

Eduardo Hollauer

Departamento de Físico-Química - Instituto de Química - Universidade Federal Fluminense - Outeiro do Valquinho - Centro Niterói - 24210-150 - RJ

Recebido em 3/12/93; cópia revisada 4/4/94

It's well known the great impulse that direct SCF calculations has given with respect to the efficiency and performances of ab-initio calculations in extended molecular systems as well as in workstations with limited memories. In this work we study the effect of three methodological parameters on the global performance of the method. The two-electron organization file, the optimization of this procedure when used within the SCF-partially direct method, the checking towards the criteria evaluation and, finally, the efficiency of the parallel direct-SCF. The results suggest the use of super-matrixes, partially direct calculations with time criteria for the two-electron storage and parallel calculations with large number of processors.

Keywords: direct-SCF; partial storage; ab-initio; parallel direct-SCF.

## INTRODUÇÃO

Há cerca de 30 anos, cálculos auto-consistentes (SCF) vem sendo realizados de maneira rotineira em sistemas químicos do mais variado tipo, envolvendo moléculas em fase gasosa, agregados, macro-moléculas<sup>14</sup> e inclusive sólidos. Infelizmente, em muitas situações, os cálculos SCF têm sua eficiência tremendamente reduzida quando não se utiliza nenhum cuidado especial na adequação desta metodologia ao tipo de sistema estudado. Particularmente, quando trabalha-se com moléculas grandes, o grande número de integrais de dois elétrons geradas, a limitada viabilidade de acesso a grandes áreas de trabalho em disco e a crescente disparidade na eficiência dos processos ocorrendo em CPU, quando comparada com a performance de I/O, têm levado muitos autores a propor e efetuar várias modificações metodológicas e operativas no modelo SCF. Dentre estas mudanças destacaremos o uso de funções de base contraídas, potencial de carôço, uso intensivo de simetria, uso de memórias estendidas, cálculos em paralelo, entre outras aplicações de grande importância.

Para sistemas maiores, onde as integrais não podem ser guardadas em disco, ou naqueles casos onde este processo é muito caro quando comparado com o custo do cálculo em CPU, têm sido bem aceita a proposta de Almlof, Faegri e Korsell<sup>1,2</sup> de se calcular as integrais de dois elétrons a cada iteração e não se utilizar nenhum periférico para guardar as integrais de dois elétrons.

Este método, conhecido pelo nome de SCF-direto, vem acompanhado de um variado número de modificações que mantiveram o tempo de CPU apenas um pouco maior que o de cálculo regular SCF para um sistema molecular de grande porte. Pode-se, resumidamente, enquadrar estas modificações em quatro linhas gerais:

- Uso de uma razoável estimativa para os valores de integrais de dois elétrons pertencendo a um conjunto de integrais processadas em um "batch". Uma boa estimativa permite o descarte de conjuntos de integrais quando não existe a real necessidade de cálculo<sup>6</sup>.
- Guarda seletiva das integrais em disco, de modo a se minimizar o custo de cálculo<sup>6,7,8</sup>.
- Uso intensivo de simetria em todas as etapas, de modo a jamais se calcular qualquer integral relacionada por transformações de simetria do grupo de ponto do esqueleto nuclear<sup>9</sup>.
- Uso da matriz densidade diferencial na construção da matriz

de Fock<sup>5,6,12,13</sup> e posterior uso de técnicas extrapolativas para aceleração da etapa SCF.

Os resultados destas medidas e inovações restringiram o custo de maneira extraordinária, tanto em termos de CPU, quanto de I/O. Um cálculo SCF com 145 funções de base pode, nestas circunstâncias, consumir até 30% mais em tempo de CPU, porém com uma diminuição da ordem de 9 vezes<sup>6</sup> na quantidade de I/O realizada. Atualmente, muitos pesquisadores vêm se empenhando na expansão desta alternativa quanto à manipulação das integrais de dois elétrons a cálculos pós-SCF tais como o de cálculos perturbativos em segunda e quarta ordem (MP2, MP4), ou ainda em cálculos multiconfiguracionais baseados na expansão completa do espaço ativo (CAS), entre outras metodologias<sup>15</sup>.

Decidimos implementar a metodologia SCF-direto como parte do pacote de programas de cálculo molecular HONDO-7. Adotamos como metodologia incorporar todas as modificações sob forma de um pequeno arquivo, contendo as rotinas alteradas e aquelas que estendem a versão atual. Como os modernos programas de carregamento ("linkedition") fazem uso apenas das primeiras ocorrências de cada sub-rotina, pode-se introduzir novas opções e alterações às opções já existentes sem a necessidade de modificações no código original. Posteriormente, estas modificações foram incorporadas sob forma de segmentos permanentes do programa HONDO na versão 8.0.

Este trabalho procura estudar parâmetros da metodologia SCF-direto ligados a velocidade e eficiência de cálculo. Dentre estes parâmetros, estudamos a organização da lista de integrais de dois elétrons, a forma de seleção das integrais para o armazenamento parcial em disco e, por fim, a estratégia de paralelização em um contexto de processamento paralelo fracamente acoplado. Em todos estes casos, apresentamos "benchmarks" em regime de trabalho compartilhado e dedicado.

Discutiremos cada um destes itens em separado, para efeito de simplicidade na apresentação.

## TEORIA

De modo a se manter limitado o número de integrais calculadas, é de grande importância selecionar os conjuntos de integrais segundo a necessidade ou não de cálculo, a cada iteração. O procedimento básico para se explorar esta possibilidade, consiste em se representar as equações de Fock na forma de

diferenças nas matrizes de Fock e de densidades tal com apresentam as equações 1 e 2.

$$1) F^{(i)} = F^{(i-1)} + \Delta F^{(i)}$$

$$2) \Delta F_{ij}^{(i)} = \sum_{k,l} \Delta D_{kl}^{(i)} \left[ \frac{(ij/kl)}{2} - \frac{(ik/lj)}{2} - \frac{(il/jk)}{2} \right]$$

A construção do processo auto-consistente desta maneira é de grande vantagem, pois na medida em que o cálculo se aproxima da convergência, a matriz densidade diferencial vai se aproximando de zero, fornecendo uma maneira barata e eficiente de se selecionar os conjuntos de integrais (*batches*) a serem calculados. Neste caso, o produto do maior elemento da matriz densidade diferencial pela maior integral presente em cada um dos conjuntos é utilizado como critério de cálculo das integrais. Sendo este produto maior que um determinado valor (usualmente aceita-se 1.0D-09 como um bom valor), o programa calcula as integrais de dois elétrons associadas a este conjunto, atualizando-as, posteriormente, na matriz de Fock diferencial.

Deste modo, se  $i$  é uma função de base pertencendo à camada ("*shell*")  $I$ ,  $j$  pertencendo à "*shell*"  $J$ ,  $k$  à  $K$  e  $l$  à  $L$  pode-se calcular a maior contribuição à matriz de Fock por:

$$3) \text{CON}(I, J, K, L) = \max \{ \Delta D_{(I, J, K, L)} * \max \{ |(ij/kl)| \} \}$$

Nesta expressão, subentende-se que todas as posições da matriz densidade relacionadas com possíveis contribuições da matriz de Fock serão consultadas com relação à expressão 3. Os exatos elementos a serem consultados vão depender de como as integrais de dois elétrons são disponíveis, diferenciando-se para o caso de armazenamento canônico ou em forma de super-matriz. No caso particular de organização canônica, consultam-se os elementos de matriz apresentados na expressão 4 para todo  $i \in I, j \in J, k \in K$  e  $l \in L$ .

$$4) \Delta D_{(I, J, K, L)} = \max_{\substack{i \in I, j \in J, \\ k \in K, l \in L}}$$

$$\left\{ \Delta D_{ij}, \Delta D_{kl}, \frac{\Delta D_{ik}}{4}, \frac{\Delta D_{il}}{4}, \frac{\Delta D_{jk}}{4}, \frac{\Delta D_{jl}}{4} \right\}$$

A expressão 5 apresenta a desigualdade de Schwartz<sup>6</sup> utilizada para estimar-se, *a priori*, o valor máximo de cada integral no conjunto de integrais que compõe cada um dos *batches*.

$$5) \left\| \int \frac{\rho_1(1) \rho_2(2)}{r_{12}} dV_1 dV_2 \right\| \leq \left\| \int \frac{\rho_1(1) \rho_1(2)}{r_{12}} dV_1 dV_2 \right\| \left\| \int \frac{\rho_2(1) \rho_2(2)}{r_{12}} dV_1 dV_2 \right\|$$

O algoritmo 1 apresenta a forma de seleção das integrais, com base no máximo uso da simetria e na estimativa da máxima contribuição dada à matriz de Fock pelo conjunto de integrais (I,J,K,L). Neste sentido, cabe deixar claro que estas integrais são calculadas se, e somente se, o valor máximo para a contribuição da matriz de Fock é maior que um parâmetro controlado pelo usuário e geralmente igual a 1.0D-09.

Analisaremos, daqui por diante, o efeito do uso de supermatrizes no cálculo de integrais no processo SCF e como este uso se compara ao tradicional procedimento baseado na ordenação aleatória das integrais.

#### a) Forma de organização das integrais no SCF-direto

Uma situação não explorada em nenhum dos trabalhos anteriores é a do uso de supermatrizes para o cálculo das integrais e obtenção das soluções SCF-direto.

**Algoritmo nº 1.** Diagrama apresentando os principais comandos em linguagem FORTRAN para a rotina de cálculo de integrais e seleção dos "*batches*".

```

C
C começa o cálculo
C
DO 9000 I=1,NSHELL
.
. uso da simetria molecular
. IF ('princípio de procrastinação') GO TO 9000
.
DO 8000 J=1,I
.
. IF ('princípio de procrastinação') GO TO 8000
.
DO 7000 K=1,I
.
. IF ('princípio de procrastinação') GO TO 7000
.
DO 6000 L=1,LMAX
.
. IF ('princípio de procrastinação') GO TO 6000
.
CON = max {ΔD(I, J, K, L)} * max {(ij/kl)}
IF(DABS(CON).LT.CRIT) GO TO 6000
.
. processa as integrais e as contribuições à matriz de
. Fock.

6000 CONTINUE
7000 CONTINUE
8000 CONTINUE
9000 CONTINUE

```

Na ordenação aleatória, cada uma das integrais atômicas de dois elétrons é guardada em disco de maneira convencional, o que gera seis possibilidades para contribuições à matriz de Fock. O algoritmo 2 apresenta estas contribuições. Deve-se ter em conta que neste algoritmo as integrais aparecem normalizadas ao produto de três funções *delta*, controlando as

**Algoritmo nº 2.** Diagrama apresentando os principais comandos em linguagem FORTRAN para a acumulação dos elementos da matriz de Fock para o caso das integrais em forma canônica.

```

C
C
C lê um conjunto de integrais
C
DO 100 IN=1, NINT
.
. desempacota os índices
. reordena os índices de forma que I>J, K>L e (IJ)>(KL)
.
VAL=BUFFER (IN)
VAL = VAL / [(1 + δI, J) * (1 + δK, L) * (1 + δIJ, KL)]
.
FOCK(I,J) = FOCK(I,J) + 4 * VAL * D(K,L)
FOCK(K,L) = FOCK(K,L) + 4 * VAL * D(I,J)
FOCK(I,K) = FOCK(I,K) - VAL * D(J,L)
FOCK(I,L) = FOCK(I,L) - VAL * D(J,K)
FOCK(J,K) = FOCK(J,K) - VAL * D(I,L)
FOCK(J,L) = FOCK(J,L) - VAL * D(I,K)
.
.
100 CONTINUE

```

coincidências em  $i$  e  $j$ , em  $k$  e  $l$  e em  $(ij)$  e  $(kl)$ .

Na ordenação em forma de supermatriz, as integrais de dois elétrons são previamente contraídas de forma a definir duas integrais denominadas de contribuição de Coulomb e Troca, tal como mostra a equação 6.

$$6) [ij/kl]_c = (ij/kl) - \frac{1}{4} \{ (ik/jl) + (jk/il) \}$$

$$[ij/kl]_k = \frac{1}{4} \{ (ik/jl) - (jk/il) \}$$

Estas integrais, quando organizadas sob esta forma, possuem a vantagem de contribuir em apenas duas posições da matriz de Fock conforme é apresentado no algoritmo 3. Coerentemente apenas as duas posições da matriz densidade são testadas como mostra a equação 7.

$$7) \max \Delta D_{(I,J,K,L)}^{PK} = \max_{\substack{i \in I, \\ j \in J, \\ k \in K, \\ l \in L}} \{ \Delta D_{(ij)}, \Delta D_{(kl)} \}$$

A tabela 1 apresenta nossos resultados para uma série de sistemas moleculares com diferentes conjuntos de base, comparando o tempo de CPU para cálculos SCF direto em forma canônica e em supermatriz.

Desses resultados observa-se que, embora em pequenos sistemas estas duas formas apresentem velocidades comparáveis,

**Algoritmo nº 3.** Diagrama apresentando os principais comandos em linguagem FORTRAN para a acumulação dos elementos da matriz de Fock no caso de integrais sob a forma de supermatriz.

```

C
C
C
  lê um conjunto de integrais
DO 100 IN=1,NINT
  desempacota os índices
  reordena os índices de forma que (IJ)>(KL)
  VAL=BUFFER (IN)
  VAL = VAL / { 1 +  $\delta_{IJ, KL}$  }
  FOCK(I,J) = FOCK(I,J) + VAL * D(K,L)
  FOCK(K,L) = FOCK(K,L) + VAL * D(I,J)
100 CONTINUE
C
C
C
  
```

**Tabela 1.** Tempo de CPU para o SCF-direto (sem uso do disco) com várias moléculas e diferentes conjuntos de base. As integrais foram calculadas com subrotinas do programa GAUSSIAN para os conjuntos  $s$ - $p$  e as subrotinas do HONDO para os conjuntos contendo funções  $d$ . JK denota ordenação canônica enquanto PK a ordenação em forma de supermatriz.

molécula	base	tipo	tempo de CPU
Benzeno	DZV	JK	336
		PK	303
Benzeno	DZP	JK	1238
		PK	1226
Nitrobenzeno	STO	JK	234
		PK	226
Nitrobenzeno	6-31G**	JK	7718
		PK	7375
Valina	STO	JK	1935
		PK	1863

quando se trabalha com moléculas maiores (ou conjuntos de bases maiores) a forma em supermatriz gera tempos de computação ligeiramente menores, por valores que oscilam de 1 a 8%. Nos exemplos estudados não se utilizou qualquer forma de armazenamento em disco.

## b) Seleção das integrais para uso parcial do disco

Um ponto importante associado ao procedimento SCF-direto, diz respeito a possibilidade de se guardar um sub-conjunto das integrais de dois elétrons em disco. Este procedimento tem sido utilizado por alguns autores na literatura<sup>6</sup> e depende fundamentalmente da relação de custo entre o processo de leitura/escrita em disco e o cálculo destas integrais em memória. Naqueles sistemas onde a interface com o disco é rápido, este procedimento pode ser extremamente vantajoso, embora a tendência de crescente melhora na performance de CPU faça dessa possibilidade algo pouco provável em um futuro não muito longínquo.

Nos primeiros trabalhos<sup>6</sup> sobre esta idéia, procurou-se utilizar uma forma de seleção das integrais baseada em critérios estritamente ligados ao tempo de CPU gasto por cada "batch" de integrais. Um parâmetro de tempo, THIME, era controlado pelo usuário e regulava todas as integrais que seriam guardadas em disco ou calculadas novamente a cada ciclo auto-consistente. Se o tempo estimado supera o parâmetro de controle, este conjunto de integrais é guardado em disco e lido a cada ciclo de auto-consistência.

Em nosso trabalho apresentamos modificações de algoritmos prévios, onde se realiza um processo de seleção competitivo entre vários sub-conjuntos de integrais, de modo a se obter rendimento ótimo no uso de disco no contexto SCF-"parcialmente direto". Primeiramente discutiremos os resultados obtidos por um amplo conjunto de testes associados ao cálculo de cada tipo de integral realizados com o programa HONDO. O objetivo básico desta avaliação de custos é o de se definir, qualitativamente, quais as integrais cujo custo unitário é mais elevado, de modo a elaborar um processo de seleção que priorize o uso destas integrais em disco.

As tabelas 2 e 3 apresentam nossos resultados para o tempo de computação (CPU) ligado à parte de cálculo de integrais, para os conjuntos contendo bases  $s$  e  $p$  além daquelas envolvendo funções  $d$ . Procuramos utilizar conjuntos de base de tamanho e grau de contração bem diversificados, de modo a se obter uma amostragem representativa para estes tempos de computação.

Para esta estimativa, utilizamos a soma dos tempos de computação gastos em cada um das rotinas descritas na expressão 8 para cálculo das funções  $s$  e  $p$ , e as apresentadas na expressão

**Tabela 2.** Tempo de CPU, número total de integrais e tempo médio por integral para o nitro-benzeno, cálculos realizados com a simetria  $C_{2v}$ .

	DZP	6-31G**
tempo para as funções de base $s$ - $p$ (secs.)	496	141
número de integrais contendo $s$ - $p$ apenas (secs.)	$3,2 \cdot 10^6$	$3,0 \cdot 10^6$
tempo médio por integral $s$ e $p$ (sec/int).	$1,5 \cdot 10^{-4}$	$4,6 \cdot 10^{-5}$
tempo para as f. de base $d$ (secs.)	841,9	554,4
número de integrais contendo f. $d$	$14,75 \cdot 10^6$	$13,77 \cdot 10^6$
tempo médio por integral do tipo $d$ (secs.)	$5,7 \cdot 10^{-5}$	$4,0 \cdot 10^{-5}$



A estimativa de tempo adotada foi a proposta por Häser e Alhrich<sup>1</sup> e é apresentada na expressão 12, dependendo apenas do momento angular máximo associado a cada um dos quartetos ("shells") envolvidos.

$$12) t_i = 6 \text{ se } l_i l_j l_k l_L = 0 \\ 3 \text{ se } l_i l_j l_k l_L = 1 \\ 2 \text{ se } l_i l_j l_k l_L = 2 \\ 1 \text{ se } l_i l_j l_k l_L > 2$$

Denominaremos este critério por A sendo importante ressaltar que ele se torna especialmente útil em longos cálculos SCF, tipicamente encontrados em compostos inorgânicos, e/ou partindo de orbitais muito ruins. Neste caso, nossos resultados mostrarão ser este critério de seleção o que dá melhores resultados.

Um segundo tipo de seleção estudada é dada pela expressão 13, onde se realiza uma média sobre os valores absolutos de Schwartz,  $Q_{IJ}$ , para as integrais de dois elétrons, em cada um dos *batches*. Neste tipo de seleção, procura-se guardar em disco somente aquelas integrais de maior valor absoluto, baseado no fato de que esta integral será utilizada mais frequentemente durante o ciclo auto-consistente. Desta forma, convém guardar este tipo de integral em disco evitando-se assim recalculá-la a cada ciclo. Utiliza-se o critério expresso por 13 na estimativa do "custo" computacional para este registro de integrais.

$$13) \text{ Crit (IJ)} = \frac{\sum_{KL} |<ij/ij>|^{1/2} * |<kl/kl>|^{1/2} * N_{(I, J, K, L)}}{\sum_{KL} N_{(I, J, K, L)}}$$

Denominaremos este critério por B, ou critério de Schwartz, cabendo ressaltar que este critério se mostrou particularmente útil ao realizarem-se otimizações de geometria com o SCF-direto. Nesta situação, os vetores de partida são, em geral, de ótima qualidade pois são obtidos de soluções RHF de geometrias próximas. Deste modo, um grande número das integrais requeridas pelo critério de descarte, eq. 5 e 6, são guardadas em disco enquanto aquelas mais frequentemente descartadas passam a ser calculadas em memória.

Finalmente, o terceiro é um critério misto entre as seleções baseadas nos critérios de Tempo e Schwartz. Toma-se como média o produto destas estimativas, dado pela expressão 14.

$$14) \text{ Crit (IJ)} = \frac{\sum_{KL} |<ij/ij>|^{1/2} * |<kl/kl>|^{1/2} * t_{(I, J, K, L)} * N_{(I, J, K, L)}}{\sum_{KL} N_{(I, J, K, L)}}$$

Realizamos cálculos em nível SCF com otimização de geometria utilizando diferentes conjuntos de base (DZV, STO, DZP, 3-21G) para vários sistemas moleculares (benzeno, valina, tiofeno). A tabela 4 apresenta nossos resultados para cálculos SCF normais, enquanto a tabela 5 apresenta cálculos com otimização de geometria.

Os resultados apresentados nestas tabelas devem ser analisados com certo cuidado, particularmente em relação ao tempo decorrido, em função destes terem sido obtidos com o sistema em regime de tempo compartilhado. Apesar desta pequena imprecisão acreditamos que os tempos apresentados dão uma razoável idéia do custo associado a cada um dos critérios de seleção adotados nestas condições.

No caso de cálculos SCF comuns, o critério A, baseado no tempo de computação, mostra usualmente uma economia de 30% a 40% no tempo total decorrido sobre os demais, devido ao grande número de iterações SCF realizadas. Interessante é notar o comportamento destes critérios quando tem-se uma otimização de geometria. Nesta situação, embora na primeira geometria o cálculo SCF seja, em geral, mais rápido utilizando-se o critério A, ao longo das outras geometrias (7 pontos) os critérios B e C mostraram-se mais eficientes, tanto em relação ao tempo de CPU quanto no tempo decorrido. Este também é um

Tabela 4. Apresenta o tempo de CPU e o tempo decorrido do para o processo SCF. Tempo em segundos. O programa fez uso de 30% das integrais armazenadas em disco.

molécula e base	critério	t. de CPU	t. decorrido	
Benzeno	SCF-normal	33	76	
	DZV	SCF-direto	303	326
		Tempo	194	231
		Schwartz	256	290
	Misto	208	239	
Benzeno	SCF-normal	146	1017	
	DZP	SCF-direto	1230	1326
		Tempo	688	808
		Schwartz	1018	1148
	Misto	829	940	
Valina	SCF-normal	274	3257	
	STO	SCF-direto	1864	2082
		Tempo	1133	2172
		Schwartz	1384	2075
	Misto	1382	2070	
Valina	SCF-normal	-	-	
	DZV	SCF-direto	-	-
		Tempo	33398	49980
		Schwartz	37465	50340
	Misto	36359	46320	

Tabela 5. Tempo total de CPU e tempo decorrido para uma otimização de geometria (7 pontos) no contexto SCF-direto. Utilizou-se a base 3-21G com polarização no enxofre para a molécula de benzenotiofeno. Todos os cálculos utilizaram 30% das integrais em disco.

Molécula	base e critério	t. de CPU (secs.)	t. decorrido (secs.)
Benzotiofeno	3-21G		
	primeira geometria		
	SCF-normal	397	-
	Tempo	2436	-
	Schwartz	2608	-
	Misto	2523	-
	otimização completa		
	SCF-normal	3360	3h:00m
	Tempo	8366	3h:29m
	Schwartz	6723	2h:39m
Misto	6580	2h:37m	

comportamento esperado, pois com vetores iniciais de boa qualidade as integrais que mais contribuem são justamente aqueles de maior valor de Schwartz. Isto, de certa forma, sugere que o critério seja, em otimizações de geometrias, dinâmico, utilizando-se o critério de tempo para a primeira geometria e o de Schwartz para as demais.

### c) Paralelização e "benchmarks"

Trabalhos envolvendo paralelização de muitas aplicações em química teórica são rotineiros em vários laboratórios do mundo. Dentre estes lugares, destacava-se o laboratório do prof. Dr. Enrico Clementi na IBM, em Kingston<sup>16-18</sup>, onde uma rede de processadores IBM-3090 realizavam cálculos fundamentalmente dependentes da performance computacional.

No laboratório acima referido, eram disponíveis dois compiladores FORTRAN com diferentes filosofias para trabalhos em

paralelo. O primeiro, LCAP, possuía apenas comunicação mestre-escravo e podia utilizar todas as CPU's disponíveis em uma rede inter-sistemas (seis IBM-3090, envolvendo em torno de 32 CPU's). O segundo compilador, PFP, permitia cálculos com comunicação entre escravos e acesso a posições de memória comuns, mas limitados à utilização de um único sistema com até seis CPU's. Este segundo é o compilador comercial desenvolvido pela IBM para os sistemas do tipo 3090.

A enorme eficiência dos cálculos SCF-direto em relação à razão entre o tempo de CPU e o decorrido fazem deste tipo de método um excelente candidato para trabalhos em paralelo.

Adaptamos versões anteriores do programa HONDO ao compilador PFP, onde instalamos o programa SCF direto com as mesmas facilidades (seleção e uso parcial do disco) que o código serial. No processo de instalação, procurou-se adotar estratégias de paralelização que mantivessem a máxima compatibilidade com a versão serial do programa HONDO. Neste sentido, elaboramos um pequeno arquivo com aproximadamente 4000 linhas de código FORTRAN, incluindo algumas subrotinas modificadas, para fazer o gerenciamento da distribuição de tarefas entre os vários escravos para as etapas de integrais, SCF e derivadas.

Procurou-se fazer com que cada escravo calculasse um conjunto de integrais próprias que são guardadas em seu próprio disco. Durante os ciclos auto-consistentes, cada escravo recebe uma cópia da matriz densidade que é processada com as integrais encontradas por cada escravo em seus próprios discos, gerando uma matriz de Fock parcial para cada escravo. Estas são somadas ao final do processamento paralelo, a matriz resultante é diagonalizada e dá-se início a um novo ciclo. As únicas informações que os escravos (formalmente um programa HONDO rodando em cada um dos nós) necessitam são o número de escravos em uso e o número daquela tarefa em especial. O algoritmo 5 apresenta o fluxograma básico para o cálculo das integrais no SCF direto em paralelo.

Realizamos inúmeros testes em regime não-compartilhado, utilizando a versão serial e paralela do programa SCF direto e

*Algoritmo nº 5. Diagrama apresentando os principais comandos em linguagem FORTRAN para a rotina de cálculo das integrais e a estratégia de paralelização adotada. IAP é o identificador da tarefa enquanto NAP é o número de tarefas distribuídas.*

```
COMMON/LCAPID/IAP,NAP
começa o cálculo
IJKLSH=0
DO 9000 I=1,NSHELL
uso da simetria molecular
IF ('princípio de procrastinação') GO TO 9000
DO 8000 J=1,I
IF ('princípio de procrastinação') GO TO 8000
DO 7000 K=1,I
IF ('princípio de procrastinação') GO TO 7000
DO 6000 L=1, LMAX
IF ('princípio de procrastinação') GO TO 6000
IJKLSH = IJKLSH + 1
IF((IJKLSH-IJKLSH/NAP).NE.IAP) GO TO 6000
CON = max {ΔD (I, J, K, L)} * max {(ijkl)}
IF(DABS(CON).LT.CRIT) GO TO 6000
processa as integrais
6000 CONTINUE
7000 CONTINUE
8000 CONTINUE
9000 CONTINUE
```

convencional. A tabela 6 apresenta nossos resultados para o tempo decorrido de um SCF convencional, partindo da diagonalização do operador de um elétron, para a molécula de valina com base 3-21G. Nesta tabela, deve-se destacar que a primeira e segunda colunas não são representativas da eficiência de cálculos em paralelo, pois sob o sistema VM/CMS o compilador PFP realiza todo o processamento de I/O de maneira serial, por incompatibilidade dos sistemas na instalação. Entretanto, comparações podem ser feitas ao longo de uma linha demonstrando que, embora o tempo de CPU seja sensivelmente aumentado, o tempo decorrido (*elapsed*) é sempre menor. Comparando-se a versão serial do SCF-direto sem qualquer uso do disco, observa-se uma aceleração por um fator de 1,8 comparado ao uso convencional do SCF nas mesmas condições.

Analisando-se o fator entre o tempo decorrido nos cálculos SCF-direto com um e seis processadores, observa-se uma eficiência de 89% no processo de paralelização, o que significa um fator de aceleração de 5,34. A taxa de aceleração deste mesmo cálculo entre os dois extremos da tabela; cálculo SCF serial com 100% das integrais em disco e o cálculo SCF-direto sem uso do disco com seis escravos atinge o impressionante fator de aceleração de 9,62. A tabela 7 apresenta a eficiência de cada cálculo relativo a um SCF regular. O fator de aceleração que obtivemos, da ordem de 10, é bastante significativo, sendo de enorme importância em inúmeros cálculos SCF em sistemas moleculares de maior porte.

Os cálculos que realizamos com cadeias de benzotiofeno nos casos de 1, 2, 3 e 5 monômeros, são representativos da

*Tabela 6. Tempo total de CPU e tempo decorrido para cálculos SCF da valina com base 3-21G em regime não compartilhado. Resultados são apresentados para o caso do SCF regular com 1 e 2 escravos e para o SCF direto (sem armazenamento) a parcialmente direto (30%) usando de 1 a 6 escravos. Todos os tempos decorridos são dados em segundos com o tempo de CPU entre parênteses.*

num. de escravos	SCF normal	30% de armazenamento	0% de armazenamento
1	17073 (1340)	11900 (7310)	9509 (9370)
2	15340 (1470)	7871 (7400)	4843 (9400)
4	-	5905 (7480)	2536 (9520)
6	-	-	1774 (9630)

*Tabela 7. Eficiência e taxa de aceleração obtida para os cálculos apresentados na tabela 6.*

num. de escravos	SCF normal	30% de armazenamento	0% de armazenamento
1	100 (1)	100 (1,43)	100 (1,80)
2	56 (1,1)	76 (2,17)	98 (3,52)
4	-	50 (2,89)	94 (6,73)
6	-	-	89 (9,62)

*Tabela 8. Tempo decorrido e número de integrais para cálculos SCF de cadeias de benzotiofeno nos casos de 1, 2 e 3 monômeros. O caso com n=5 apresenta números estimados. O conjunto de base usa 3-21G para todos os átomos mais funções de polarização no enxofre. Todos os cálculos utilizam simetria C<sub>s</sub>, C<sub>2v</sub> e C<sub>2v</sub>. Os dois primeiros fazem uso de programas SCF normais.*

nº de monômeros	nº de base	nº de integrais (milhões)	tempo decorrido	uso do disco (%)	tamanho máximo do disco
1	103	7,3	3h	100	143
2	202	35,0	3d:4h	100	686
3	301	108,0	9d:2h	30	1350 (451)
5	499	386,0	27d:0h	10	10080 (732)

ajuda que se pode obter pelo uso de alternativas diretas para o cálculo SCF. Realizamos cálculos com otimização de geometria para esta cadeia utilizando um programa convencional. Neste último caso, as integrais ocuparam 686 cilindros e crescem a uma taxa proporcional à terceira potência do número de funções de base. O cálculo para  $n=3$  utilizou apenas 30% das integrais em disco, consumindo 451 cilindros. Considerando-se que o maior disco disponível naquele momento não ultrapassava os 1769 cilindros, conclui-se que este cálculo ainda podia ser realizado com os programas convencionais, embora já com alguma margem de chance de perda na tarefa. A partir de  $n=6$ , o uso convencional da técnica SCF já exige alterações de códigos de modo a se utilizar mais de um disco para armazenamento das integrais de dois elétrons.

Se analisarmos o fato de que utilizando o método SCF tradicional o aumento no tempo decorrido foi de aproximadamente 25 vezes entre o dímero e o trímero, e procurando comparar este número com o valor de 3 vezes obtido para a razão no tempo de computação decorrido utilizando-se o método SCF-direto entre o trímero e o  $n=4$ , chegaremos a conclusão que esta última metodologia deve ser preferencialmente utilizada em sistemas de grande demanda computacional em função da melhor taxa de crescimento quando comparada com o método SCF em sua formulação tradicional.

Outras metodologias importantes e relacionadas ao uso de técnicas de aceleração e convergência, tais como a inversão do espaço iterativo descrita por Pula<sup>12,13</sup>, uso de um limite de seleção variável, entre outras ferramentas, não foram atacados por falta de tempo. Estes pontos, entretanto, merecem consideração pois bom número de artigos discutem sua relevância no contexto de cálculos SCF-direto.

## CONCLUSÕES

Este trabalho apresenta modificações na metodologia SCF-direto tanto na forma de organização das integrais de dois elétrons como na forma de seleção para uso parcial e cálculos SCF-direto em paralelo. Resultados muito positivos são apresentados quanto à forma de seleção das integrais em disco e à paralelização desta metodologia.

Vários exemplos são apresentados, onde utiliza-se o método SCF-direto, em um ou mais processadores, em cálculos de diferentes sistemas químicos e diferentes conjuntos de base, podendo-se observar a grande superioridade deste tipo de cálculo quando aplicados a sistemas moleculares de grande porte. Esta tendência se torna particularmente interessante se aplicada a estações de trabalho em paralelo, onde a relativa limitação no espaço em disco e a crescente disponibilidade de CPUs rápidas e eficientes tornam esta proposta economicamente viável.

## AGRADECIMENTOS

Agradecemos de maneira toda especial ao Dr. M. Dupuis pela segura, exigente e eficiente supervisão no uso do programa HONDO, de sua autoria, e ao prof. Dr. Enrico Clementi pela hospitalidade e auxílio financeiro de seu laboratório no curto período em que este trabalho foi realizado.

## REFERÊNCIAS

1. Almlof, J.; Faegri Jr., K.; Korsell, K.; *J. Comp. Chem.*; (1980), **3**, 385.
2. Luethi, H. P.; Almlof, J.; *Chem. Phys. Lett.*; (1987), **135**, 357.
3. Pulay, P.; *Chem. Phys. Lett.*; (1980), **73**, 393.
4. Cremer, D.; Gauss, J.; *J. Comp. Chem.*; (1986), **7**, 274.
5. Ahlrichs, R.; Bar, M.; Haser, M.; Horn, H.; Komel, C.; "Electronic structure calculations on workstations computers"; The Program system TURBOMOLE, comunicações pessoais.
6. Haser, M.; Ahlrichs, R.; *J. Comp. Chemistry*; (1989), **10**, 104.
7. Hegarty, D.; Velde, G. van der; *Int. J. Quant. Chem.*; (1983), **23**, 1135.
8. Hegarty, D.; "Advanced theories and computational approaches to the electronic structure of molecules"; Dykstra, D.E.; Ed. D. Reidel Publishing Company, Dordrecht, Holland. 1984.
9. Dupuis, M.; King, H. F.; *Int. J. Quant. Chem.*; (1977), **11**, 613.
10. Dupuis, M.; Watts, J. D.; *Theor. Chim. Acta*; (1987), **71**, 91.
11. Ahlrichs, R.; *Theor. Chim. Acta*; (1974), **61**, 1300.
12. Pulay, P.; *J. Comp. Chem.*; (1986), **3**, 556.
13. Hamilton, T. P.; Pulay, P.; *J. Chem. Phys.*; (1986), **84**, 5728.
14. Clementi, E.; MOTECC; Ed. E. Clementi, ESCOM Science Pub., 1990.
15. Luthi, H. P.; Mertz, J.E.; Feyereissen, M. W.; Almlof, J. E.; "University of Minnesota SDupercomputer Institute Research", Report N91/141.
16. Corongiu, G.; Detrich, J. H.; IBM Technical Report, KGN 1"; Março 31, 1984.
17. Corongiu, G., Detrich, J. H.; *IBM J. Res. Dev.*, **29**, 422.
18. Clementi, E.; Chin, S.; Logan, D.; "Chaire Francqui", Lecture Series, Part 9, IBM, Kingston.
19. Clementi, E.; Corongiu, G.; Detrich, J. H.; Khanmohammadbaigi, H.; Chin, S.; Domingo, L.; Laaksonem, A.; Nguyen, H. L.; *IBM Technical Report*; KGN-2, Maio 1984.