funções para implementar as soluções das equações diferenciais ordinárias e funções para simular o modelo de reator PMSR.

No algoritmo original, a tabulação na árvore, que é iniciada simultaneamente com a integração da Eq.(3.2), começa sem qualquer folha na árvore binária de busca. Os registros ISAT, que são referidos como folhas de uma árvore binária, contém informações sobre os estados iniciais ( $\varphi_0$ ), o mapa de reação ( $\mathbf{R}^I$ ), a matriz de sensibilidade ( $\mathbf{A}$ ) e a matriz do elipsoide de acurácia ( $\mathbf{L}$ ). Como foi visto na seção 3.1, a técnica consiste, basicamente de três etapas: tentativa de recuperação do mapa de reação, tentativa de crescimento do elipsoide de acurácia e adição de uma nova folha na tabela ISAT, caso ela não esteja saturada.

Na formulação original tem-se que, durante a evolução inicial do processo termoquímico, antes que a solução de estado estacionário seja atingida, ocorrem os estados transientes da composição que não são necessariamente representativos da solução de regime estatisticamente estacionário, solução esta, de interesse na maioria dos casos práticos. Uma vez que, a priori, todos os estados que são realizados na simulação são tabulados na árvore ISAT, esses mapeamentos iniciais podem representar um desperdício de memória, uma vez que não seriam utilizados na solução em regime permanente, o que pode prejudicar o desempenho global do algoritmo. Em particular, no caso em que a tabela é preenchida por completo, ou seja, a árvore binária de busca é saturada, esta contém informações de um estado transiente, as quais não seriam úteis no processo de recuperação do estado termoquímico, uma vez que o regime estatisticamente estacionário for alcançado. Assim, pode ser bastante benéfico “limpar” a tabela ISAT.

De fato, este inconveniente é observado, por exemplo, quando a entalpia e a composição iniciais, predominantes dentro do reator, estão longe do estado de equilíbrio ou do estado de mistura de gases frescos. Nos casos em que o processo de mistura é lento, quando comparado com o tempo de residência, o processo transiente inicial é longo. Portanto, durante este tempo, há uma grande probabilidade de que as folhas que são adicionadas à árvore binária não sejam pertinentes à operação do reator em estado estacionário.

Como consequência, a árvore binária resultante é muito maior do que o necessário e, possivelmente, também levaria a recuperações relativamente imprecisas. Isto seria agravado em situações onde o número máximo de entradas na árvore é alcançado, que é o caso mais frequentemente observado nas simulações numéricas. Nesse caso, é possível que a árvore sature antes de alcançar o estado estacionário, obrigando o algoritmo a realizar mais avaliações diretas do que recuperações e diminuindo a eficiência do ISAT. Essa discussão aponta algumas debilidades do algoritmo original, que resultam em grande memória demandada e tempo de processamento.

Na tentativa de solucionar tais problemas encontrados na configuração original do ISAT, foi feita uma modificação na utilização do ISAT de modo que a tabulação na árvore binária de busca aconteça somente após o término do estado transiente inicial.

Esta alteração, introduzida no algoritmo ISAT durante o presente trabalho, inicia o algoritmo após um transiente, no qual apenas integrações diretas são realizadas. A maneira como a nova estratégia de tabulação, aqui proposta, está implementada, permite que se escolha o instante em que a tabulação inicia, ou seja, pode se fazer um estudo a respeito do tempo que o algoritmo permanece no regime transiente, realizando apenas integrações e escolher o melhor momento para que o algoritmo inicie a tabela ISAT.

Pretende-se, com isso, diminuir o esforço computacional exigido no armazenamento de dados, tendo em vista que as primeiras composições realizadas na simulação não estarão na tabela e, ainda, garantir que a tabela não esteja saturada antes de atingir o regime permanente, de forma que o algoritmo não perca eficiência. Uma vez que apenas integrações são feitas inicialmente, a nova estratégia de tabulação permite que se tenha maior acurácia nos resultados da simulação de processos de combustão, como será demonstrado no capítulo 4.

### **3.3.2. Dados de Entrada**

Os dados de entrada do algoritmo são divididos em dados do reator e dados do ISAT modificado e são apresentados juntamente com alguns cuidados necessários na escolha dos valores de cada um deles. Ao utilizar o algoritmo, esperam-se, como resposta, as propriedades termoquímicas do escoamento reativo para um dado mecanismo de cinética química, em função dos parâmetros abaixo discriminados:

#### **3.3.2.1. Reator**

Os dados de entradas referentes ao Reator são:

- Tempo de residência ( $\tau_r$ ): Caracteriza o tempo decorrente entre a entrada e a saída de uma partícula constituinte do reator.
- Tempo de mistura ( $\tau_m$ ): É a escala de tempo que caracteriza o processo de micro mistura.

- Tempo de emparelhamento ( $\tau_p$ ): O emparelhamento é um dos três tipos de eventos do reator. Define quantos pares de partículas irão se emparelhar e mudar de pares dentro do reator.

- Estado termodinâmico da mistura na entrada do reator e dos gases queimados que inicialmente preenchem o reator ( $p, T, Y$ ).

### 3.3.2.2. ISAT

Os dados de entradas referentes ao ISAT são:

- Número máximo de entradas na árvore binária ( $n_e$ ): Esse parâmetro limita a quantidades de folhas que podem ser adicionadas à árvore de busca do ISAT. Quando a tabela está lotada, todas as consultas que não podem ser recuperadas da mesma são diretamente integradas. Considerando que as recuperações são muito mais rápidas que as integrações diretas, grandes tabelas ISAT são mais eficazes que as pequenas. Portanto, deve-se definir o número máximo de entradas na tabela de forma que esta ocupe uma grande parte da memória disponível no computador, tendo o cuidado de não excedê-la.

- Tolerância do ISAT ( $\epsilon_{tol}$ ): Conforme a tolerância de erro do ISAT decresce, os requisitos de memória e tempo de CPU para a construção da tabela ISAT aumentam. A especificação de um erro menor do que a tolerância necessária para alcançar uma determinada acurácia global resulta em uma penalidade no desempenho do algoritmo. Esse parâmetro deve ser reduzido criteriosamente e gradualmente durante simulações exploratórias.

- Número de partículas no reator ( $n_p$ ): Esse parâmetro tem influência direta no tempo de processamento do programa e na acurácia dos resultados.

- Tempo para tabulação inicial ( $t_T$ ): Deve corresponder a uma fração do tempo gasto no período transiente da simulação. Há que se cuidar para que este valor não seja muito grande, o que aumentaria o tempo de CPU gasto na simulação. Para que a eficiência da nova estratégia seja garantida, deve-se buscar um equilíbrio entre o tempo dispendido com as integrações diretas, realizadas enquanto não há tabulação e o tempo de CPU total gasto na simulação.

- Tempo total de simulação ( $\tau^*$ ): É aconselhável que este seja bem maior que o tempo de residência do reator, para que se garanta o regime estacionário.

Quanto maior for este parâmetro, melhor será o ganho obtido com o uso do ISAT em termos de tempo de simulação, quando comparada com a integração direta das equações governantes.

- Passo de tempo ( $\Delta t$ ):  $\Delta t = \frac{n_{in}}{n_p} \tau_r$ , onde  $n_{in}$  é o número de partículas que entram no reator.

### 3.3.3. Saídas do ISAT

O programa ISAT fornece como saída diversas propriedades termoquímicas e informações de desempenho do algoritmo, dentre os quais se destacam a fração de massa das espécies químicas presentes na mistura reativa e a evolução da temperatura dentro do reator. O programa fornece, também, dados estatísticos para comparação entre a formulação original e a formulação modificada com a nova estratégia de tabulação. Esses dados são a evolução no tempo de cada uma das saídas características, isto é, o número de recuperações, adições e crescimentos na árvore binária, além das avaliações diretas feitas durante o processamento, a altura da árvore binária de busca e a derivada no tempo desses parâmetros de desempenho. Isto permite acompanhar a evolução da simulação.

As saídas do programa são apresentadas, sob forma gráfica, a partir da Figura 3-2 até a Figura 3-5, usando, como exemplo, o mecanismo de Gardiner (Gardiner, 2000), para a combustão do CO com O<sub>2</sub>, cujas espécies químicas consideradas são CO, O, CO<sub>2</sub> e O<sub>2</sub>. As condições iniciais e de contorno relativas aos resultados mostrados nestas figuras serão discutidas no capítulo 4.

O programa fornece, ainda, um arquivo de saída que possui os parâmetros de entrada do ISAT, do reator e da mistura que entra no mesmo; informações referentes ao mecanismo cinético utilizado; número total de adições, crescimentos, recuperações, avaliações diretas e seus tempos médios de processamento; número total de folhas e nós na árvore binária; altura dessa árvore e tempo total de processamento do programa. Os valores do erro global, máximo e médio das propriedades calculadas também são saídas do programa.

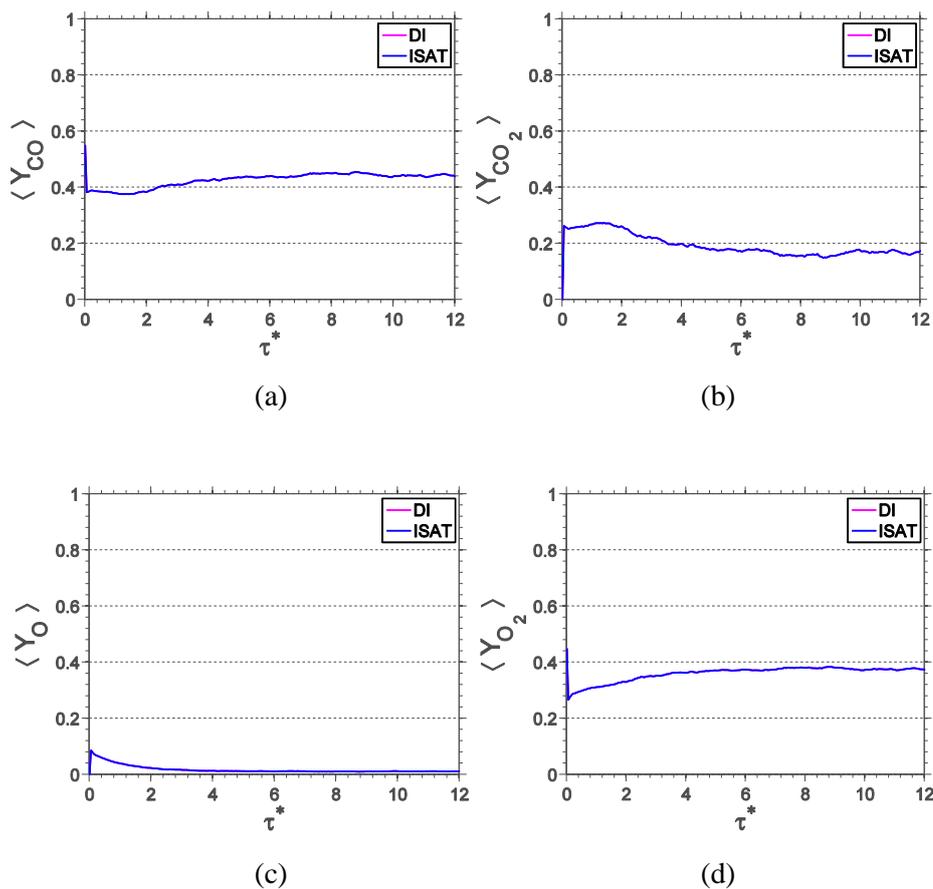


Figura 3-2. Evolução da medida da fração de massa das espécies químicas: (a) CO, (b) CO<sub>2</sub>, (c) O, (d) O<sub>2</sub>.

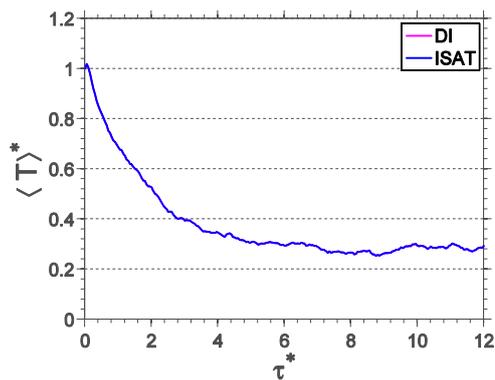


Figura 3-3. Evolução da temperatura média dentro do reator.

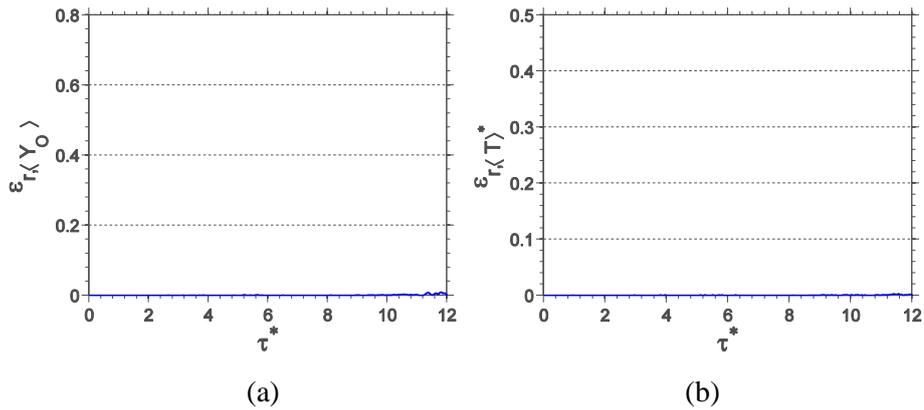


Figura 3-4. Evolução do erro médio: (a) Erro médio para o cálculo da fração de massa do O. (b) Erro médio para o cálculo da temperatura dentro do reator.

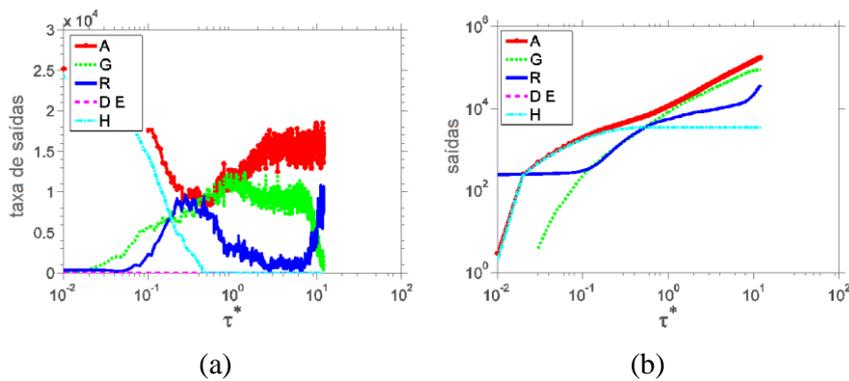


Figura 3-5. Gráficos de saídas características. A=número de adições; G=número de crescimentos; R= número de recuperações; DE= número de avaliações diretas; H=altura da árvore binária de busca. (a) Derivada no tempo das saídas características, (b) Saídas características.

A Figura 3-5 apresenta as saídas características do algoritmo original para um caso de combustão do CO. Observa-se do gráfico da esquerda que a altura da árvore binária é grande, da ordem de  $10^5$  e, como foi discutido nas seções anteriores, isso interfere negativamente no tempo de processamento do código. Outro problema que pode ser verificado nesta figura é que o número de recuperações não ultrapassa o número de adições, o que prejudica a eficiência da técnica. As modificações propostas na versão original do ISAT abordam estas questões e melhoram o desempenho do algoritmo, uma vez que gera árvores binárias menores e, como será visto no próximo capítulo, o número de

recuperações aumenta quando comparado com a versão original. Ou seja, espera-se que a nova estratégia de tabulação garanta maior eficiência ao código.

Outra informação importante também fornecida pelo código é o tempo médio de cada operação realizada pelo ISAT, medido durante as simulações e usando um processador Intel Core QUAD de 3,00 GHz. Esses valores empíricos, apresentados na Tabela 3-1, são quantidades médias considerando todas as simulações realizadas e são calculados como sendo a razão entre o tempo total gasto em cada tipo de saída característica e suas correspondentes quantidades.

Tabela 3-1. Tempos médios das operações características do ISAT, no caso da combustão CO/O<sub>2</sub>.

Saídas Características	Tempo (μs)
Adição	10 <sup>3</sup>
Crescimento	10 <sup>2</sup>
Recuperação	10 <sup>0</sup>
Avaliação Direta	10 <sup>2</sup>

### 3.3.4. Caracterização do Ganho Com Relação ao Tempo de Processamento

Para caracterizar o tempo de CPU,  $\tau_{CPU}$ , utilizado na execução do código, considera-se a seguinte aproximação:

$$\tau_{CPU} = \tau_{DI} + \tau_{ISAT} \cdot \quad (3.24)$$

A integração direta (DI) é realizada durante o período em que não há tabulação na árvore binária, ou seja, durante  $t_T$ . O tempo gasto com o ISAT,  $\tau_{ISAT}$ , refere-se ao período em que há tabulação, ou seja,  $(\tau^* - t_T)$ . Logo, tem-se que o tempo gasto para realizar uma integração direta e o tempo gasto no ISAT são obtidos, respectivamente, por:

$$\tau_{DI} = \$_{DI} Nt_T, \quad (3.25)$$

$$\tau_{ISAT} = \$_{ISAT} (N\tau^* - Nt_T), \quad (3.26)$$

onde:

$\$_{DI}$  é o custo por iteração para realizar DI,

$\$_{ISAT}$  é o custo por iteração para realizar ISAT,

$Nt_T$  é o número de iterações em  $t_T$ ,  $Nt_T \cdot \Delta t = t_T$

$N\tau^*$  é o número de iterações em  $\tau^*$ ,  $N\tau^* \cdot \Delta t = \tau^*$ .

Logo, tem-se que o tempo de CPU é dado por:

$$\tau_{CPU} = \$_{DI} Nt_T + \$_{ISAT} (N\tau^* - Nt_T). \quad (3.27)$$

Sabendo que

$$\$_{ISAT} = f(\$_{DI}, \$_A, \$_G, \$_R) = a\$_{DI} + b\$_A + c\$_G + d\$_R \quad (3.28)$$

onde  $\$_{DI}, \$_A, \$_G$  e  $\$_R$  correspondem ao custo por iteração de se realizar uma integração direta (DI), adição (A), crescimento (G) ou recuperação (R), na árvore binária dentro do algoritmo ISAT. Os coeficientes  $a, b, c$  e  $d$  são, respectivamente as porcentagens de DI, A, G e R:

$$a + b + c + d = 1, \quad (3.29)$$

Considerando os valores apresentados na Tabela 3-1, em particular para uma situação especial em que  $\$_{DI} = \$_G$  e utilizando as Equações (3.16) e (3.24) a (3.29), chega-se à seguinte relação para o ganho do tempo de CPU:

$$g_{\tau_{CPU}} = \frac{(9b - d)Nt_T}{(1 + 9b - d)N\tau^*}. \quad (3.30)$$

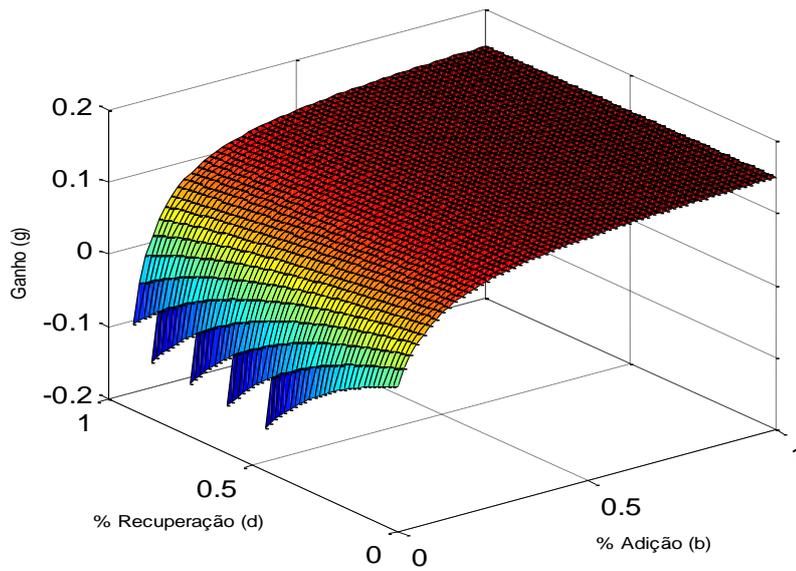
Definindo-se a razão entre o tempo de tabulação e o tempo total de simulação,

$$R_t = \frac{Nt_T}{N\tau^*} = \frac{t_T}{\tau^*}, \quad (3.31)$$

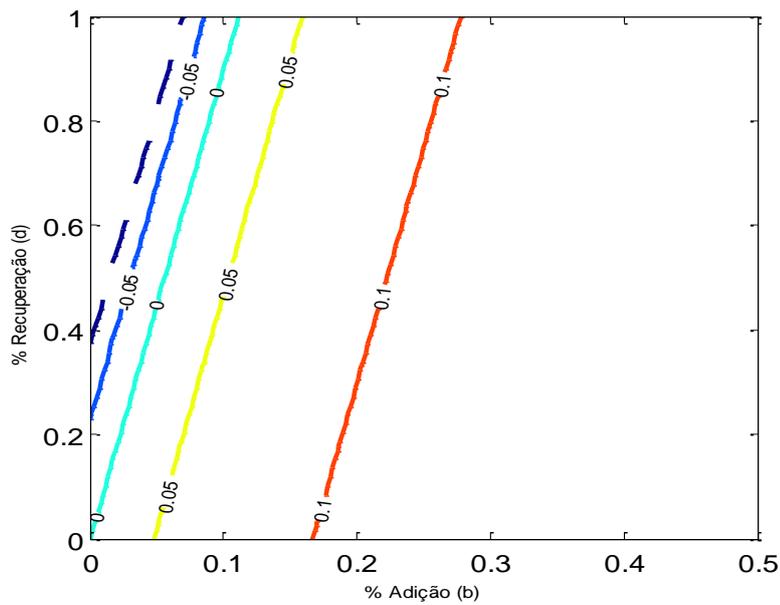
logo, o ganho em termos de tempo de CPU, para o caso especial, pode ser expresso como:

$$g_{\tau_{CPU}} = \frac{(9b-d)}{(1+9b-d)} R_t. \quad (3.32)$$

Para os casos estudados de combustão do CO com O<sub>2</sub>, admitindo  $R_t = 1/6$  e sabendo que  $g \in [-1,1]$ ,  $b \in [0,1]$  e  $d \in [0,1]$ , o ganho para o caso especial pode ser expresso graficamente, no intervalo considerado, tal como apresentado na Figura 3-6. Esta figura mostra apenas uma tendência para o ganho no tempo de CPU, considerando as aproximações feitas para esse caso em particular. Uma vez que o ganho depende da simulação realizada, isto é, dos valores de cada parcela da Eq. (3.28), a quantificação do ganho deve ser examinada caso a caso.



a) Curva do ganho para caso particular.



b) Isolinhas mostrando alguns valores de ganho.

Figura 3-6. Curva de ganho com relação ao tempo de CPU para caso particular.  $CO/O_2$  com  $R_f=1/6$ .