



MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÃO
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

sid.inpe.br/mtc-m21b/2014/12.01.15.57-TDI

**PARALELIZAÇÃO DO TERMO DE REATIVIDADE
QUÍMICA DO MODELO AMBIENTAL CCATT-BRAMS
UTILIZANDO UM SOLVER BASEADO EM
ESTIMAÇÃO LINEAR ÓTIMA**

Alex de Almeida Fernandes

Dissertação de Mestrado do Curso de Pós-Graduação em Computação Aplicada, orientada pelos Drs. Stephan Stephany, e Jairo Panetta, aprovada em 28 de novembro de 2014.

URL do documento original:

<http://urlib.net/8JMKD3MGP3W34P/3HG84GL>

INPE
São José dos Campos
2014

PUBLICADO POR:

Instituto Nacional de Pesquisas Espaciais - INPE

Gabinete do Diretor (GB)

Serviço de Informação e Documentação (SID)

Caixa Postal 515 - CEP 12.245-970

São José dos Campos - SP - Brasil

Tel.:(012) 3208-6923/6921

Fax: (012) 3208-6919

E-mail: pubtc@sid.inpe.br

**COMISSÃO DO CONSELHO DE EDITORAÇÃO E PRESERVAÇÃO
DA PRODUÇÃO INTELECTUAL DO INPE (DE/DIR-544):****Presidente:**

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

Membros:

Dr. Gerald Jean Francis Banon - Coordenação Observação da Terra (OBT)

Dr. Amauri Silva Montes - Coordenação Engenharia e Tecnologia Espaciais (ETE)

Dr. André de Castro Milone - Coordenação Ciências Espaciais e Atmosféricas
(CEA)

Dr. Joaquim José Barroso de Castro - Centro de Tecnologias Espaciais (CTE)

Dr. Manoel Alonso Gan - Centro de Previsão de Tempo e Estudos Climáticos
(CPT)

Dr^a Maria do Carmo de Andrade Nono - Conselho de Pós-Graduação

Dr. Plínio Carlos Alvalá - Centro de Ciência do Sistema Terrestre (CST)

BIBLIOTECA DIGITAL:

Dr. Gerald Jean Francis Banon - Coordenação de Observação da Terra (OBT)

Clayton Martins Pereira - Serviço de Informação e Documentação (SID)

REVISÃO E NORMALIZAÇÃO DOCUMENTÁRIA:

Simone Angélica Del Duca Barbedo - Serviço de Informação e Documentação
(SID)

Yolanda Ribeiro da Silva Souza - Serviço de Informação e Documentação (SID)

EDITORAÇÃO ELETRÔNICA:

Marcelo de Castro Pazos - Serviço de Informação e Documentação (SID)

André Luis Dias Fernandes - Serviço de Informação e Documentação (SID)



MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÃO
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

sid.inpe.br/mtc-m21b/2014/12.01.15.57-TDI

**PARALELIZAÇÃO DO TERMO DE REATIVIDADE
QUÍMICA DO MODELO AMBIENTAL CCATT-BRAMS
UTILIZANDO UM SOLVER BASEADO EM
ESTIMAÇÃO LINEAR ÓTIMA**

Alex de Almeida Fernandes

Dissertação de Mestrado do Curso de Pós-Graduação em Computação Aplicada, orientada pelos Drs. Stephan Stephany, e Jairo Panetta, aprovada em 28 de novembro de 2014.

URL do documento original:

<<http://urlib.net/8JMKD3MGP3W34P/3HG84GL>>

INPE
São José dos Campos
2014

Dados Internacionais de Catalogação na Publicação (CIP)

Fernandes, Alex de Almeida.

F391p Paralelização do termo de reatividade química do modelo ambiental ccatt-brams utilizando um solver baseado em estimação linear ótima / Alex de Almeida Fernandes. – São José dos Campos : INPE, 2014.

xxii + 80 p. ; (sid.inpe.br/mtc-m21b/2014/12.01.15.57-TDI)

Dissertação (Mestrado em Computação Aplicada) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2014.

Orientadores : Drs. Stephan Stephany, e Jairo Panetta.

1. CCATT-BRAMS. 2. OpenMP. 3. Química atmosférica.
4. Estimação linear ótima. I.Título.

CDU 004.272:551.510.41

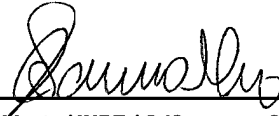


Esta obra foi licenciada sob uma Licença [Creative Commons Atribuição-NãoComercial 3.0 Não Adaptada](https://creativecommons.org/licenses/by-nc/3.0/).

This work is licensed under a [Creative Commons Attribution-NonCommercial 3.0 Unported License](https://creativecommons.org/licenses/by-nc/3.0/).

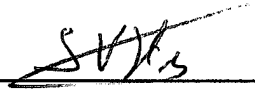
Aprovado (a) pela Banca Examinadora
em cumprimento ao requisito exigido para
obtenção do Título de **Mestre** em
Computação Aplicada

Dr. Solon Venâncio de Carvalho



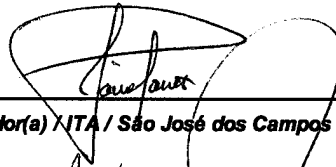
Presidente / INPE / SJC Campos - SP

Dr. Stephan Stephany



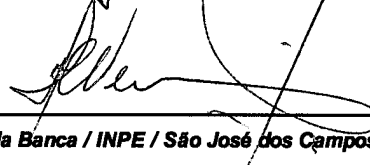
Orientador(a) / INPE / SJC Campos - SP

Dr. Jairo Panetta



Orientador(a) / ITA / São José dos Campos - SP

Dr. Haroldo Fraga de Campos Velho



Membro da Banca / INPE / São José dos Campos - SP

Dr. Carlos Alberto Alonso Sanches



Convidado(a) / ITA / São José dos Campos - SP

Este trabalho foi aprovado por:

maioria simples

unanimidade

Aluno (a): **Alex de Almeida Fernandes**

São José dos Campos, 28 de Novembro de 2014

“A vontade é impotente perante o que está para trás dela. Não poder destruir o tempo, nem a avidez transbordante do tempo, é a angústia mais solitária da vontade”.

Friedrich Nietzsche

A meu Irmão Bravo, a minha família, aos amigos de longe e de perto, daqui e de lá...

AGRADECIMENTOS

Agradeço aos meus professores, de agora e de antes, todos que me ensinaram e que continuam me empurrando nesta caminhada que é a vida. Que eles saibam o valor inestimável que possuem, não apenas no tocante a minha educação, mas para a humanidade, difundindo o conhecimento em um gesto de abertura e desprendimento, sem egoísmo. Entrego-lhes meu sincero agradecimento.

Grato, também, sou ao meu irmão mais bravo, que mesmo não podendo escutá-lo todos os dias, sei que se faz presente em minha caminhada. E quando digo minha caminhada, que esta seja a nossa! E que, para o alto da eternidade seja a nossa direção. Agradeço-lhe por me retirar da inércia. Para ti, meu profundo obrigado.

A minha esposa, Liza, pela compreensão. Somente a convivência e o amor para transpor tamanha dificuldade. Obrigado pela companhia, obrigado pela vida conjunta, obrigado pelos desafios. E que continuemos assim, com muitos desafios, para que a vida jamais seja monótona.

E por último agradeço as minhas pequenas, pelos choros, sorrisos puros e pela obrigatoriedade das madrugadas de estudo. Mesmo que elas ainda não compreendam o que isso tem significado para o meu “relógio biológico” e minhas poucas horas de sono. Obrigado.

RESUMO

O Centro de Previsão de Tempo e Estudos Climáticos (CPTEC/INPE) desenvolveu e utiliza o modelo ambiental CCATT-BRAMS que acopla um modelo de transporte de gases traço ao modelo regional BRAMS. Este modelo ambiental possui um termo de reatividade química que atualiza a composição química da atmosfera para o passo de tempo considerado. Este termo modela as reações químicas entre componentes da atmosfera e deve ser resolvido para cada ponto da grade espacial. A composição química da atmosfera influi na absorção de radiação solar pela atmosfera impactando seu balanço térmico. Entretanto, seu custo computacional ao ser executado a cada passo de tempo é uma ordem de grandeza superior ao do modelo BRAMS, de forma que seu uso operacional é possível com sua execução a cada 4 passos de tempo. Esse custo deve-se à resolução de um sistema linear de equações para cada ponto de grade. O *solver* utilizado correntemente tem uma implementação sequencial e sua paralelização demandaria alterações significativas do código correspondente. Assim, este trabalho propõe o uso de um *solver* baseado em estimação linear ótima que possui um código mais simples e mais apropriada à paralelização. Testes demonstraram que o *solver* proposto fornece resultado numericamente equivalentes àqueles do *solver* original e que possibilita um ganho de desempenho ao ser paralelizado com a biblioteca OpenMP. Investigou-se também a possibilidade de portá-lo para execução em placa aceleradora gráfica (GPU) prevendo a disponibilidade desses aceleradores num futuro supercomputador do CPTEC/INPE.

PARALLELIZATION OF THE CHEMICAL REACTIONS MODULE OF THE CCATT-BRAMS ENVIRONMENTAL MODEL USING A SOLVER BASED ON OPTIMAL LINEAR ESTIMATION

ABSTRACT

The Center for Weather Forecasting and Climate Studies (CPTEC/INPE) developed and employs the environmental model CCATT-BRAMS that couples a pollutant transport model and the regional numerical model BRAMS. This environmental model includes a chemical reactions module that updates the chemical composition of the atmosphere for the considered timestep. This module models the chemical reactions between atmospheric components and must be solved for each gridpoint. The atmospheric chemical composition influences the absorption of solar radiation by the atmosphere thus affecting its thermal balance. However the computational cost of executing this module at every timestep is one order of magnitude higher than that of the BRAMS model. Therefore, its operational use is feasible with an execution every 4 timesteps. Such computational cost is due to the resolution of a linear system of equations for each gridpoint. The currently employed solver has a serial version and would demand large modifications of the code in order to be parallelized. This work proposes the use of a solver based on optimal linear estimation that has a code that is simpler and more suitable for parallelization. Numerical tests show that the proposed solver achieved results that are numerically equivalent to those of the original solver. In addition, the OpenMP parallel version of the proposed solver achieved reasonable speed-up. This work also investigated the portability of the proposed solver for execution in a graphics processing unit (GPU) since such accelerators may be available in a future supercomputer of CPTEC/INPE.

LISTA DE FIGURAS

	<u>Pág.</u>
Figura 3.1: Interações entre a atmosfera, a superfície terrestre e a atividade antropogênica.	11
Figura 3.2: Área abrangida pelo modelo CCAT-BRAMS para uma grade de 200 x 200 x 38 pontos e dividida em quadrados de 1 grau x 1 grau (à esquerda), sendo um desses quadrados detalhado com a resolução do modelo de 0,5 grau (à direita).	14
Figura 3.3: Matriz esparsa A , com o posicionamento dos elementos não nulos representado em preto e o dos elementos nulos, em branco.	23
Figura 4.1: Evolução do desempenho em GFlop/s de processadores Intel, coprocessadores Intel Phi e GPGPUs da Nvidia e AMD de 2007 a 2013. .	37
Figura 4.2: Evolução do desempenho em (GFlop/s)/watt de processadores Intel, coprocessadores Intel Phi e GPGPUs da Nvidia e AMD de 2007 a 2013 ..	38
Figura 6.1: Figura esquemática da divisão do domínio em MPI em blocos de pontos de grade e o mapeamento de pontos de grade de um bloco para execução por <i>threads</i> em OpenMP da resolução do termo de reatividade química.	49
Figura 6.2: Tempo de execução da versão 1-OMP do novo <i>solver</i> em função do número de <i>threads</i>	53
Figura 6.3: <i>Speed-up</i> da versão 1-OMP do novo <i>solver</i> em função do número de <i>threads</i>	53
Figura 6.4: Eficiência da versão 1-OMP do novo <i>solver</i> em função do número de <i>threads</i>	54
Figura 6.5: Taxa de operações em ponto flutuante da versão 1-OMP do novo <i>solver</i> em função do número de <i>threads</i>	54
Figura 6.6: Aumento relativo do tempo de processamento do modelo CCATT-BRAMS em função do tamanho de bloco para execuções com chamadas ao termo de reatividade química (curva vermelha), e para execuções sem essas chamadas (curva azul).	56
Figura 6.7: Tempos de processamento do termo de reatividade química na máquina Raven expressos em função do tamanho de bloco, para execuções de um passo de tempo.	57
Figura 6.8: Tempo de execução da versão 1-OMP do novo <i>solver</i> em função do número de <i>threads</i> para diferentes tamanhos de bloco.....	58
Figura 6.9: <i>Speed-up</i> da versão 1-OMP do novo <i>solver</i> em função do número de <i>threads</i> para diferentes tamanhos de bloco.....	59

Figura 6.10: Eficiência da versão 1-OMP do novo <i>solver</i> em função do número de <i>threads</i> para diferentes tamanhos de bloco.....	59
Figura 6.11: Tempo de execução da versão 2-OMP do novo <i>solver</i> em função do número de <i>threads</i> para diferentes tamanhos de bloco.....	62
Figura 6.12: <i>Speed-up</i> da versão 2-OMP do novo <i>solver</i> em função do número de <i>threads</i> para diferentes tamanhos de bloco.....	63
Figura 6.13: Eficiência da versão 2-OMP do novo <i>solver</i> em função do número de <i>threads</i> para diferentes tamanhos de bloco.....	63
Figura 6.14: Tempos de processamento dos blocos para a grade completa do CCATT-BRAMS.....	64
Figura 6.15: Tempo de execução da versão 2-OMP do novo <i>solver</i> em função do número de <i>threads</i> para o laço dos blocos percorrido na ordem original (curva azul) e na ordem inversa (curva vermelha).....	66
Figura 6.16: <i>Speed-up</i> da versão 2-OMP do novo <i>solver</i> em função do número de <i>threads</i> para o laço dos blocos percorrido na ordem original (curva azul) e na ordem inversa (curva vermelha).....	66
Figura 6.17: Eficiência da versão 2-OMP do novo <i>solver</i> em função do número de <i>threads</i> para o laço dos blocos percorrido na ordem original (curva azul) e na ordem inversa (curva vermelha).....	67
Figura 6.18: Tempos de execução do termo de reatividade química com o novo <i>solver</i> para diferentes tamanhos de bloco para a versão sequencial, a versão OpenACC e para a versão 2-OMP executada com 2, 4, 8, 16 e 32 <i>threads</i>	70
Figura 6.19: Tempos de execução do termo de reatividade química com o novo <i>solver</i> para diferentes tamanhos de bloco para a versão sequencial, a versão OpenACC (linha azul cheia) e para a versão 2-OMP executada com 2, 4, 8, 16 e 32 <i>threads</i> (colunas azuis) e com o <i>solver</i> original (área sombreada azul). A linha vermelha tracejada mostra o melhor tempo do <i>solver</i> original, relativo ao tamanho de bloco 48.....	71

LISTA DE TABELAS

	<u>Pág.</u>
Tabela 3.1: Profile do tempo de execução para um único passo de tempo do modelo acoplado CCATT-BRAMS com e sem a execução do termo de reatividade química do modelo (tempos em segundos).....	12
Tabela 4.1: Número de sistemas lineares para execução de 1 hora de previsão, considerando-se diferentes resoluções.	29
Tabela 5.1: <i>Profile</i> do tempo de execução do termo de reatividade química com o <i>solver</i> original para 3 execuções (T1, T2 e T3) correspondendo a um único passo de tempo do modelo CCATT-BRAMS e um único bloco de 64 pontos de grade (tempos em segundos). As linhas com fundo azul correspondem às chamadas ao <i>solver</i> original.	44
Tabela 6.1: Tempos de execução dos <i>solvers</i> considerando um passo de tempo do modelo e um bloco de 64 pontos de grade. (tempos em segundos).	46
Tabela 6.2: Número de blocos em função do número de ajustes de passo necessários nos estágios do método de Rosenbrock e tempos de execução totais para ambos os <i>solvers</i> considerando um passo de tempo do modelo e todos os 23.750 blocos de pontos de grade (tempos em segundos).	46
Tabela 6.3: <i>Profile</i> do tempo de execução do termo de reatividade química com o novo <i>solver</i> para 3 execuções (T1, T2 e T3) correspondendo a um único passo de tempo do modelo CCATT-BRAMS e um único bloco de 64 pontos de grade (tempos em segundos). As linhas com fundo azul correspondem às chamadas ao novo <i>solver</i>	48
Tabela 6.4: Concentrações das 47 espécies químicas obtidas pelo novo <i>solver</i> e pelo <i>solver</i> original. As colunas referem-se às concentrações máximas encontradas, às coordenadas (i,j,k) da grade onde estas ocorreram, e às concentrações médias em toda a grade, sendo cada linha correspondente a uma espécie química (concentrações em parte por bilhão ou em moléculas/cm ³ segundo a espécie).	72

LISTA DE SIGLAS E ABREVIATURAS

API	<i>Application Program Interface</i>
BRAMS	<i>Brazilian Developments on the Regional Atmospheric Modeling System</i>
CATT	<i>Coupled Aerosol and Tracer Transport</i>
CCATT	<i>Coupled-Chemical, Aerosol and Tracer Transport</i>
CPTEC	Centro de Previsão do Tempo e Estudos Climáticos
CPU	<i>Central Processing Unit</i>
CUDA	<i>Compute Unified Device Architecture</i>
DSP	<i>Digital Signal Processor</i>
ELO	Estimação Linear Ótima
FPGA	<i>Field-Programmable Gate Array</i>
GFlop	<i>Giga Floating Point Operations</i>
GPGPU	<i>General Purpose Graphics Processing Units</i>
GPU	<i>Graphics Processing Unit</i>
HPC	<i>High-performance computing</i>
INPE	Instituto Nacional de Pesquisas Espaciais
MFlop	<i>Mega Floating Point Operations</i>
MIC	<i>Intel Many Integrated Core Architecture</i>
MPI	<i>Message Passing Interface</i>
OpenMP	<i>Open Multi-Processing</i>
OpenACC	<i>Open Accelerators</i>
PGF90	<i>The Portland Group Fortran 90</i>
RACM	<i>Regional Atmospheric Chemistry Mechanism</i>
RAMS	<i>Regional Atmospheric Modeling System</i>
RELACS	<i>Regional Lumped Atmospheric Chemical Scheme</i>
RISC	<i>Reduced Instruction Set Computer</i>

SUMÁRIO

	<u>Pág.</u>
1	INTRODUÇÃO 1
2	POLUIÇÃO ATMOSFÉRICA 5
3	TERMO DE REATIVIDADE QUÍMICA 9
3.1.	Modelo ambiental CCAT-BRAMS 9
3.2.	Modelagem do termo de reatividade química 15
3.3.	Solução numérica do termo de reatividade química 18
3.3.1.	<i>Solver</i> original 23
3.3.2.	Novo <i>solver</i> 25
4	PARADIGMAS DE PARALELIZAÇÃO UTILIZADOS 29
4.1.	Padrão OPENMP 30
4.2.	Padrão OPENACC 32
4.2.1.	Placas aceleradoras gráficas 35
5	IMPLEMENTAÇÃO COMPUTACIONAL DO SOLVER ORIGINAL 39
5.1.	<i>Profile</i> do tempo de execução do termo de reatividade química com o <i>solver</i> original 43
6	IMPLEMENTAÇÃO COMPUTACIONAL DO NOVO SOLVER 45
6.1.	Paralelização usando o padrão OpenMP 48
6.1.1.	Primeira versão OpenMP 50
6.1.2.	Influência do tamanho de bloco 55
6.1.3.	Segunda versão OpenMP 60
6.1.4.	Balanceamento de carga 64
6.2.	Paralelização com OpenACC 67
6.3.	Diferenças numéricas nos resultados 71
7	CONCLUSÕES 75
	REFERÊNCIAS BIBLIOGRÁFICAS 77

1 INTRODUÇÃO

O ar atmosférico seco na troposfera é composto tipicamente em volume por nitrogênio (78%), oxigênio (21%) e os restantes 1% incluem os chamados gases traço, assim chamados devido às suas pequenas concentrações. Nesta categoria aparecem alguns gases nobres, o gás carbônico e gases poluentes gerados por causas antropogênicas ou naturais. Numa atmosfera real é preciso incluir também vapor d'água, além dos aerossóis, que analogamente aos gases poluentes, são provenientes da queima de combustíveis fósseis, processos industriais, queimadas, erupções vulcânicas, processos biogênicos, etc. Assim, gases poluentes e aerossóis são constantemente lançados na atmosfera, seja em grandes centros ou em regiões desabitadas. Deste modo, torna-se desejável que modelos numéricos possam incluir gases traço e aerossóis de forma a considerar suas influências.

Os gases traço mais comuns são o monóxido de carbono, óxidos de enxofre, óxidos de nitrogênio e compostos orgânicos voláteis, além do ozônio, este último no caso da baixa troposfera. No caso dos aerossóis, que são partículas em suspensão na atmosfera, podem ser sólidos, líquidos ou de origem biológica. Como exemplo de aerossóis sólidos, pode-se citar a poeira ou a fuligem, enquanto que gotículas de vapor d'água condensado fora de nuvens são aerossóis líquidos e aerossóis de origem biológica são vírus e bactérias em suspensão na atmosfera. Concentrações altas desses gases traço e aerossóis impactam a atmosfera e a biosfera terrestre ocasionando problemas de saúde, degradação de biomas, desequilíbrio ecológico, migrações de animais, perda de plantações, secas, tempestades, etc.

É importante notar que esses gases traço e aerossóis podem também afetar o balanço térmico da atmosfera. Esse efeito nem sempre é incluído em modelos numéricos de previsão de tempo e de clima. Mesmo quando as concentrações de gases traço e aerossóis são levadas em conta, muitas vezes essas

concentrações são consideradas invariáveis no tempo, por simplicidade ou devido ao custo computacional. Uma modelagem adequada requer o acoplamento de um modelo de transporte de gases traço e aerossóis ao modelo numérico de previsão de tempo.

O modelo de transporte resolve a equação de continuidade de massa dos gases traço e aerossóis, que inclui termos tais como advecção (movimentação do fluído na horizontal), convecção (movimentação do fluído na vertical), turbulência, deposição e emissão. Advecção e convecção normalmente são termos utilizados na meteorologia, sendo que a engenharia não difere convecção de advecção. Na realidade, os gases traço reagem quimicamente entre si modificando a composição do ar atmosférico ao longo do tempo. Estas reações químicas podem ser modeladas pela inclusão de um termo de reatividade química nessa equação da continuidade de massa. Obviamente, a modelagem resultante permite prever melhor o comportamento da atmosfera e das concentrações de gases traço e aerossóis para cada ponto de grade possibilitando mitigar efeitos nocivos para a população e para a economia.

O Centro de Previsão de Tempo e Estudos Climáticos do Instituto Nacional de Pesquisas Espaciais (CPTEC/INPE) utiliza modelos de previsão numérica de tempo e de clima desde 1994 (PANETA et. al., 2006). Modelos acoplados são mais recentes, como no caso do modelo CATT-BRAMS, constituído pelo modelo

de transporte de gases traço e aerossóis, o *Coupled Aerosol Tracer Transport* (CATT) (FREITAS et. al., 2006) e pelo modelo numérico de previsão de tempo BRAMS (*Brazilian Developments on the Regional Atmospheric Modeling System*) (FREITAS et. al., 2007). Estes modelos são executados concorrentemente de forma acoplada, sendo suas funções de cálculos atmosféricos e de transporte intercaladas em um mesmo passo de tempo (*timestep*), ou seja, o CATT atualiza as concentrações dos gases traço e

aerossóis em cada ponto de grade para um dado *timestep* e então executa-se o BRAMS para o mesmo *timestep*.

Mais recentemente, a inclusão do termo de reatividade química ao CATT-BRAMS resultou no modelo ambiental CCATT (*Coupled-Chemical, Aerosol and Tracer Transport*) (LONGO et. al. 2013), correntemente utilizado pelo CPTEC/INPE, que assim se tornou pioneiro na execução operacional de um modelo numérico ambiental para a predição das concentrações de espécies químicas na atmosfera.

O CCATT-BRAMS permite prever a dinâmica química dos gases traço presentes na atmosfera, disponibilizando previsões sobre a qualidade do ar e estimando o balanço energético da atmosfera com maior precisão. Entretanto, devido ao seu custo computacional, o termo de reatividade química somente é executado a cada 4 passos de tempo (*timesteps*), no caso do modelo acoplado utilizar resoluções espaciais altas. Considera-se que para que uma previsão numérica de tempo seja viável, o tempo de execução do modelo não deve ultrapassar 2% do tempo referente à previsão (GOVETT, 2009), o que não ocorreria se o termo de reatividade química fosse executado a cada *timestep*.

Tipicamente, o cálculo desse termo representa de 50% a 95% do tempo de processamento total do modelo acoplado (ZHANG et. al., 2011; LINFORD et. al., 2009). Em particular, no caso do CCATT-BRAMS, representa 80% do tempo total, quando executado no supercomputador Tupã (Cray XE6) do CPTEC/INPE.

Neste contexto, o presente trabalho busca uma alternativa viável em termos de tempo de processamento para a execução do termo de reatividade química a cada *timestep*. Isso implica na resolução, para cada ponto da grade espacial considerada, de um sistema de equações lineares que expressa as reações químicas envolvendo gases traço da atmosfera. A resolução desse sistema para todos os pontos da grade, implementada computacionalmente por um *solver* muito eficiente, porém sequencial, é muito custosa

computacionalmente. Alternativamente, investiga-se aqui o uso de um *solver* alternativo, baseado em estimação linear ótima (ELO), seja na versão sequencial, seja na versão paralelizada com a biblioteca OpenMP (openmp.org). Compara-se o desempenho do *solver* original, executado sequencialmente, com o desempenho do *solver* proposto, analisando-se também o desempenho paralelo deste último. Finalmente, investiga-se a possibilidade de portar o *solver* proposto para uma versão executável numa placa aceleradora gráfica de uso geral, em inglês, *General Purpose Graphics Processing Units* (GPGPUs), na qual cada núcleo/core da GPGPU executaria uma *thread* que resolveria o sistema de equações lineares para um determinado ponto de grade.

As contribuições dessa dissertação são relativas à (i) implementação computacional do *solver* baseado em estimação linear ótima para resolução do termo de reatividade química em Fortran 90, sendo esse programa integrado ao modelo ambiental CCATT-BRAMS, à (ii) paralelização do código do termo de reatividade química utilizando-se o *solver* baseado em ELO com a biblioteca OpenMP baseada na execução paralela de *threads* em máquinas de memória compartilhada, e à (iii) investigação de sua portabilidade para biblioteca OpenACC (openacc.org), visando sua execução em placas aceleradoras gráficas. Note-se que o *solver* originalmente utilizado no modelo CCATT-BRAMS foi implementado na linguagem C, ao passo que o restante do modelo, em Fortran 90, o que dificultou a análise de seu desempenho computacional e impediu sua paralelização.

Como resultado deste trabalho, verificou-se que o novo *solver* proposto forneceu resultados numericamente equivalentes àqueles do *solver* original, além de reduzir o tempo de processamento quando paralelizado com OpenMP.

2 POLUIÇÃO ATMOSFÉRICA

A constituição química da atmosfera terrestre não é a mesma de 3,5 bilhões de anos atrás, quando a atmosfera era provavelmente desprovida de oxigênio, até que os primeiros seres vivos surgiram (ALTERMANN, KAZMIERCZAK; 2003) e, eventualmente, a fotossíntese iniciou a injeção de oxigênio no ar. Assim, a constituição química da atmosfera sofreu alterações contínuas e, desde a Revolução industrial, iniciada em fins do século XVIII, essas alterações são em grande parte antropogênicas, ou seja, causadas pela influência humana. As mais notórias são as emissões na atmosfera de gases do efeito estufa, de diversos tipos de aerossóis, gases primários tóxicos e/ou precursores de gases secundários nocivos à saúde, como por exemplo, o ozônio (O_3). Além dessas emissões, houve a mudança do uso e cobertura do solo, que alteram o balanço entre a radiação incidente e a refletida (albedo) na atmosfera (FREITAS et. al. 2009).

Tipicamente, modelos de previsão numérica de tempo e clima utilizam uma composição química simplificada da atmosfera, assumindo que as concentrações dos diversos componentes permaneçam invariáveis no tempo. Essa seria uma hipótese razoável, por exemplo, sobre os oceanos, onde não ocorre injeção de gases traço e aerossóis, mas é preciso considerar o transporte advectivo, devido ao vento, de aerossóis sólidos, líquidos ou gasosos de origem antropogênica ou natural que sofrem dispersão, se espalham, ao serem injetados na atmosfera. Deste modo, partículas de areia do deserto do Saara são transportadas para a costa oriental da América do Sul (MORRIS et. al. 2006.). Além disso, a presença de aerossóis sólidos altera o balanço térmico da atmosfera, assim como ocorre com os aerossóis líquidos.

No Brasil, queimadas antropogênicas são comuns nos meses de seca, quando a vegetação nativa, campos de culturas agrícolas ou então áreas de pastagens são queimados para dar lugar a novas culturas ou pastagens. Afora as queimadas antropogênicas no Brasil e em diversos outros países, sabe-se

que, uma das maiores fontes de energia em países de terceiro mundo provém da queima de biomassa (lenha, bagaço de cana, etc.), e que cerca de 80% desta queima ocorre nos trópicos (ANDREAE; CRUTZEN, 1997, ARBEX et al. 2004). Embora a queima de biomassa ocorra principalmente na região dos trópicos, note-se que, segundo o *Emissions Database for Global Atmospheric Research* (OLIVIER et al., 2014), os maiores poluidores encontram-se no hemisfério norte.

A combustão de biomassa é responsável direta pela maior parte de emissão de componentes químicos na atmosfera, entre os quais o gás carbônico, metano e perfluorcarbonetos (ARBEX et. al. 2004), que contribuem muito para o efeito estufa. Também geram óxido e dióxido de nitrogênio, além de metano, que modificam a composição química da atmosférica por causarem reações químicas que causam elevado nível de ozônio troposférico e a chamada chuva ácida (WARD et. al. 1992). O monóxido de carbono e os N-óxidos de nitrogênio são gases precursores do ozônio troposférico, juntamente com outros produtos da combustão, como o vapor d'água (BREMER et. al. 2004, GALANTER et. al. 2000).

Os aerossóis sólidos, como o material particulado proveniente da queima de biomassa, afetam também a microfísica das nuvens, fazendo com que as gotas de chuva não alcancem massa suficiente para que precipitem, ficando suspensas na atmosfera. Além disso, a presença de uma maior quantidade de gotículas não precipitáveis aumenta a refletância da atmosfera, diminuindo a radiação solar incidente na superfície do planeta e conseqüentemente tornando a convecção junto à superfície menos intensa, o que inibe a formação de nuvens convectivas. Isso implica em menos chuvas, alterando de maneira importante o ciclo hidrológico numa escala regional e alterando o padrão de redistribuição planetária de energia dos trópicos para as latitudes médias e altas (FREITAS et. al., 2009), o que implica em menos chuva. Esses aerossóis sólidos também possuem influência durante as estações mais secas ao

contrabalancear o aquecimento causado pelos gases do efeito estufa (COX et. al. 2008).

A combustão da biomassa e a conseqüente produção de material particulado que apresenta toxicidade afeta obviamente a saúde da população. Esse material é constituído em sua maior parte (94% em massa) por partículas finas e ultrafinas (PM25, partículas com 2.5 nanômetros) que atingem as vias respiratórias e transpõem a barreira epitelial, atingindo o interstício pulmonar e sendo responsáveis pelo desencadeamento de processos inflamatórios (ARBEX et. al. 2004). Compostos oxidantes provindos da combustão de biomassa ou mesmo da poluição de grandes centros, também afetam a saúde humana, pois apresentam alta reatividade química causando a liberação de radicais livres, os quais oxidam os aminoácidos do tecido proteico (BRICKUS et. al. 1999).

Outro produto indireto da combustão de biomassa já mencionado, o ozônio troposférico, é altamente tóxico e também afeta a saúde humana quando respirado, além de degradar plantas (FREITAS et. al. 2009). O ozônio é também considerado o composto com maior potencial oxidante presente na atmosfera, sendo que sua presença pode deteriorar materiais numa gama que vai desde a borracha até materiais de construção (BRICKUS et. al. 1999). Assim, dadas as alterações na composição química da atmosfera provocadas pela emissão de gases traço e aerossóis, torna-se desejável modelar melhor as reações químicas associadas para possibilitar uma melhor previsão da qualidade do ar. A disponibilidade de uma composição química mais acurada da atmosfera possibilita que os modelos meteorológicos sejam também mais precisos ao simular a atmosfera, seja em escala regional ou sinótica. Em meteorologia, esta última escala se aplica a áreas com dimensões acima de 1000 km, sendo também é chamada de grande escala.

3 TERMO DE REATIVIDADE QUÍMICA

Este capítulo trata do termo de reatividade química, que adiciona ao modelo ambiental CCATT-BRAMS a funcionalidade de simular as reações químicas entre gases traço da atmosfera. O cálculo do termo de reatividade química permite atualizar as concentrações dos gases traço ao longo dos passos de tempo do modelo. Este cálculo implica na resolução de vários sistemas de equações lineares, conforme será descrito adiante. O modelo desenvolvido e implementado pelo CPTEC/INPE baseia-se na resolução desse sistema pelo chamado *solver* original, enquanto que este trabalho propõe o uso de um novo *solver*, baseado em estimação linear ótima, que possibilita paralelizações.

Nas seções seguintes, descreve-se o modelo ambiental CCATT-BRAMS, a modelagem do termo de reatividade química, sua solução numérica que redundava no sistema de equações lineares e, finalmente, o *solver* original e o proposto. Destaca-se que o CCATT-BRAMS e o termo de reatividade química foram codificados em Fortran 90 (LONGO et. al. 2013), exceto pelo *solver* original que resolve esse sistema de equações lineares, que foi codificado na linguagem C.

3.1. Modelo ambiental CCAT-BRAMS

A modelagem numérica de tempo e clima busca prever um estado futuro da atmosfera baseado no estado atual, modelando a evolução temporal da atmosfera. Lewis Fry Richardson, na década de 1920, utilizou alguns procedimentos descritos por Vilhelm Bjerknes para calcular manualmente a primeira previsão de tempo de 6 horas considerando apenas dois pontos sobre a Europa Central, mas isso demandou várias semanas de cálculos e resultou em grandes erros em relação aos valores observados. Mas tarde foi constatado que esses erros decorreram da falta de técnicas de suavização de dados (LYNCH, 2008).

Entretanto, lançou a concepção da previsão de tempo como sendo um problema de condição inicial.

Na década de 1950, John Von Neumann teve a percepção de que a modelagem numérica de tempo se tornaria viável com o advento da computação, i.e. que os cálculos demandados poderiam ser realizados num tempo muito menor que o tempo de previsão. Assim, realizou a primeira previsão numérica de tempo no computador ENIAC. Em 1954, utilizando o mesmo modelos, Carl-Gustav Rossby realizou a primeira previsão operacional de tempo. Em 1955, iniciou-se a previsão operacional de tempo no EUA. Ao longo dos anos, a modelagem numérica do tempo foi aperfeiçoada e novos supercomputadores permitiram o uso de maiores resoluções espaciais e temporais, melhorando a precisão das previsões (LYNCH, 2008).

No Brasil, o CPTEC/INPE fornece operacionalmente previsões numéricas de tempo e clima desde 1995, empregando diversos modelos, escalas e resoluções.

Atualmente, além de diversos modelos em escala global e regional, o CPTEC/INPE executa operacionalmente o modelo ambiental CCATT-BRAMS, que acopla o modelo numérico de previsão de tempo BRAMS (FREITAS et al., 2007) a um modelo CATT de transporte de gases traço e aerossóis (FREITAS et al., 2006) e resolve ao longo dos passos de tempo o termo de reatividade química. Esses 3 modelos (BRAMS, CATT e a química) são executados concorrentemente, com o CATT atualizando as concentrações dos gases traço e aerossóis em cada ponto de grade para um dado passo de tempo e o BRAMS simulando o estado da atmosfera para o mesmo passo de tempo. E seria desejável que o termo de reatividade química fosse também executado a cada passo de tempo do modelo acoplado.

À semelhança de outros modelos meteorológicos e ambientais, o CCATT-BRAMS adota uma grade tridimensional que cobre regiões do globo terrestre, sendo composto por uma grade horizontal de latitudes e longitudes e uma

grade vertical, especificada em níveis de pressão atmosférica. A Figura 3.1 ilustra alguns dos fenômenos que podem ser simulados pelo modelo, que considera a interação da atmosfera com a superfície terrestre e a atividade antropogênica.

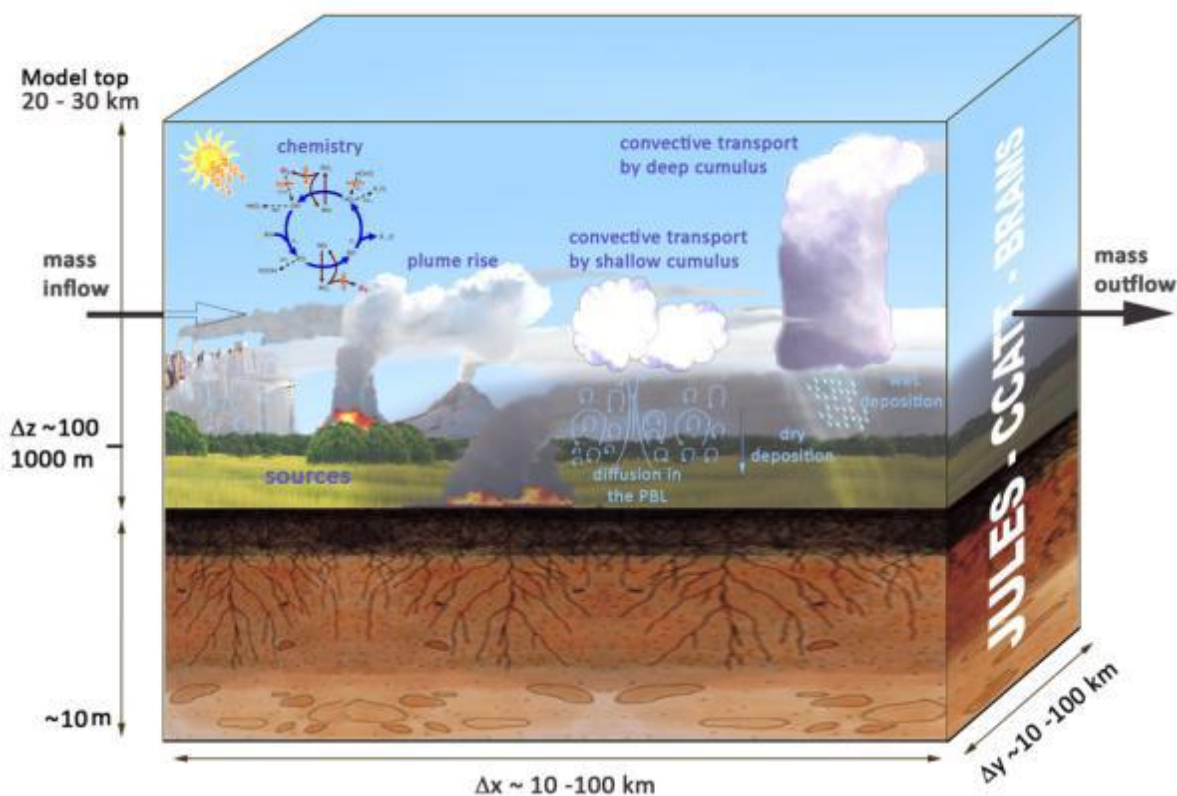


Figura 3.1: Interações entre a atmosfera, a superfície terrestre e a atividade antropogênica.

Fonte: MOREIRA et. al.; 2013.

O modelo BRAMS deriva do *Regional Atmospheric Modeling System* (RAMS), desenvolvido pela *Colorado State University*, modelo de previsão numérica do tempo. O RAMS é um modelo com opção não hidrostática (i.e. considera o ar compressível), de modo que todas as escalas meteorologicamente relevantes possam ser consideradas e que possui parametrizações avançadas para representar a interação entre a superfície terrestre e os diversos níveis de pressão (PIELKE, et al., 1992). Essas parametrizações permitem modelar a

turbulência, a convecção, a radiação solar incidente e refletida (albedo) e a microfísica das nuvens. O modelo permite o uso de grades aninhadas com diferentes resoluções, variando de vários quilômetros a alguns metros. O BRAMS incorpora parametrizações específicas para as regiões tropical e subtropical (FREITAS et. al. 2009).

Na versão operacional do CCATT-BRAMS, o tempo de processamento excessivo demandado pelo termo de reatividade química acarretou sua execução a cada 4 passos de tempo. Isso pode ser observado na Tabela 3.1, no *profile* de tempo de execução do CCATT-BRAMS com e sem a resolução do termo de reatividade química (referido como "química") para um passo de tempo do modelo utilizando-se o *solver* original. Nota-se que o tempo de processamento aumenta mais de 100 vezes. Este profile refere-se à execução sequencial nos nós de processamento auxiliar da máquina Tupã para o código gerado pelo compilador PGF90 (Fortran 90) da PGI e para uma grade espacial de 200 x 200 x 38 pontos (38 níveis verticais).

Tabela 3.1: Profile do tempo de execução para um único passo de tempo do modelo acoplado CCATT-BRAMS com e sem a execução do termo de reatividade química do modelo (tempos em segundos).

	Tempo (s) com Química	Tempo (s) sem Química
Sources Driver	0.092	0.092
Radiate	0.004	0.004
Surface Layer	1.079	1.065
co2_biosource	0	0
drydep_driver	0	0
Corlos	0.018	0.0018
Advection	0.074	0.073
CuPArm	4.875	4.89
Chemistry 1	0.019	0.019
Chemistry 2	0	0
Chemistry 3	0.686	0.686
Chemistry 4	0.683	0.682
Chemistry 5	834.424	0
Total	841.953	7.529

Na tabela 3.1, as funções *Chemistry 1* a *5* representam funções executadas pelo módulo de química. A atualização da produção líquida e da concentração das espécies químicas na atmosfera são executados pelo *Chemistry 5*, em *timesteps* múltiplos, por exemplo, a cada 4 *timesteps* do modelo CCATT-BRAMS.

Entretanto, essa tabela refere-se a execuções do modelo CCATT-BRAMS sem o módulo de radiação, executado a cada 30 passos de tempo. Mesmo nesse caso, o tempo de execução do modelo CCATT-BRAMS ainda é dominado pelo termo de química. O passo de tempo do modelo é de 40 s, sendo o termo de reatividade química resolvido a cada 4 passos de tempo do modelo, e o módulo de radiação a cada 30. Assim, a execução do modelo para 1 hora (3600 s) de tempo de simulação é então de $90 \times 7.5 \text{ s} = 11.25 \text{ min}$, somado a $22 \times (841.9 \text{ s} - 7.5 \text{ s}) = 306 \text{ min}$ e mais a radiação $3 \times 850 \text{ s} = 42.5 \text{ min}$. Obviamente executar o modelo CCATT-BRAMS com essa resolução seria inviável, mesmo sem o termo de reatividade química, pois cada hora simulada demandaria 53,75 min e só esse termo adicionaria 5 horas de processamento! Assim, essa elevada resolução espacial foi utilizada para demonstrar esse ponto. Essa grade de $200 \times 200 \times 38$ pontos corresponde a uma resolução espacial de 0.05 grau (cerca de 5 km), sendo a área total abrangida apresentada na Figura 3.2.

CCATT-BRAMS

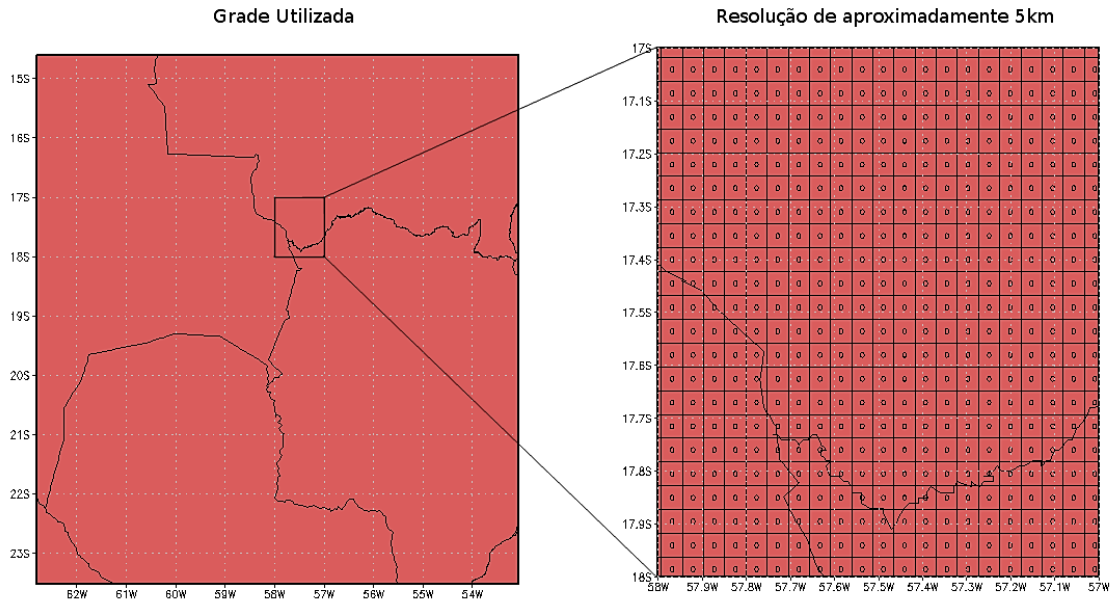


Figura 3.2: Área abrangida pelo modelo CCATT-BRAMS para uma grade de 200 x 200 x 38 pontos e dividida em quadrados de 1 grau x 1 grau (à esquerda), sendo um desses quadrados detalhado com a resolução do modelo de 0,5 grau (à direita).

O sistema CCATT-BRAMS resolve, a cada passo de tempo do modelo, a equação geral da continuidade de massa dos gases traço (LONGO et. al., 2013):

$$\frac{\partial \bar{S}}{\partial t} = \left(\frac{\partial \bar{S}}{\partial t} \right)_{adv} + \left(\frac{\partial \bar{S}}{\partial t} \right)_{PBL\ diff} + \left(\frac{\partial \bar{S}}{\partial t} \right)_{deep\ conv} + \left(\frac{\partial \bar{S}}{\partial t} \right)_{shallow\ conv} + \left(\frac{\partial \bar{S}}{\partial t} \right)_{chem} + W + R + Q \quad (3.1)$$

onde \bar{S} é a razão de mistura média dos gases traço, o termo "adv" representa o transporte devido à advecção pelo vento médio, o termo "PBL diff", a turbulência na escala de subgrade na camada limite planetária (PBL, do inglês *Planetary Boundary Layer*), o termo "deep conv", a convecção profunda, o termo "shallow conv", a convecção rasa, e o termo "chem" as reações

químicas entre gases traço. Aparecem, também, o termo W , que considera a remoção úmida (somente para aerossóis), o termo R relativo à deposição seca (para gases traço e aerossóis sólidos) e o termo fonte Q que contabiliza as emissões de gases traço na atmosfera e que inclui o mecanismo de elevação da pluma gerada por queimadas.

No modelo anterior, CATT-BRAMS, o termo "chem" apenas levava em conta a vida média de gases traço passivos na atmosfera, sendo parametrizado, e conseqüentemente, não sendo atualizado com os passos de tempo do modelo. Por outro lado, o CCATT-BRAMS, que acrescenta no início o "C" de *chemistry*, termo "chem" modela as reações químicas entre os gases traço instáveis, sendo acoplado ao modelo, i.e. atualizado ao longo dos passos de tempo. Entretanto, essas reações químicas não incluem reações entre gases traço e aerossóis. A solução numérica deste termo no CCATT-BRAMS é exposta na seção seguinte.

3.2. Modelagem do termo de reatividade química

Para a modelagem do termo "chem" da equação geral da continuidade de massa (Equação 3.1) exposta na seção anterior, assume-se que os gases traço atmosféricos reagem apenas no ponto de grade considerado (LONGO et. al. 2013), i.e. no volume associado a esse ponto de grade, cujas dimensões dependem da resolução espacial considerada. Descartam-se assim influências de gases traço de volumes vizinhos ou suas reações químicas. Isso possibilita reescrever a derivada parcial que aparecia na Equação 3.1 como uma derivada ordinária. Assim como no modelo CATT-BRAMS, este termo baseia-se numa descrição Euleriana dos escoamento na atmosfera, i.e. fazendo-se os cálculos de qualquer grandeza para o volume associado a cada ponto de grade. Considerando-se N espécies químicas reagentes, que são os gases traço instáveis, o termo de reatividade química para o gás traço k é então dado por:

$$\left(\frac{\partial \rho_k}{\partial t}\right)_{chem} = \left(\frac{d\rho_k}{dt}\right) = P_k(\boldsymbol{\rho}) - L_k(\boldsymbol{\rho}) \quad (3.2)$$

Nesta equação, $\boldsymbol{\rho} = \{\rho_1, \rho_2, \dots, \rho_N\}$ é o vetor de densidade das espécies químicas reagentes consideradas, e P_k e L_k são os termos de produção e perda da espécie química k (LONGO et. al., 2013). Na solução desta equação utiliza-se, a cada passo de tempo de integração τ , o vetor de densidades no instante inicial para se calcular o vetor de densidades no instante final. Os termos de produção e perda modelam também processos fotoquímicos relativos a cada espécie. Define-se como mecanismo químico um esquema qualquer que modele as reações químicas entre gases traço, incluindo reações fotoquímicas e eventualmente reações com aerossóis. Embora o CCATT-BRAMS permita incorporar qualquer mecanismo químico, limitações de tempo de processamento levaram à escolha de um mecanismo químico com menos espécies, no caso 37, que é o *Regional Atmospheric Chemistry Mechanism* (RACM, STOCKWELL et al., 1997) que modela reações entre 77 espécies. Existe, por exemplo, outro mecanismo, o RELACS (*Regional Lumped Atmospheric Chemical Scheme* (RELACS, CRASSIER et al., 2000). Na verdade, o RELACS é uma simplificação do RACM, mas provê resultados muito parecidos ao modelar as reações químicas. O RELACS considera 37 espécies químicas e 128 reações possíveis entre elas, incluindo as reações fotolíticas. A maioria das reações químicas entre as espécies devem-se a oxidantes como o radical OH.

Os termos de produção ($P_k(\boldsymbol{\rho})$) e de perda ($L_k(\boldsymbol{\rho})$) da Equação 3.2 são funções que modelam as reações químicas e fotoquímicas para as espécies consideradas por meio de taxas de variação das concentrações, que são calculadas em função das condições atmosféricas (pressão, temperatura) e das próprias concentrações por um módulo de cinética química, que é

executado internamente com o termo de reatividade química. Há também um módulo de fotoquímica, mas que é executado apenas uma vez, por simplificação. Assim, as funções de produção e perda são atualizadas a cada passo de tempo do modelo. No caso do módulo de fotoquímica, exemplifica-se abaixo a fotólise que ocasiona a perda de uma espécie A e sua transformação numa espécie B (produção de B) por efeito da luz visível ou ultravioleta (FREITAS, 2009). Esta reação fotoquímica é expressa pelo sistema de equações diferenciais abaixo, onde ρ_A e ρ_B são suas concentrações e q_A e q_B representam as taxas de produção ou perda.

$$\begin{cases} \frac{d\rho_A}{dt} = -q_A\rho_A \\ \frac{d\rho_B}{dt} = +q_B\rho_B \end{cases} \quad (3.3)$$

No caso do módulo de cinética química, exemplificam-se abaixo as reações que ocasionam a perda das espécies A e B e a produção das espécies C e D , ou seja a reação química entre A e B gerando C e D . Esta reação é expressa pelas respectivas equações diferenciais, onde ρ_A , ρ_B , ρ_C e ρ_D são suas concentrações e r_{ABCD} representa a taxa de produção/perda que será utilizada para atualizar as funções de produção/perda dos elementos A , B , C e D .

$$\begin{cases} \frac{d\rho_A}{dt} = \frac{d\rho_B}{dt} = -r_{ABCD}\rho_A\rho_B \\ \frac{d\rho_C}{dt} = \frac{d\rho_D}{dt} = +r_{ABCD}\rho_C\rho_D \end{cases} \quad (3.4)$$

3.3. Solução numérica do termo de reatividade química

O termo de reatividade química do modelo CCATT-BRAMS foi implementado em Fortran 90 (LONGO et al. 2013) e resolve o sistema de equações diferenciais da Equação 3.2. Conforme descrito na seção anterior, as funções de produção e de perda são atualizadas em determinados passos de tempo do modelo. Entretanto, esse sistema de equações diferenciais é rígido (ou stiff), devido à disparidade entre as ordens de grandeza das taxas de variação de concentrações devidas às reações químicas e fotoquímicas (ZHANG et al, 2011). Um sistema rígido poderia ser resolvido por um método implícito ou explícito (CURTISS, HIRSCHFELDER, 1952). Entretanto, o uso de um método explícito demandaria um passo de integração pequeno o suficiente para descrever todas as variações da função, acarretando um grande custo computacional e inviabilizando a solução num tempo de processamento conveniente. Assim, para o CCATT-BRAMS foi adotado o método de Rosenbrock (HARIER; WANNER, 1991; VERWER et al., 1999), que é um método implícito do tipo preditor-corretor pertencente à família dos métodos Runge-Kutta implícitos, em que o passo de integração é ajustado em função do erro calculado.

Esse método permite transformar um sistema de equações não lineares em soluções baseadas em sistemas lineares algébricos (LAMBERT, 1992), resolvendo o sistema (*stiff*) não linear de equações diferenciais da Equação 3.2 por meio de uma integração numérica no tempo em que as concentrações das espécies químicas são atualizadas, a partir das concentrações iniciais em t_0 , após um passo de tempo τ por meio de incrementos algébricos.

A solução de sistemas do tipo *stiff* demanda métodos específicos, onde o método mais simples normalmente utilizado é o método implícito de Euler, porém o seu uso acarreta resultados muito imprecisos. Neste escopo, Rosenbrock (1963) propôs uma alternativa com múltiplos estágios, baseada

nos métodos de Runge-Kutta, para obtenção de resultados com maior precisão.

A formulação para o método de Rosenbrock é comumente encontrada na literatura como se segue (HARRIER; WANNER, 1991; VERWER et al., 1999):

$$y_{n+1} = y_n + \sum_{i=1}^s b_i K_i \quad (3.5)$$

$$K_i = \left[f \left(y_n + \sum_{j=1}^{i-1} \alpha_{ij} K_j \right) + J \sum_{j=1}^i \gamma_{ij} K_j \right] \Delta t \quad (3.6)$$

Onde y é a função a ser integrada numericamente, discretizada num passo de tempo Δt ; α_i , b_i e γ_i são constantes definidas apropriadamente, s é o número de estágios, e K_i é o incremento da função no estágio i . f é uma aproximação da função y e J é a matriz jacobiana de ordem N calculada apenas em t_0 (tempo inicial) sobre a função $F(\rho_0)$, que é o termo de produção líquida inicial.

Aplicando-se o método de Rosenbrock (Equações 3.5 e 3.6) à equação diferencial ordinária (Equação 3.2) ao, obtém-se as equações descritas em LONGO et al. 2013, nas quais a função y corresponde ao vetor de concentrações químicas ρ e o termo de produção líquida no estágio i é denotado por $F(\rho_i)$:

$$\rho_i = \rho(t_0) + \sum_{i=1}^s b_i K_i \quad (3.7)$$

$$K_i = \left[F(\rho_i) + J \sum_{j=1}^i \gamma_{ij} K_j \right] \Delta t \quad (3.8)$$

Devida a natureza implícita do método, \mathbf{K}_i aparece em ambos os lados da equação 3.8, sendo necessário reescrevê-la de forma a isolar este vetor, conforme abaixo:

$$(\mathbf{I} - \gamma\Delta t\mathbf{J}) \cdot \mathbf{K}_i = \Delta t \left[\mathbf{F}(\boldsymbol{\rho}_i) + \mathbf{J} \cdot \sum_{j=1}^{i-1} \gamma_{ij} \mathbf{K}_j \right] \quad (3.9)$$

Porém, a formulação acima tem elevado custo computacional, considerando-se que há multiplicação de matrizes e vetores em ambos os lados da equação. Por conveniência, seguindo procedimento de PRESS et al. (1992), define-se por conveniência um vetor \mathbf{g} , de forma a agrupar alguns termos da Equação 3.9:

$$\mathbf{g}_i = \sum_{j=1}^i \gamma_{ij} \mathbf{K}_j = \sum_{j=1}^{i-1} \gamma_{ij} \mathbf{K}_j + \gamma \mathbf{K}_i \quad (3.10)$$

Aplicando-se a Equação 3.10 na Equação 3.8 para 4 estágios do método de Rosenbrock, chega-se às equações efetivamente codificadas no termo de reatividade química:

$$\begin{aligned} (\mathbf{I}/\gamma\Delta t - \mathbf{J}) \cdot \mathbf{g}_1 &= \mathbf{F}(\boldsymbol{\rho}_0) \\ (\mathbf{I}/\gamma\Delta t - \mathbf{J}) \cdot \mathbf{g}_2 &= \mathbf{F}(\boldsymbol{\rho}_0 + a_{21}\mathbf{g}_1)/\gamma + c_{21}\mathbf{g}_1/(\Delta t\gamma) \\ (\mathbf{I}/\gamma\Delta t - \mathbf{J}) \cdot \mathbf{g}_3 &= \mathbf{F}(\boldsymbol{\rho}_0 + a_{31}\mathbf{g}_1 + a_{32}\mathbf{g}_2) + (c_{21}\mathbf{g}_1 + c_{32}\mathbf{g}_2)/\Delta t \\ (\mathbf{I}/\gamma\Delta t - \mathbf{J}) \cdot \mathbf{g}_4 &= \mathbf{F}(\boldsymbol{\rho}_0 + a_{41}\mathbf{g}_1 + a_{42}\mathbf{g}_2 + a_{43}\mathbf{g}_3) + (c_{21}\mathbf{g}_1 + c_{32}\mathbf{g}_2 \\ &\quad + c_{43}\mathbf{g}_3)/\Delta t \end{aligned} \quad (3.11)$$

Assim, sendo obtidos os valores de g_i , atualizam-se posteriormente os valores de K_i pela equação 3.10.

Nas equações acima, a matriz jacobiana J é calculada apenas no estágio inicial, sendo reutilizada nos demais estágios, a , c , γ são constantes já determinadas conforme o algoritmo exposto por PRESS et al..

A cada estágio, as concentrações intermediárias são derivadas dos valores de g_i de forma a atualizar a função F de produção líquida $F(\rho_i)$ que representa a diferença entre o termo de produção $P(\rho_i)$ e o termo de perda $L(\rho_i)$, em função do vetor das concentrações das espécies químicas ρ . Estes termos de produção e de perda constituem o lado direiro da equação 3.2. Essa função de produção líquida atualizada é então utilizada no estágio seguinte.

$$F(\rho_i) = P(\rho_i) - L(\rho_i) \quad (3.12)$$

O conjunto de sistemas de equações lineares expresso pelas Equações 3.11 possui ordem N correspondente ao número de espécies químicas consideradas, e tem a forma básica:

$$A \cdot x = b \quad (3.13)$$

Na Equação 3.13, o vetor de incógnitas x correspondem aos incrementos g_i para os 4 estágios do método de Rosenbrock, e a matriz A corresponde ao Jacobiano J , que expressa a taxa de reações das espécies químicas consideradas no instante t_0 . em função de suas concentrações. Finalmente, o termo independente b , corresponde ao termo de produção líquida: $F(\rho_i)$, sendo que, a cada estágio, este termo depende das aproximações intermediárias K_i . Fica claro que o procedimento é iterativo, pois, estima-se um valor de K_i para se estimar concentrações ρ_i , as quais são usadas para estimar o termo de

produção líquida $F(\rho_i)$ e estas, para estimar novamente K_i . No quarto e último estágio, o método de Rosenbrock prevê que o K_4 seja próximo de zero. Caso esse valor ultrapasse um limiar determinado, o passo τ é diminuído para o próximo passo de tempo de integração, no qual serão executados os 4 estágios seguintes do método. Obviamente, o passo de tempo do modelo é muito maior que o passo de tempo de integração do método de Rosenbrock.

O sistema da Equação 3.12 tem particularidades que facilitam sua solução. É um sistema muito esparso, uma vez que a matriz A , que corresponde ao Jacobiano da produção líquida, tem aproximadamente apenas 10% de elementos não nulos. Deve-se observar que, devido à modelagem do problema, todos os termos da diagonal dessa matriz são não-nulos. Outro aspecto é que, para cada ponto de grade, essa matriz tem seus elementos alterados ao longo do tempo, mas o posicionamento dos elementos não-nulos dentro da matriz não varia, pois as reações sempre ocorrem entre os mesmos componentes. Na figura 3.3 é apresentada a estrutura da matriz do sistema linear, onde o posicionamento dos elementos não nulos é representado em preto.

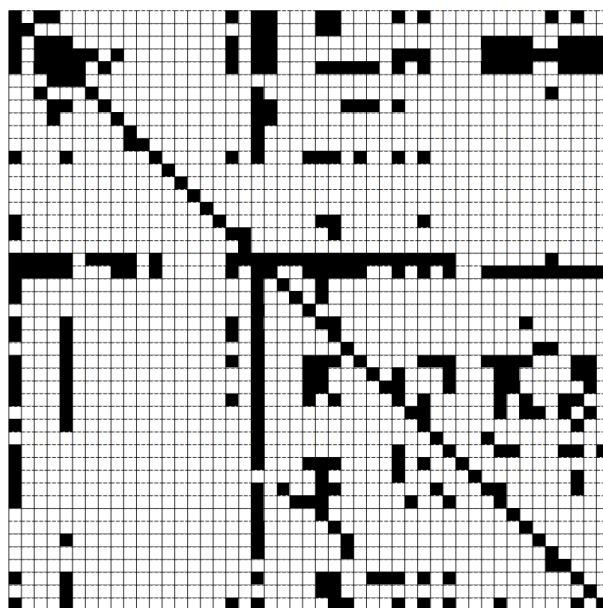


Figura 3.3: Matriz esparsa **A**, com o posicionamento dos elementos não nulos representado em preto e o dos elementos nulos, em branco.

A resolução do sistema da Equação 3.12 é feita no CCATT-BRAMS pelo *solver* original, que utiliza a decomposição LU, codificada em linguagem C, conforme descrito na Seção 3.3.1. Este trabalho propõe a resolução deste sistema por um novo *solver*, baseado em estimação linear ótima e codificado em Fortran 90, apresentado na Seção 3.3.2.

3.3.1. *Solver* original

O *solver* atualmente utilizado no termo de reatividade química do CCATT-BRAMS foi implementado na linguagem C e resolve o sistema de equações lineares da Equação 3.12 pela decomposição da matriz *A* de coeficientes em 3 matrizes: diagonal (*D*), triangular superior (*U*) e triangular inferior (*L*) (KUNDERT, 1986; KUNDERT; SANGIOVANNI-VINCENTELLI, 1988):

$$A = D + L + U \tag{3.14}$$

O método de Gauss-Seidel é então aplicado nestas matrizes, de forma a aproveitar a esparsividade do sistema.

O *solver* original implementa a decomposição LU e sua correspondente solução conforme proposta por Kundert (1986), que resolve o sistema estimando-se a solução na iteração $(m + 1)$ a partir da solução na iteração m conforme abaixo ($m = 1, 2, \dots$):

$$\mathbf{x}^{m+1} = \mathbf{D}^{-1}[-(\mathbf{L} + \mathbf{U})\mathbf{x}^m + \mathbf{b}] \quad (3.15)$$

Este método foi implementado usando-se rotinas na linguagem C da biblioteca Sparse 1.3. Esta biblioteca otimiza o armazenamento da matriz em função de sua esparsividade, armazenando apenas os elementos não nulos e realizando operações apenas entre estes elementos. As rotinas utilizam uma lista encadeada de elementos não nulos. Como a matriz A é invariante, seus elementos não nulos ocupam sempre as mesmas posições e a lista pode ser reutilizada ao longo dos sucessivos passos de tempo, apenas com a eventual atualização destes valores não nulos. Deve-se observar que, devido à modelagem do problema, todos os termos da diagonal dessa matriz são não-nulos. Este reuso da estrutura das matrizes para ordenação e fatoração, torna o *solver* original extremamente rápido e estável na solução do sistema.

Além da rotina específica para decomposição LU, a biblioteca Sparse 1.3 inclui também outras rotinas, que são utilizadas na implementação do *solver* original, para controle de erros, fatoração, pré-condicionamento de matrizes, etc. Além disso, utiliza uma estrutura de dados própria, de uso comum para suas diversas rotinas. Entretanto, as otimizações típicas da linguagem C incluídas na programação dessas rotinas, tais como alocação dinâmica da memória e o uso de ponteiros para listas encadeadas, tornam-nas de difícil paralelização.

Essas rotinas na linguagem C correspondem a quase 13.000 linhas de código e foram compiladas e incorporadas ao termo de reatividade química na forma de objetos. Entretanto, todo o restante deste termo foi codificado em Fortran 90, somando mais de 27.000 linhas de código. Isso impede o uso de paradigmas como o OpenMP, baseado em paralelismo de *threads* pois, em geral, compiladores OpenMP não permitem chamadas a funções sem fazer *inlining* das mesmas, o que naturalmente impede o uso de funções em uma linguagem diferente.

3.3.2. Novo *solver*

O escopo deste trabalho é a implementação de um novo *solver* que possibilite a execução do termo de reatividade química, a cada passo de tempo do modelo CCATT-BRAMS, para resolução espacial e temporal com que esse modelo é correntemente executado no ambiente operacional do CPTEC/INPE. Isso eventualmente viabilizaria a execução do próprio CCATT-BRAMS com resoluções espaciais e temporais melhores (i.e. menores). Em modelos tais como o CCATT-BRAMS, o aumento da resolução espacial requer um aumento conjunto da resolução temporal, de forma a garantir a estabilidade numérica do modelo.

O modelo CATT-BRAMS é executado em paralelo com uso da biblioteca de comunicação por troca de mensagens MPI, sendo o termo de reatividade química executado sequencialmente em cada nó do supercomputador tupã para o conjunto de pontos de grade dado pela divisão do domínio feita na paralelização. Uma vez que a implementação sequencial do *solver* original é muito eficiente, um novo *solver* somente atingiria esse objetivo por meio de sua paralelização, sendo adotada a linguagem Fortran 90. Optou-se por não alterar a divisão de domínio feito pelo MPI, adotando-se assim para o novo *solver* o padrão OpenMP de paralelização por *threads*. Isso motivou a busca por um novo *solver* que fosse conveniente para esta paralelização, sendo a

simplicidade e o tamanho do código correspondente um dos fatores considerados. Além disso, a precisão numérica da solução, a convergência rápida para esta e a estabilidade numérica também foram consideradas.

O novo *solver* escolhido para implementação e testes de desempenho computacional foi proposto por Pinto e Rios Neto (1990). Trata-se de um *solver* baseado em estimação linear ótima, que utiliza uma versão fatorizada do algoritmo sequencial de filtro de Kalman para resolver o sistema de equações lineares. A fatorização empregada é a fatoração de raiz quadrada de Potter, que torna o algoritmo mais estável numericamente e menos suscetível a erros de arredondamento. Esse *solver* pode ser aplicado a sistemas rígidos, independentemente de serem ou não esparsos, pois prescinde do pré-condicionamento da matriz A do sistema $A \cdot x = b$ de ordem N (Equação 3.13).

O algoritmo sequencial de filtro de Kalman, é descrito pelo pseudocódigo a seguir, onde h_i é o vetor dos coeficientes da i -ésima equação do sistema da Equações 3.13, h_i^T é o vetor transposto de h_i , x_i é o vetor solução, b_i é o termo independente, e P_i é o vetor de projeção associado à matriz A .

INICIALIZA $x_i = 0.0$ (vetor solução com estimativa inicial)

INICIALIZA $P_0 = I_N$ (matriz identidade de ordem N)

PARA $i=1$ até N **FAÇA** (laço das espécies químicas)

$$\begin{aligned}\beta_i &= (r_i + h_i^T P_{i-1} h_i)^{-1} \\ p_i &= \beta_i P_{i-1} h_i \\ P_i &= P_{i-1} - p_i h_i^T P_{i-1} \\ x_i &= x_{i-1} + (b_i - h_i^T x_{i-1}) p_i\end{aligned}$$

FINALIZE (PARA)

No algoritmo descrito no pseudocódigo acima, considerando-se nulos os elementos r_i , obtêm-se que $\beta_i = (h_i^T P_{i-1} h_i)^{-1}$. Aplicando-se a fatoração de raiz quadrada de Potter nesse algoritmo chega-se ao algoritmo implementado, denominado neste trabalho por novo *solver*, que utiliza a matriz quadrada S :

INICIALIZA $x_i = 0.0$ (vetor com estimativa inicial)

INICIALIZA $S_0 = I_N$ (matriz identidade de ordem N)

PARA $i=1$ até N **FAÇA** (laço das espécies químicas)

$$\begin{aligned}v_i &= h_i^T S_{i-1} \\ \beta_i &= (v_i^T v_i)^{-1} \\ p_i &= \beta_i S_{i-1} v_i \\ S_i &= S_{i-1} - p_i v_i\end{aligned}$$

FINALIZE (PARA)

ENQUANTO precisão desejada não for alcançada **FAÇA**

PARA $i=1$ até N **FAÇA** (laço das espécies químicas)

$$x_i = x_{i-1} + (b_i - h_i^T x_{i-1}) p_i$$

FINALIZE (PARA)

FINALIZE (ENQUANTO)

Nota-se que o vetor de projeções p_i depende apenas das equações h_i , que são invariantes. Conseqüentemente, nos estágios do método de Rosenbrock, estes vetores p_i não precisam ser recalculados (PINTO, RIOS NETO, 1990). Assim, cada estimação do vetor solução x_i no laço ENQUANTO-FAÇA resume-se a executar uma multiplicação matriz-vetor ($h_i \times x_i$) de ordem de complexidade $O(N^2)$. O solver originalmente utilizado (Gauss-Siedel) também possui complexidade de $O(N^2)$, enquanto a eliminação de Gauss possui complexidade $O(N^3)$. Por outro lado, o método do gradiente conjugado possui complexidade $O(N)$, porém requer o pré-condicionamento da matriz A para obter estabilidade numérica.

Entretanto, a matriz A é esparsa, com a maioria dos elementos fora da diagonal nulos e, portanto, muitos termos dessa multiplicação não precisam ser calculados. Note-se que essa esparsividade não pode ser explorada nos passos intermediários, pois a matriz S , bem como os vetores v e p passam a ter cada vez mais elementos não nulos.

Além disso, as estimativas iniciais para os vetores solução x_i são nulas apenas no primeiro estágio, mas em cada estágio subsequente aproveita-se a estimativa obtida no estágio anterior. Esse esquema melhora a precisão final da solução e provê estabilidade numérica ao método. Adicionalmente, diminui o número de iterações em cada estágio do método de Rosenbrock. Constatou-se que, na maioria das vezes, basta uma única iteração para se chegar a uma solução, sem que haja necessidade de pré-condicionamento da matriz.

O novo *solver* proposto, graças às vantagens expostas acima e à simplicidade de seu código, apresentou um desempenho sequencial e uma precisão numérica compatíveis com o *solver* original, além de possibilitar sua paralelização.

4 PARADIGMAS DE PARALELIZAÇÃO UTILIZADOS

Para exemplificar o potencial de paralelismo existente no termo de reatividade química, é exposta a tabela 4.1, que mostra a quantidade de pontos de grade/sistemas lineares que são resolvidos para uma hora de previsão para diferentes resoluções espaciais. Nesta tabela observa-se que uma grade com melhor resolução espacial (mais refinada) possui mais pontos de grade e requer um menor passo de tempo (*timestep*) para garantir a estabilidade numérica, acarretando mais execuções para gerar uma hora de previsão. Assim, o aumento da resolução provoca um aumento exponencial da quantidade de sistemas lineares a serem resolvidos (mais pontos de grade, menores passos de tempo). Existe portanto um paralelismo potencial no termo de reatividade química, uma vez que esses sistemas lineares podem ser resolvidos independentemente.

Tabela 4.1: Número de sistemas lineares para execução de 1 hora de previsão, considerando-se diferentes resoluções.

Resolução	NX	NY	NZ	Nº Pontos	<i>Timestep</i> (s)	Exec. (1 Hr)	Nº Sistemas
20 Km	340	370	42	5.283.600	30	30	158.508.000
10 Km	680	740	42	21.134.400	15	60	1.268.064.000
05 Km	1360	1480	42	84.537.600	9	100	8.453.760.000
05 Km	200	200	38	1.520.000	40	22	33.440.000

Explorar esse paralelismo no termo de reatividade química foi o caminho adotado para se diminuir o tempo de execução do modelo CCATT-BRAMS. Entretanto, como esse modelo já é paralelizado pela biblioteca de comunicação por troca de mensagens (MPI), não seria conveniente ou viável utilizar essa biblioteca para paralelização do *solver* do termo de reatividade química, uma vez que a paralelização com MPI pressupõe uma divisão de domínio espacial, cabendo um conjunto de pontos de grade a cada nó de memória compartilhada do supercomputador tupã do CPTEC/INPE.

Assim, a escolha natural seria a paralelização desse *solver* segundo o padrão OpenMP. Entretanto, antevendo-se um possível futuro supercomputador com GPGPUs, explorou-se também o padrão OpenACC. Há trabalhos em que modelos de química foram portados com sucesso para execução em GPGPU, embora com a API CUDA C (ELLER, 2009). As seções a seguir descrevem sucintamente estes dois padrões, bem como as GPGPUs.

4.1. Padrão OPENMP

Nos últimos anos, processadores multinúcleos tornaram-se onipresentes, seja em máquinas individuais, seja em supercomputadores. Assim, comumente tem-se máquinas de memória compartilhada com vários processadores multinúcleo. No caso do supercomputador tupã do CPTEC, os 1304 nós de processamento computacional tem 2 processadores de 12 núcleos cada um. A execução do MPI em máquinas de memória compartilhada é comum, resultando na execução concorrente de vários processos. Entretanto, surgiu um novo padrão específico para essas arquiteturas, baseado na execução concorrente de vários *threads* de um único processo, o padrão OpenMP (Open Multi-Processing). Em tese, nesse tipo de arquitetura, a execução paralela de *threads* seria mais eficiente do que a execução concorrente de processos porque não haveria necessidade de troca de mensagens entre *threads* (como ocorre entre processos no MPI), uma vez que todos os *threads* pertencem a um único processo e tem acesso à sua área de memória. Cada troca de mensagens implica no custo de se executar uma rotina da biblioteca MPI, o que não ocorre no caso do OpenMP. Obviamente, já existia a programação por *threads* utilizando bibliotecas específicas, mas o OpenMP constitui uma API que permite um maior nível de abstração, tornando mais simples a programação paralela, mesmo em relação ao MPI.

O OpenMP é uma API (*Application Programming Interface*) proposta e desenvolvida por um consórcio que reúne desenvolvedores de software e

fabricantes de computadores/processadores como AMD, Intel, NVidia, IBM, Cray, HP e outros. Foi lançado na segunda metade da década de 90, e possui versões C/C++ e Fortran 77/90 para diversos sistemas operacionais. Inclui um conjunto de diretivas de paralelização e sincronização, um conjunto de variáveis de ambiente e uma biblioteca de funções. Baseia-se na geração de regiões paralelas em trechos específicos da execução de um programa. Nessas regiões são criadas múltiplas *threads* que são executadas concorrentemente. Existem diretivas paralelas para execução paralela de laços, sendo possível especificar a forma de mapeamento das iterações do laço para as diversas *threads*, seja num esquema estático, em que blocos de iterações são divididos entre os *threads*, seja num esquema dinâmico, em que as iterações são atribuídas aos *threads* à medida que estes ficam livres, num esquema de rodízio. Em ambos os casos é possível especificar um tamanho de bloco. Quando a execução atinge uma região paralela do programa suas variáveis são visíveis a todos os *threads*. Entretanto, devido a eventuais dependências de dados, pode-se criar variáveis privadas de cada *thread*. Existem também diretivas de sincronização. Uma observação final é que o OpenMP pode ser utilizado conjuntamente como MPI, ou seja, o domínio do problema pode ser dividido para execução paralela pelos vários nós computacionais de um supercomputador usando MPI, podendo ainda ser subdividido para execução paralela pelos processadores/núcleos dos nós com MPI ou então com OpenMP. Uma grande vantagem é poder portar programas sequenciais para o OpenMP com a adição de apenas um conjunto de diretivas, sem alterar o código original sequencial. Por outro lado, um programa paralelo OpenMP pode ser compilado sequencialmente, pois essas diretivas são tratadas como comentários.

Um exemplo do uso de diretivas OpenMP, em Fortran 90, extraído deste trabalho, é:

```
!$OMP DO SCHEDULE (DYNAMIC, 2)
DO i=1, nob(ngrid)
```

...

END DO

Nesta diretiva, o "\$OMP DO" define um laço paralelo, abrindo uma região paralela a ser executada por múltiplos *threads*, delimitada pelo "END DO". A cláusula "SCHEDULE (DYNAMIC,2)" faz com que as iterações do laço sejam distribuídas para execução pelos *threads* duas a duas, num esquema de rodízio. A cláusula default seria correspondente ao "STATIC", em que as iterações são divididas em blocos entre os *threads*.

4.2. Padrão OPENACC

A programação de aceleradores de processamento surgiu com a API CUDA (KIRK, HWU, 2011) para a linguagem C, mas portar programas sequenciais para a execução paralela em GPGPUs não é uma tarefa trivial. Assim, nos últimos anos, foram propostos alguns padrões de API, propostos por consórcios compostos por fabricantes de processadores e desenvolvedores de software, para prover diretivas de paralelização a serem utilizadas em programas escritos nas linguagens C/C++ e também Fortran. É o caso do OpenCL, proposto em 2009 para o uso com linguagens C/C++, Python, Java, Julia, e destinado a GPGPUs, FPGAs e DSPs (processador digital de sinais, do inglês, *Digital Signal Processor*). Estes últimos aceleradores são microprocessadores específicos para tratamento digital de sinais, como o próprio nome indica. O padrão mais recente é o OpenACC (*Open Accelerators*, SC11, 2011) anunciado pela Nvidia, PGI (*The Portland Group*, que desenvolve compiladores), CAPS (um consórcio francês que desenvolve compiladores) e a Cray no evento *Supercomputing 2011* e lançado no *Supercomputing 2012* para acelerar códigos em C/C++ e Fortran, sendo inspirado no padrão OpenMP que provê paralelização baseada em *threads* para a execução paralela em máquinas de memória compartilhada com vários processadores multinúcleo. Mais recentemente, a Nvidia adquiriu a PGI, enquanto a Intel, que também

desenvolve compiladores e integra o consórcio que propôs o OpenMP, direciona esforços para que este padrão integre também o uso de aceleradores.

O padrão OpenACC provê diretivas de alto nível para paralelização, a exemplo do OpenMP, especificando regiões do código que devem ser executadas pelo acelerador considerado, como ilustrado na Figura 4.1. Assim, à semelhança do OpenMP, um código sequencial pode ser portado com relativa facilidade para a execução paralela com o uso de aceleradores. A título de exemplo, o mesmo trecho de código em Fortran 90 que foi mostrado na seção anterior com OpenMP, aparece abaixo portado para OpenACC (laço paralelo):

```
!$ACC PARALLEL  
  
DO i=1, nob(ngrid)  
  
...  
  
END DO
```

Na API CUDA, tem-se um programa sendo executado sequencialmente pela CPU com a chamada a uma função kernel que é executada na GPGPU. Na API OpenACC, como visto na figura acima, usa-se uma diretiva específica para criar uma região paralela que será executada no acelerador de processamento considerado. Assim como na API CUDA, a programação na API OpenACC exige o gerenciamento da memória local da GPGPU e do tráfego de dados entre esta e a CPU, sendo demandados:

- Alocação de memória na GPGPU;
- Cópia de dados da memória principal para a memória local da GPGPU;
- Cópia de dados da memória local da GPGPU para a memória principal;

No caso da API CUDA, o domínio do problema é particionado por uma grade uni ou bidimensional e sua execução é mapeada para blocos de *threads*. Os

blocos são executados pelos vários multiprocessadores que compõem a GPGPU, cada um composto por um conjunto de núcleos. Um parâmetro denominado *warp size* determina o número máximo de *threads* que pode ser executado simultaneamente num multiprocessador da GPGPU. Como exemplo, a GPGPU NVidia GTX 590, baseada na arquitetura NVidia Fermi, tem 1024 núcleos divididos fisicamente em 16 multiprocessadores, cada um com 64 núcleos, sendo o *warp size* de 32. Um problema hipotético cujo domínio pudesse ser dividido para execução paralela com 8192 *threads*, poderia ser mapeado para 128 blocos de 64 *threads* e estes mapeados para execução nos 16 multiprocessadores cabendo 8 blocos a cada um. Entretanto, cada multiprocessador executará seu conjunto de 8 blocos limitado à execução simultânea de 32 *threads*, ou seja, executará meio bloco de cada vez. No caso da API OpenACC, o domínio é mapeado para gangs de *workers*, sendo estes correspondentes às *threads*. Tanto na API CUDA como na API OpenACC, o desempenho paralelo da execução na GPGPU depende do mapeamento da divisão de domínio do problema para a execução em blocos/gangs com tamanho convenientemente definido pelo programador.

Uma vantagem complementar do OpenACC provém da possibilidade de uso com diferentes aceleradores, já que é prevista a portabilidade entre os diversos aceleradores definidos pelo seu padrão, os quais podem incluir futuramente placas gráficas de outros fabricantes além da Nvidia ou mesmo de aceleradores que não são placas gráficas. Por outro lado, a API CUDA presta-se apenas a placas gráficas da Nvidia e sua programação está limitada à linguagem C/C++, a menos que se utilize o compilador não gratuito CUDA Fortran da Portland Group. Consequentemente, no presente trabalho, adotou-se a API OpenACC.

4.2.1. Placas aceleradoras gráficas

As placas aceleradoras gráficas de uso geral (GPGPU, do inglês *General Purpose Graphics Processing Unit*) constituem uma das categorias dos chamados aceleradores de processamento, que são uma espécie de coprocessador complementar à CPU (ou aos seus núcleos), os quais têm capacidade de executar instruções inteiras e em ponto-flutuante a uma taxa muito maior que as CPUs. Dentre os aceleradores, além das GPGPUs, citam-se as famílias de coprocessadores Intel Phi, constituídos por dezenas de núcleos, também conhecido como MIC (*Intel Many Integrated Core Architecture*) e os FPGAs (*Field –Programmable Gate Array*), circuitos integrados que podem ser configurados pelos usuários para executar programas específicos. Arquiteturas em que a capacidade de processamento das CPUs (multinúcleo) são complementadas por esses aceleradores enquadram-se na categoria de computação heterogênea ou híbrida (KOGGE; DYSART, 2011), apresentando vantagens tais como o baixo custo de aquisição e operação (consumo de energia) por GFlop/s (bilhões de operações de ponto flutuante executadas por segundo, do inglês *Giga Floating Point Operations per Second*). No mercado de computação de alto desempenho podem ser citadas ainda outros tipos de processadores tais como os *cell processors*, microprocessadores multinúcleos desenvolvidos para a indústria de jogos por um consórcio formado por Sony, Toshiba e IBM, conhecido como STI, e os (mais atuais) processadores RISC (*Reduced Instruction Set Computer*) ARM, (da empresa inglesa ARM) desenvolvidos para celulares, *smartphones* e outros dispositivos eletrônicos, que tem arquitetura RISC e que estão disponíveis no mercado na forma de servidoras compostas por inúmeros processadores desse tipo com arquitetura 64-bit (tamanho da palavra da memória) e capacidade de processamento em ponto flutuante com precisão dupla.

As placas aceleradoras gráficas (GPU, do inglês *Graphics Processing Unit*) são processadores dedicados à geração de imagens *raster* para monitores gráficos, permitindo acelerar a execução de operações básicas de computação

gráfica tais como mapeamento, sombreamento, recorte de polígonos, etc. A primeira placa gráfica denominada como GPU foi a GeForce-256 da Nvidia, lançada em 1999 e, com a necessidade de apresentar na tela imagens e vídeos relativos a aplicativos tais como jogos eletrônicos, o uso de GPUs se intensificou bem como sua capacidade de processamento. A disponibilidade de tal capacidade fez com que programadores utilizassem as GPUs para processamento numérico, complementando as CPUs. Em função disto, a Nvidia lançou no mercado em 2007 um kit de desenvolvimento de software denominado CUDA (*Compute Unified Device Architecture*), que permite a chamada a rotinas que são executadas em GPGPUs da Nvidia a partir de programas escritos em linguagem C e executados na CPU utilizando a API (*Application Programming Interface*) do mesmo nome. Além disso, a GPGPU tem sua memória interna independente da memória principal da máquina, exigindo uma programação que minimize os atrasos devidos ao tráfego de dados entre essa memória e a memória principal da máquina, que ocorre via instruções específicas executadas na CPU.

O uso de aceleradores de processamento vem se difundindo, inclusive no mercado de supercomputação. Na lista mais recente, de junho de 2014, dos supercomputadores mais rápidos do mundo (www.top500.com), o mais rápido é o chinês Tianhe-2, com CPUs Intel Xeon e aceleradores Phi, totalizando 3.120.000 núcleos, obtendo 33,86 PFlop/s na execução do programa de *benchmark* Linpack. O segundo mais rápido, Titan (EUA) da família XK7, com CPUs AMD Opteron e GPGPUs Nvidia, tem pouco mais da metade do desempenho do Tianhe-2. Entre os dez mais rápidos, aparecem dois que utilizam Intel Xeon Phi e outros dois que utilizam GPGPUs Nvidia. Há uma disputa grande entre os fabricantes dos aceleradores da Nvidia e da Intel, a qual envolve as APIs disponíveis para utilizá-los, sendo difícil prever qual será a tendência do mercado de alto desempenho. Entretanto, é possível distinguir dois tipos de arquiteturas de computadores, sem e com aceleradores de processamento, conforme abaixo, respectivamente:

- Arquiteturas *Multicore*: máquinas com CPUs multinúcleo com frequências menores que 3 GHz e com miniaturização crescente (já há processadores com 12 núcleos);
- Arquiteturas *Manycore*: máquinas com CPUs multinúcleo que utilizam aceleradores de processamento com dezenas ou centenas de núcleos também com frequências menores que 3GHz, mas com unidades de controle e conjuntos de instruções mais simples que aqueles das CPUs, sendo os aceleradores GPGPUs e ou processadores Intel Phi;

Desde 2003, o desempenho em GFlop/s das GPGPUs têm aumentado mais que os das CPUs multinúcleos (KIRI; WUH, 2011), conforme ilustrado na Figura 4.1. A comparação é feita também para o desempenho em GFlop/s/watt na Figura 4.2, sempre considerando operações em precisão dupla.

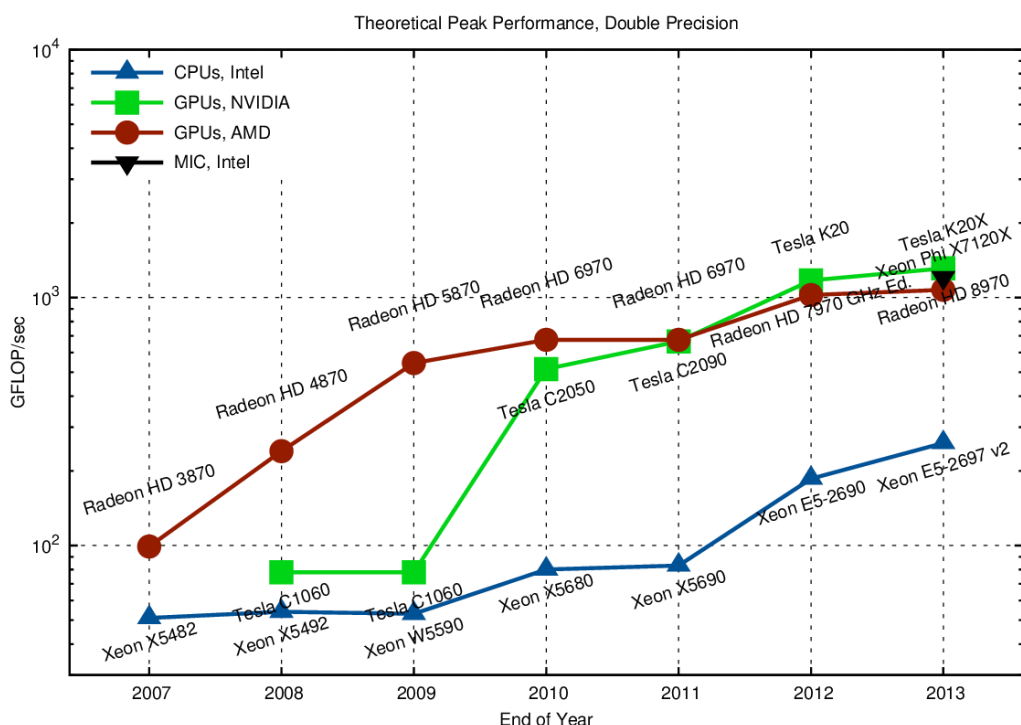


Figura 4.1: Evolução do desempenho em GFlop/s de processadores Intel, coprocessores Intel Phi e GPGPUs da Nvidia e AMD de 2007 a 2013.

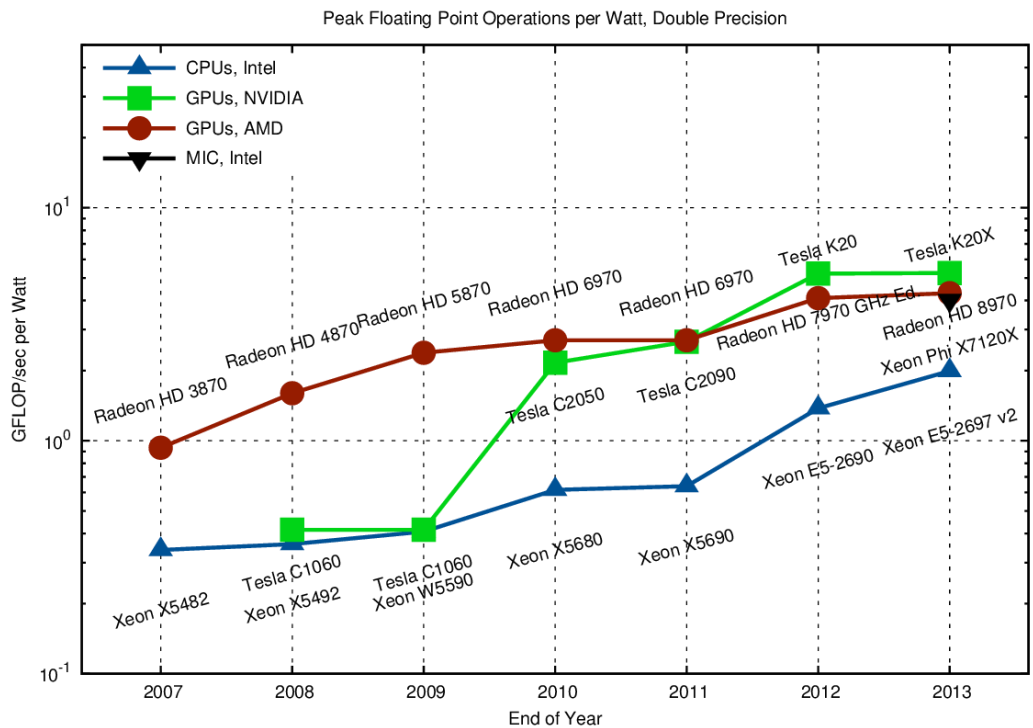


Figura 4.2: Evolução do desempenho em (GFlop/s)/watt de processadores Intel, coprocessadores Intel Phi e GPGPUs da Nvidia e AMD de 2007 a 2013

5 IMPLEMENTAÇÃO COMPUTACIONAL DO SOLVER ORIGINAL

O acoplamento do termo de reatividade química ao modelo CCATT-BRAMS é complexo, pois demanda a leitura da estrutura de dados relativa ao modelo BRAMS (FAZENDA, 2006), pois as funções de produção e de perda das espécies demandam valores atualizados de algumas variáveis atmosféricas, tais como pressão e temperatura. A complexidade deste acoplamento deve-se também à linguagem C adotada para o *solver* original, pois o CATT-BRAMS foi codificado em Fortran 90. A cada execução, o termo de reatividade química deve fornecer ao CCATT-BRAMS as concentrações atualizadas das gases traço instáveis correspondentes às espécies químicas consideradas no mecanismo RELACS para cada ponto de grade do modelo. Assim, há leitura de dados do CATT-BRAMS na execução do termo de reatividade química e vice-versa. A otimização do acesso à memória (no caso, à memória cache) correspondente a essas leituras e escritas é feita pela técnica clássica de *blocking*, no caso, pela divisão dos pontos de grade do modelo em blocos de tamanho fixo. No termo de reatividade química, cada bloco é armazenado como um vetor unidimensional. O *solver* original utiliza blocos de 64 pontos de grade. Será demonstrado neste trabalho que este é efetivamente o tamanho ótimo de bloco, considerando-se execução no supercomputador tupã. Um tamanho de bloco maior permitiria um aumento da granularidade do problema, permitindo acelerar os cálculos, mas por outro lado piora o acesso à memória, aumentando os tempos correspondentes. O termo de reatividade química processa os blocos de pontos de grade em sequência, conforme a divisão de domínio feita pelo MPI, ou seja, processa os blocos que foram mapeados para o nó computacional do supercomputador que está sendo considerado.

O termo de reatividade química inclui a função que implementa o método de Rosenbrock, que tem como entradas variáveis atmosféricas do modelo BRAMS, concentrações das espécies químicas no passo de tempo corrente, número de espécies químicas consideradas, número de pontos de grade por bloco, etc. É importante ressaltar que não há dependência entre pontos de

grade de um mesmo bloco (ou de blocos diferentes). Assim, torna-se possível resolver de maneira independente os sistemas de equações lineares associados à resolução do termo de reatividade química. Entretanto, no *solver* original, essa resolução não é paralelizada, não devido a dependências de dados, que inexistem entre pontos de grade, mas pelas razões já comentadas nas Seção 3.3.1 e 3.3.2. Essas razões são, respectivamente, a impossibilidade de se utilizar o padrão OpenMP devido ao CCATT-BRAMS ser codificado em Fortran 90 e as chamadas a funções da biblioteca Sparse 1.3 não poderem ser eliminadas por *inlining*, e a impossibilidade de se usar o MPI, uma vez que a grade do computacional do modelo já estar dividida por MPI. Considerando-se o padrão OpenMP, a paralelização na dimensão dos pontos de grade (mapear um ponto de grade para cada *thread*) não é possível, nem tampouco na dimensão dos blocos (mapear um bloco de pontos de grade para cada *thread*). Essas paralelizações explorariam a não dependência entre pontos de grade e, conseqüentemente, entre blocos. Finalmente, a paralelização "interna" ao *solver* original também fica inviável pela complexidade do código e pela sua baixa granularidade.

Apresenta-se a seguir o pseudocódigo da implementação do método de Rosenbrock, utilizando o *solver* original:

INICIALIZA estrutura de dados da *Sparse 1.3*

PARA CADA bloco **FAÇA** (laço dos blocos)

PARA CADA ponto de grade **FAÇA** (laço dos pontos de grade)

LEIA variáveis do BRAMS

Atualize as taxas de fotólise

FINALIZE (PARA CADA)

Calcule as reações cinéticas iniciais

ENQUANTO (passo de tempo < limite) **FAÇA**

Calcule o Jacobiano da matriz de concentrações

Atualize os termos de produção e perda (Eq. 3.2)

PARA CADA espécie químicas **FAÇA**

PARA CADA ponto de grade **FAÇA**

Atualize $F(\rho)$ na estrutura de dados

FINALIZE (PARA CADA)

FINALIZE (PARA CADA)

ENQUANTO (erro > tolerância) **FAÇA**

PARA CADA espécie químicas **FAÇA**

PARA CADA ponto de grade **FAÇA**

Atualize a matriz **A**

FINALIZE (PARA CADA)

FINALIZE (PARA CADA)

Atualize b_i

PARA CADA ponto de grade **FAÇA**

Calcule 1^o estágio do método de Rosenbrock

FINALIZE (PARA CADA)

Atualize b_i

PARA CADA ponto de grade **FAÇA**

Calcule 2º estágio do método de Rosenbrock

FINALIZE (PARA CADA)

Atualize a matriz de concentrações ρ

Atualize o termo de produção $F(\rho)$

Atualize b_i

PARA CADA ponto de grade **FAÇA**

Calcule 3º estágio do método de Rosenbrock

FINALIZE (PARA CADA)

Atualize a matriz de concentrações ρ

Atualize o termo de produção $F(\rho)$

Atualize b_i

PARA CADA ponto de grade **FAÇA**

Calcule 4º estágio do método de Rosenbrock

FINALIZE (PARA CADA)

Atualiza matriz de concentrações ρ

Calcule erro e *arredondamento*

SE (tolerância – *arredondamento* < 1.0) **FAÇA**

Aceite a solução

SENÃO

Calcule o novo passo de integração

FINALIZE (SE)

Atualize o passo de integração

FINALIZE (ENQUANTO)

FINALIZE (ENQUANTO)

Atualize as variáveis do BRAMS

FINALIZE (PARA CADA) Blocos

5.1. *Profile* do tempo de execução do termo de reatividade química com o *solver* original

Nos resultados aqui expostos, o modelo CCATT-BRAMS foi executado em um único passo de tempo de forma a prover os dados de entrada para o termo de reatividade química que foi assim executado na forma *stand alone*, ou seja, independentemente do restante do modelo CCATT-BRAMS. A máquina utilizada foi a Raven, situada nos EUA e cedida pela companhia Cray Inc., que é composta por:

- 20 nós do sistema XE6, cada um com 2 processadores AMD Opteron de 16 núcleos e 32 GB de memória;
- 8 nós do sistema XK7, cada um com 1 processador AMD Opteron de 16 núcleos, 1 GPGPU NVidia Kepler K20 Tesla e 6 GB de memória.

Em todos os testes computacionais foram utilizados nós do sistema XE6, exceto naqueles que utilizaram o padrão OpenACC. Embora fosse disponível um compilador da PGI, optou-se pelos compiladores da própria Cray, Fortran 90 (crayftn versão 8.3.2.104) e C (craycc versão 8.3.2.104), além da ferramenta de *profiling* perf tools da Cray, versão 6.2.1.12922. Este *profile* de tempo de execução foi gerado utilizando-se o modelo CCATT-BRAMS na versão 4.5 com uma resolução espacial correspondente a uma grade de 200 x 200 x 38 pontos, ou seja, um total de 1.520.000 pontos, mas para um único bloco de 64 pontos de grade e para um único passo de tempo do modelo. O *profile* de tempo de execução é mostrado na Tabela 5.1, em que as linhas com fundo branco correspondem aos tempos das funções diversas do termo de reatividade química, que atualiza as concentrações das espécies químicas pelo método de Rosenbrock para a resolução do sistema de equações diferenciais (Equação 3.2). Em fundo azul aparecem apenas os tempos correspondentes ao o *solver* original codificado na linguagem C para resolução do sistema de equações lineares (Equação 3.19). Foram gerados 3 *profiles* de tempo de execução, correspondentes a execuções independentes para o mesmo passo

de tempo do modelo. Nota-se que as chamadas ao *solver* original nos 4 estágios do métodos de Rosenbrock correspondem a mais de 70% do tempo total.

Tabela 5.1: *Profile* do tempo de execução do termo de reatividade química com o *solver* original para 3 execuções (T1, T2 e T3) correspondendo a um único passo de tempo do modelo CCATT-BRAMS e um único bloco de 64 pontos de grade (tempos em segundos). As linhas com fundo azul correspondem às chamadas ao *solver* original.

Tempos (s) de execução com o <i>solver</i> original.				
	T1	T2	T3	T_Medio
Leitura das variáveis	6.24E-05	4.68E-05	4.68E-05	5.20E-05
Atualização na estrutura de dados	9.36E-05	8.58E-05	8.58E-05	8.84E-05
Cálculo das reações Iniciais	1.17E-04	1.87E-04	1.25E-04	1.43E-04
Cálculo do Jacobiano	1.11E-03	9.67E-04	4.37E-04	8.37E-04
Atualização dos termos P e L	1.33E-04	2.50E-04	1.40E-04	1.74E-04
Atualização da matriz A	3.63E-03	3.57E-03	3.52E-03	3.57E-03
<i>solver</i> original (1º Estágio Rosenbrock)	3.08E-03	3.73E-03	3.08E-03	3.30E-03
<i>solver</i> original (2º Estágio Rosenbrock)	3.20E-03	3.16E-03	3.16E-03	3.17E-03
<i>solver</i> original (3º Estágio Rosenbrock)	3.42E-03	3.41E-03	3.44E-03	3.42E-03
<i>solver</i> original (4º Estágio Rosenbrock)	3.40E-03	3.38E-03	3.38E-03	3.39E-03
Atualização solução e ajuste de passo	1.79E-04	1.79E-04	1.79E-04	1.79E-04
Tempo Total (s):	1.84E-02	1.90E-02	1.76E-02	1.83E-02
Tempo <i>solver</i> original:	1.31E-02	1.37E-02	1.31E-02	1.33E-02
Porcentagem dos Tempos	71.11%	72.17%	74.23%	72.47%

6 IMPLEMENTAÇÃO COMPUTACIONAL DO NOVO SOLVER

O novo *solver* implementado no termo de reatividade química, baseado na teoria de estimação linear ótima, foi codificado em Fortran 90 resultando em apenas 60 linhas de código. Considerando os pseudocódigos apresentados na Seção 3.3.2, nota-se que o primeiro refere-se apenas ao cálculo dos vetores de projeção \mathbf{p} , calculados apenas no primeiro estágio de Rosenbrock para todos os pontos de grade do bloco que está sendo processado. Este cálculo dos vetores de projeção \mathbf{p} demanda uma parcela grande do tempo total do novo *solver*. Nos demais estágios, apenas o termo independente b_i é atualizado e os vetores solução de cada estágio são usados como solução inicial no estágio seguinte acelerando a convergência da solução. Os erros resultantes nas concentrações das espécies químicas são da ordem de 10^{-11} , ligeiramente menores que os erros do *solver* original, que foram da ordem de 10^{-8} . Constatou-se também que ajustes do passo de tempo de integração nos estágios do método Rosenbrock são menos frequentes do que no *solver* original, contribuindo para o bom desempenho computacional do novo *solver*.

Os testes computacionais mostrados neste capítulo foram executados na máquina Raven descrita no capítulo anterior e utilizando o compilador Cray Fortran 90 (crayftn) e a ferramenta de *profiling* perftools. A Tabela 6.1 compara os tempos de processamento das execuções sequenciais de ambos os *solvers* considerando a resolução do termo de reatividade química para um passo de tempo do modelo e para um único bloco de 64 pontos de grade. Observa-se que o novo *solver* demanda apenas cerca de 35% de tempo a mais que o *solver* original. Entretanto, esses tempos foram específicos de um bloco de 64 pontos de grade. Eventualmente, para outros blocos, sistemas de equações lineares diferentes terão que ser resolvidos e a estabilidade numérica do novo *solver* pode acarretar tempos de processamento menores do que os demandados pelo *solver* original. A Tabela 6.2 apresenta o número de ajustes de passo requeridos pelos dois *solvers*, sendo que a maioria absoluta dos

blocos requereu apenas um ajuste de passo para os pontos de grade contidos em cada bloco.

Além disso, o novo *solver* pode ser paralelizado, como será demonstrado adiante.

Tabela 6.1: Tempos de execução dos *solvers* considerando um passo de tempo do modelo e um bloco de 64 pontos de grade. (tempos em segundos).

Comparação Novo Solver Vs Solver Original			
Tempo Novo Solver (s)		Tempo Solver Original (s)	
Estágio 1	1.62E-02	Estágio 1	3.30E-03
Estágio 2	5.41E-04	Estágio 2	3.17E-03
Estágio 3	6.27E-04	Estágio 3	3.42E-03
Estágio 4	6.14E-04	Estágio 4	3.39E-03
Total	1.79E-02	Total	1.33E-02

Tabela 6.2: Número de blocos em função do número de ajustes de passo necessários nos estágios do método de Rosenbrock e tempos de execução totais para ambos os *solvers* considerando um passo de tempo do modelo e todos os 23.750 blocos de pontos de grade (tempos em segundos).

Novo solver		Solver original	
Nº de Ajustes De Passo	Nº de Blocos	Nº de Ajustes De Passo	Nº de Blocos
1	21917	1	21553
2	1408	2	1653
3	300	3	384
4	109	4	139
5	16	5	20
6	0	6	1
7	0	7	0
Total de Blocos (s)	23750	Total de Blocos (s)	23750
Tempo Rosenbrock(s)	468,8697	Tempo Rosenbrock (s)	354,3224

A seguir, apresenta-se o profile de tempo de execução do *solver* novo na Tabela 6.3, para execução do CCATT-BRAMS, versão 4.5, na máquina Raven (nó bprocessado) com uma resolução espacial correspondente a uma grade de 200 x 200 x 38 pontos, ou seja, um total de 1.520.000 pontos, mas para um único bloco de 64 pontos de grade e para um único passo de tempo do modelo. Nessa tabela, as linhas com fundo branco correspondem aos tempos das funções diversas do termo de reatividade química, que atualiza as concentrações das espécies químicas pelo método de Rosenbrock para a resolução do sistema de equações diferenciais (Equação 3.2). Em fundo azul aparecem apenas os tempos correspondentes ao novo *solver* codificado em Fortran 90 para resolução do sistema de equações lineares (Equação 3.6). Foram gerados 3 profiles de tempo de execução, correspondentes a execuções independentes para o mesmo passo de tempo do modelo e o mesmo bloco. Nota-se que as chamadas ao novo *solver* original nos 4 estágios do método de Rosenbrock correspondem quase 80% do tempo total (eram pouco mais de 70% com o *solver* original).

Tabela 6.3: *Profile* do tempo de execução do termo de reatividade química com o novo *solver* para 3 execuções (T1, T2 e T3) correspondendo a um único passo de tempo do modelo CCATT-BRAMS e um único bloco de 64 pontos de grade (tempos em segundos). As linhas com fundo azul correspondem às chamadas ao novo *solver*.

Tempos (s) de execução com o novo <i>solver</i> .				
	T1	T2	T3	T_Medio
Leitura das variáveis	6.24E-05	4.68E-05	4.68E-05	5.20E-05
Atualização na estrutura de dados	9.36E-05	8.58E-05	8.58E-05	8.84E-05
Cálculo das reações Iniciais	1.17E-04	1.87E-04	1.25E-04	1.43E-04
Cálculo do Jacobiano	1.11E-03	9.67E-04	4.37E-04	8.37E-04
Atualização dos termos P e L	1.33E-04	2.50E-04	1.40E-04	1.74E-04
Atualização da matriz A	3.63E-03	3.57E-03	3.52E-03	3.57E-03
Novo <i>solver</i> (1º Estágio Rosenbrock)	1.61E-02	1.62E-02	1.62E-02	1.62E-02
Novo <i>solver</i> (2º Estágio Rosenbrock)	5.38E-04	5.38E-04	5.46E-04	5.41E-04
Novo <i>solver</i> (3º Estágio Rosenbrock)	6.16E-04	6.55E-04	6.09E-04	6.27E-04
Novo <i>solver</i> (4º Estágio Rosenbrock)	6.16E-04	6.09E-04	6.16E-04	6.14E-04
Atualização solução e ajuste de passo	1.79E-04	1.79E-04	1.79E-04	1.79E-04
Tempo Total (s):	2.32E-02	2.32E-02	2.25E-02	2.30E-02
Tempo <i>solver</i> original (s):	1.79E-02	1.80E-02	1.79E-02	1.79E-02
Porcentagem dos Tempos	77.09%	77.28%	79.81%	78.04%

6.1. Paralelização usando o padrão OpenMP

Considerando-se o termo de reatividade química, a abordagem mais intuitiva é a paralelização na dimensão dos pontos de grade, uma vez que a resolução do sistema de equações lineares é independente para cada ponto de grade. Para cada estágio do método de Rosenbrock devem ser resolvidos os correspondentes sistemas para todos os pontos de grade. Entretanto, a implementação do termo de reatividade química recorre a técnica de *blocking* para otimizar o acesso à memória, dividindo os pontos de grade em blocos de tamanho fixo que tem posições contíguas na memória. Assim, resolvem-se os pontos de grade de sucessivos blocos, que contém entre 64 e 65 pontos de grade. No caso do uso da interface de programação OpenMP, que explora o paralelismo por *threads* para arquiteturas de memória compartilhada, é

mapeada para cada *thread* um conjunto de pontos de grade pertencentes ao bloco que está sendo processado. A seguir, discutem-se as várias implementações que foram testadas.

Conforme mencionado antes, a grade tridimensional é dividida em blocos de 64 pontos e a atualização das concentrações das espécies químicas (i.e. a resolução do termo de reatividade química) pode ser feito independentemente para cada ponto de grade. A versão paralela corrente em MPI do modelo CCATT-BRAMS divide o domínio entre os nós computacionais com partições da grade horizontal, conforme mostrado na Figura 6.1. Neste esquema, cabe a cada nó computacional a execução para um conjunto de blocos que são processados em sequência. Assim, na primeira versão da paralelização com OpenMP optou-se por paralelizar na dimensão dos pontos de grade de um mesmo bloco, mapeando cada ponto de grade para cada *thread*, conforme ilustrado na mesma figura.

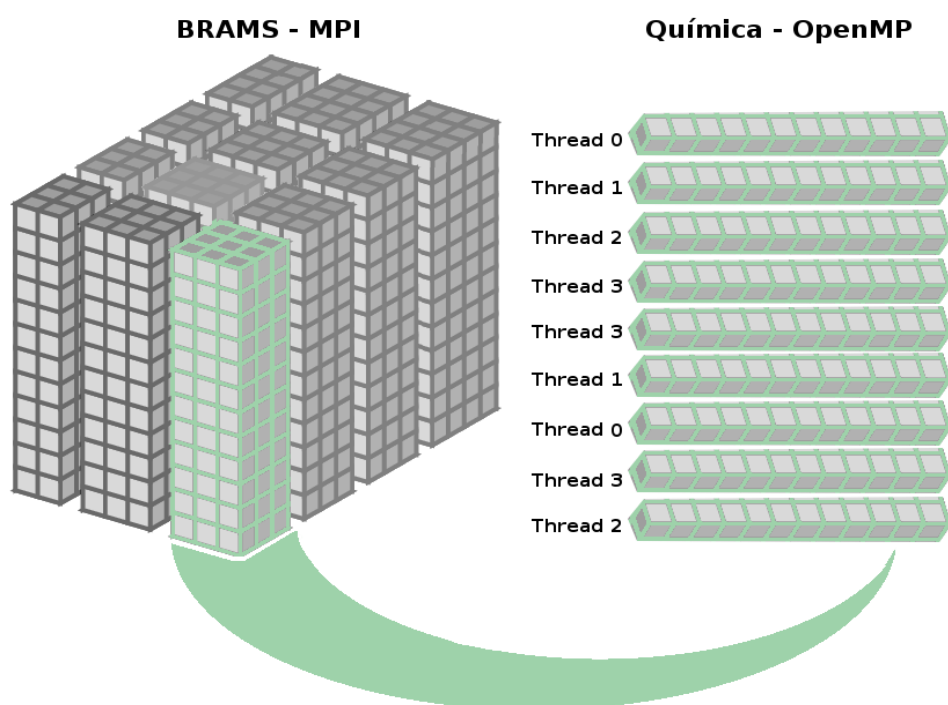


Figura 6.1: Figura esquemática da divisão do domínio em MPI em blocos de pontos de grade e o mapeamento de pontos de grade de um bloco para execução por *threads* em OpenMP da resolução do termo de reatividade química.

6.1.1. Primeira versão OpenMP

O novo *solver* pode ser implementado com poucas linhas de código, relativamente ao *solver* original, e não contém chamadas a funções, facilitando a definição de variáveis privadas e compartilhadas na paralelização com OpenMP. Variáveis locais ao *solver* foram tratadas como *private* (variáveis privadas), sendo os vetores e matrizes que descrevem o sistema linear e que contém a aproximação do resultado para cada estágio do método de Rosenbrock tratados como variáveis compartilhadas. A preparação dos dados demanda pouco tempo do processamento segundo o *profiling* realizado e, dessa forma, foi implementado sequencialmente. Tomando como base o pseudocódigo do novo *solver* (Seção 3.2.2) essa primeira versão em OpenMP tem o pseudocódigo a seguir, em que são criadas 4 regiões paralelas correspondentes a laços paralelos para os pontos de grade do bloco para cada estágio do método de Rosenbrock.

PARA CADA bloco **FAÇA** (laço dos blocos)

ENQUANTO (erro > tolerância) **FAÇA**

INICIALIZE (OpenMP)

PARA CADA ponto de grade **FAÇA** (laço dos pontos de grade)

Calcule 1^o estágio do Método de Rosenbrock

FIM (PARA CADA)

FINALIZE (OpenMP)

INICIALIZE (OpenMP)

PARA CADA ponto de grade **FAÇA** (laço dos pontos de grade)

Calcule 2^o estágio do Método de Rosenbrock

FIM (PARA CADA)

FINALIZE (OpenMP)

INICIALIZE (OpenMP)

PARA CADA ponto de grade **FAÇA** (laço dos pontos de grade)

Calcule 3^o estágio do Método de Rosenbrock

FIM (PARA CADA)

FINALIZE (OpenMP)

INICIALIZE (OpenMP)

PARA CADA ponto de grade **FAÇA** (laço dos pontos de grade)

Calcule 4^o estágio do Método de Rosenbrock

FIM (PARA CADA)

FINALIZE (OpenMP)

Calcula erro

FINALIZE (ENQUANTO)

FINALIZE (PARA CADA)

Essa versão OpenMP (Fortran 90) foi compilada e executada num nó biprocessado da máquina Raven e executada com até 32 *threads*, sempre considerando a execução de um passo de tempo do modelo e todos os 23.750 blocos da grade de 200 x 200 x 38 pontos. Entretanto, essa versão obteve bom desempenho paralelo somente até 4 *threads*, com a eficiência caindo abaixo de 50% acima de 8 *threads*, conforme pode-se observar nas Figuras 6.2, 6.3, 6.4 e 6.5, correspondentes ao tempo de execução, ao *speed-up*, à eficiência e ao número de operações em ponto flutuante executadas por segundo, respectivamente. Isso pode ser interpretado como sendo decorrente da baixa granularidade do problema, uma vez que um bloco de 64 pontos resultaria no mapeamento de apenas 8 pontos de grade para cada *thread*, aproximadamente.

Na paralelização dos 4 laços de pontos do bloco, relativas aos 4 estágios do método de Rosenbrock, foi utilizada a cláusula SCHEDULE (STATIC) do OpenMP, que divide estaticamente os pontos de grade do bloco entre os *threads*. A implementação do método de Rosenbrock faz com que os processamentos para os pontos de grade sejam iguais e, além disso, a atribuição do mesmo número de pontos de grade para cada *thread* assegura o balanceamento de carga em cada região paralela correspondente a cada estágio do método, justificando essa distribuição estática dos pontos de grade para os *threads*. No final do quarto estágio, se o erro médio exceder um limiar especificado, todos os pontos de grade do bloco terão que ser reprocessados, repetindo-se então os 4 estágios, mas mantendo-se o balanceamento de carga.

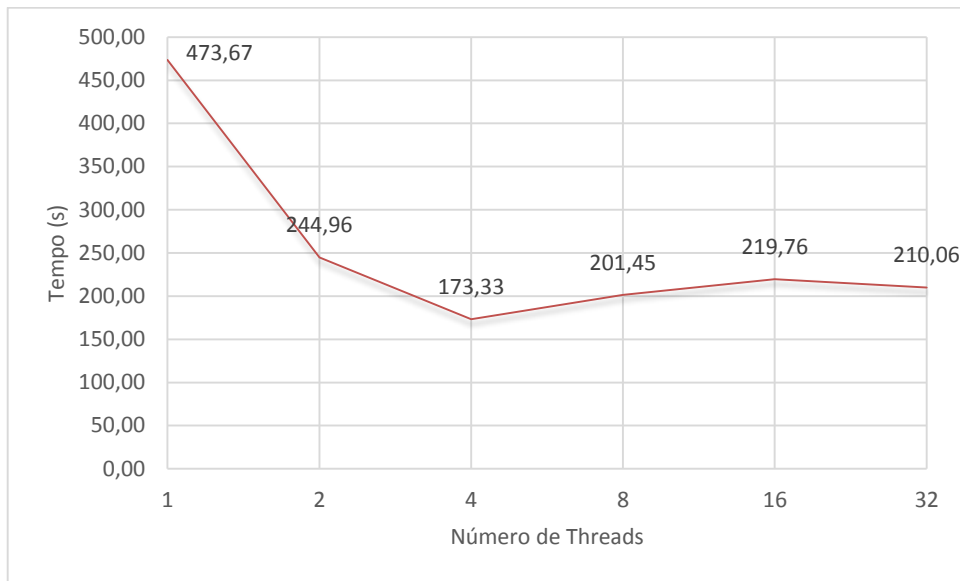


Figura 6.2: Tempo de execução da versão 1-OMP do novo *solver* em função do número de *threads*.

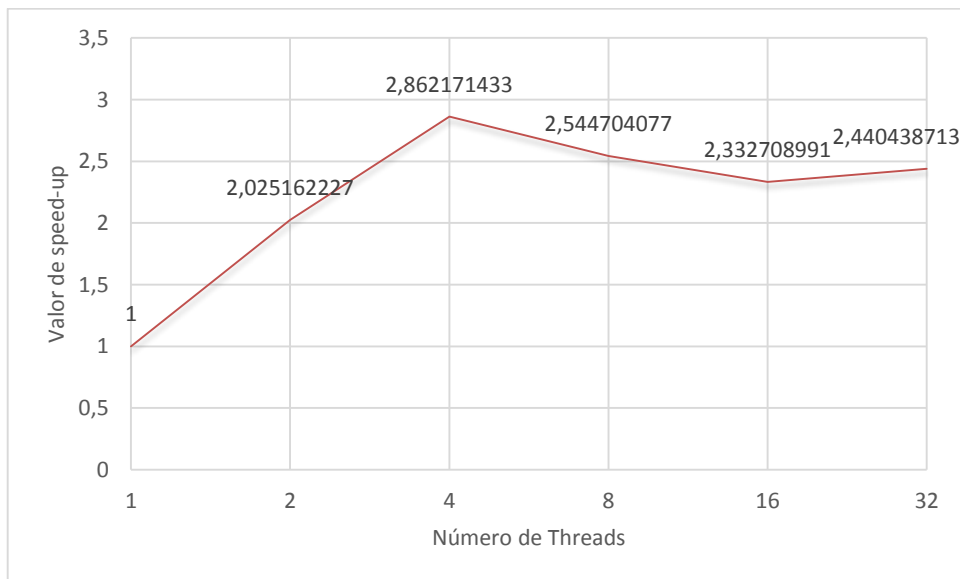


Figura 6.3: *Speed-up* da versão 1-OMP do novo *solver* em função do número de *threads*.

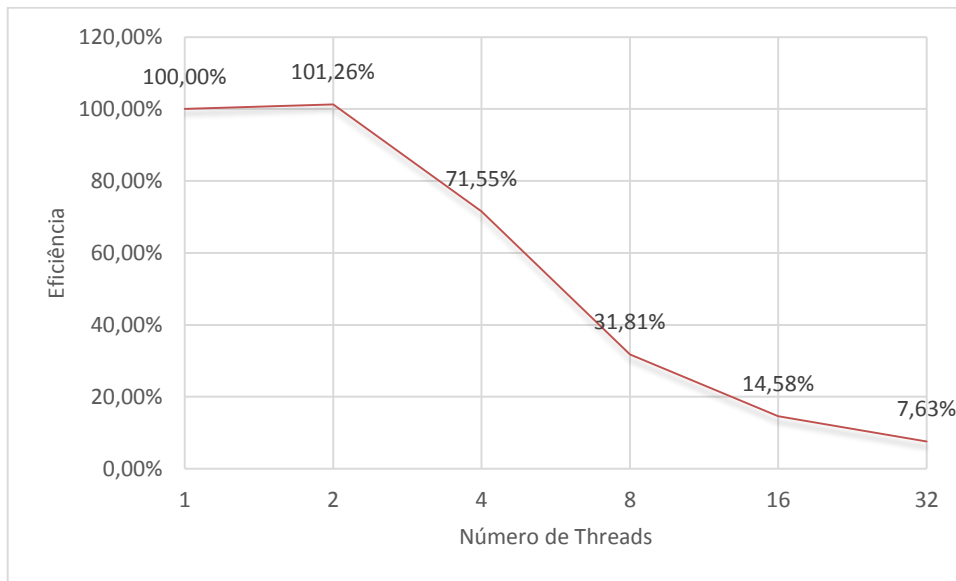


Figura 6.4: Eficiência da versão 1-OMP do novo *solver* em função do número de *threads*.

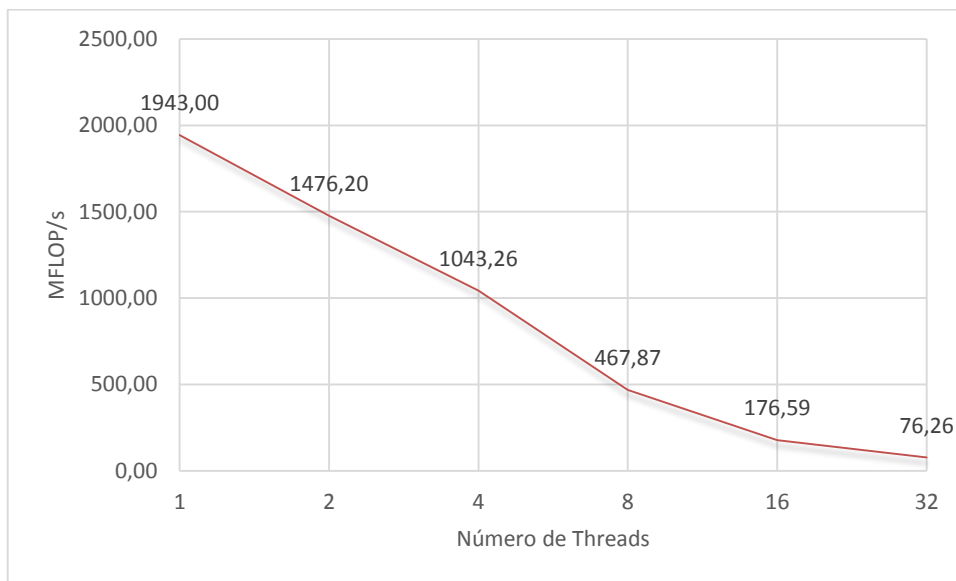


Figura 6.5: Taxa de operações em ponto flutuante da versão 1-OMP do novo *solver* em função do número de *threads*.

6.1.2. Influência do tamanho de bloco

Conforme mencionado anteriormente, o termo de reatividade química calculado pelo modelo ambiental CCATT-BRAMS utiliza a técnica de *blocking* para otimizar o acesso à memória, em termos de que um bloco inteiro possa caber em um dos níveis de memória cache, evitando acessos mais demorados à memória principal da máquina ou, ainda mais demorado, à área de *swap* que implementa a memória virtual. Na seção anterior, constatou-se que o mau desempenho paralelo era possivelmente causado pela baixa granularidade, sendo esta decorrente do pequeno tamanho do bloco (64 pontos de grade). Assim, modificou-se o modelo CCATT-BRAMS para gerar saídas com diferentes tamanhos de blocos. Estas saídas são utilizadas como dados de entrada para o termo de reatividade química *stand alone*.

Já na execução serial do modelo CCATT-BRAMS usando o *solver* original, pôde-se verificar um aumento no tempo quando executado de forma serial na máquina Tupã, com arquitetura XE6, fabricada pela Cray Inc., conforme observado na Figura 6.6, onde aparece a razão entre os tempos de processamento do modelo CCATT-BRAMS executado no Tupã, expressos em função do tamanho de bloco, para (i) execuções de um passo de tempo com chamadas ao termo de reatividade química e (ii) sem essas chamadas. Essas duas curvas são independentes, sendo a razão unitária correspondente à execução com bloco de 64 pontos. Percebe-se, neste caso, um aumento linear do tempo de processamento do modelo com o aumento do tamanho dos blocos, principalmente na curva correspondente à execução com chamadas à química, em que esse tempo chega até a triplicar. Assim, tamanhos de blocos maiores afetam o desempenho de acesso à memória.

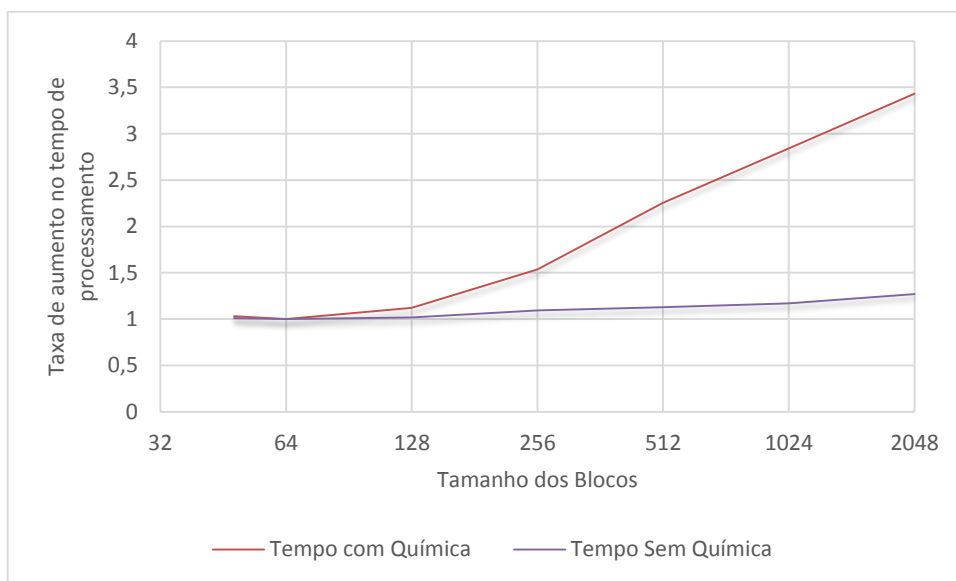


Figura 6.6: Aumento relativo do tempo de processamento do modelo CCATT-BRAMS em função do tamanho de bloco para execuções com chamadas ao termo de reatividade química (curva vermelha), e para execuções sem essas chamadas (curva azul).

A seguir, na figura 6.7, mostram-se os tempos de processamento sequencial do termo de reatividade química na máquina Raven expressos em função do tamanho de bloco, para um passo de tempo do modelo. Nota-se um aumento expressivo do tempo de processamento a partir de blocos de tamanho 256. Além disso, o tamanho ótimo na Raven de bloco parece ser 48 pontos, diferentemente da Tupã, em que esse tamanho é de 64 pontos.

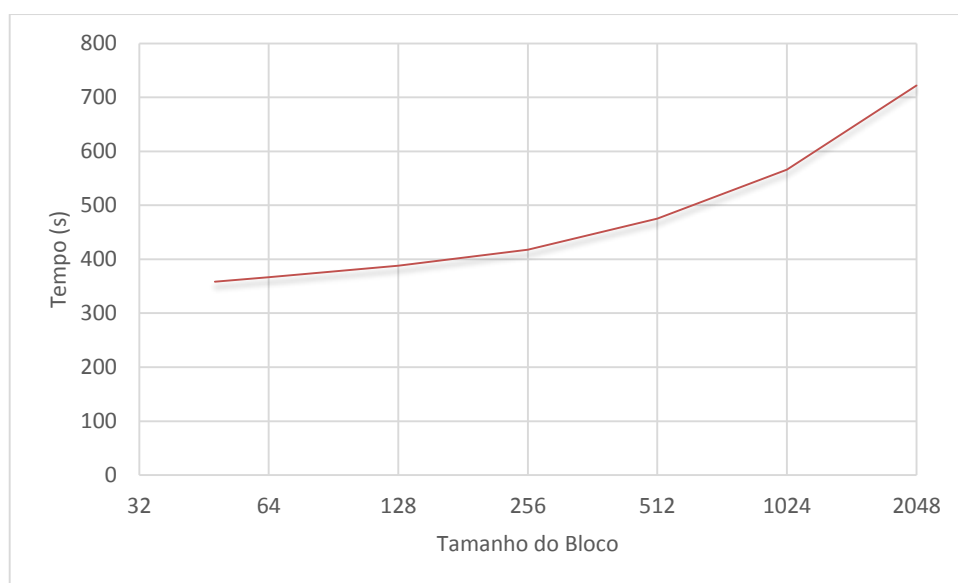


Figura 6.7: Tempos de processamento do termo de reatividade química na máquina Raven expressos em função do tamanho de bloco, para execuções de um passo de tempo.

As Figuras 6.2 a 6.5 mostraram o desempenho paralelo da versão 1-OMP do novo *solver* executado na Raven, mas sempre considerando um tamanho de bloco de 64 pontos. As figuras a seguir, Figuras 6.8, 6.9 e 6.10, mostram respectivamente o tempo de processamento, *speed-up* e eficiência da versão 1-OMP em função do tamanho de bloco. As conclusões são semelhantes, com desempenho paralelo aceitável até 4 *threads*, com a eficiência caindo abaixo de 50% acima de 8 *threads*. O tamanho ótimo de bloco para a Raven é de 48 pontos. Essa baixa escalabilidade pode ser atribuída à baixa granularidade do problema. Entretanto, isso não é compensado pelo uso de blocos maiores pois

o acesso à memória é muito penalizado com o aumento do tamanho do bloco, como pode ser visto nos tempos sequenciais para diferentes tamanhos de bloco. O baixo desempenho paralelo desta versão 1-OMP, que paraleliza na dimensão dos pontos de grade de um bloco, poderia talvez ser melhorado numa versão que explorasse a paralelização na dimensão dos blocos. Isso é mostrado na seção seguinte. Nestas figuras, *speed-up* e eficiência sempre foram calculados em relação à versão sequencial correspondente ao mesmo tamanho de bloco.

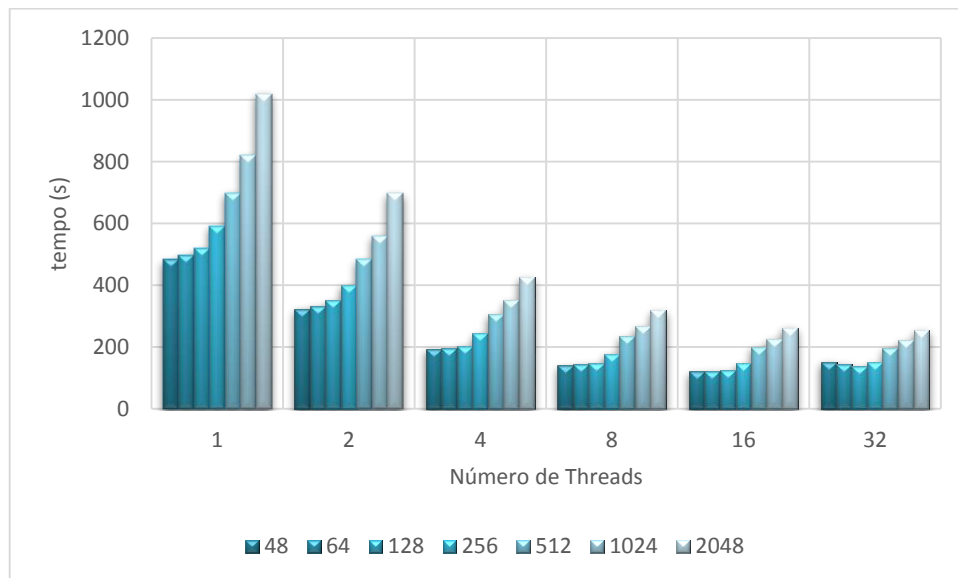


Figura 6.8: Tempo de execução da versão 1-OMP do novo *solver* em função do número de *threads* para diferentes tamanhos de bloco.

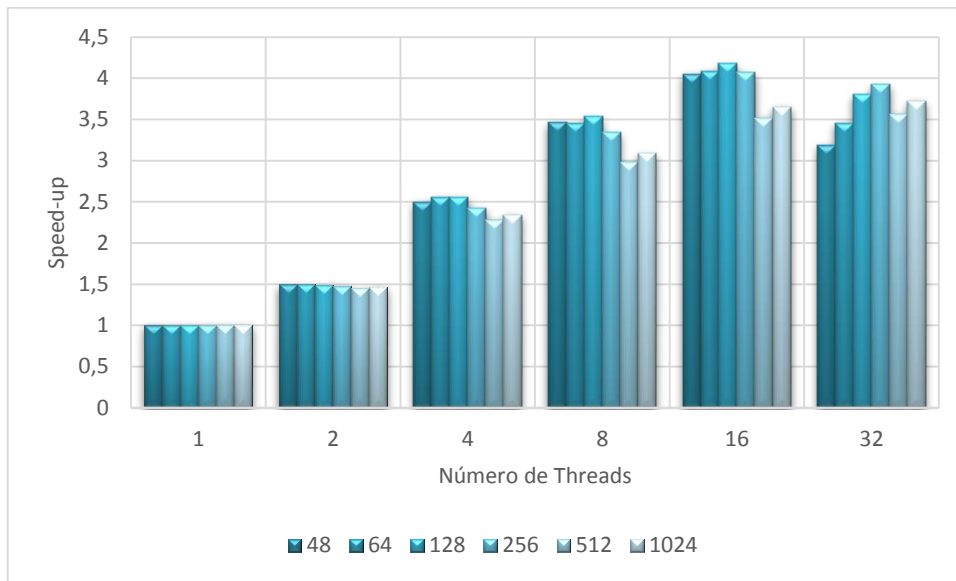


Figura 6.9: *Speed-up* da versão 1-OMP do novo *solver* em função do número de *threads* para diferentes tamanhos de bloco.

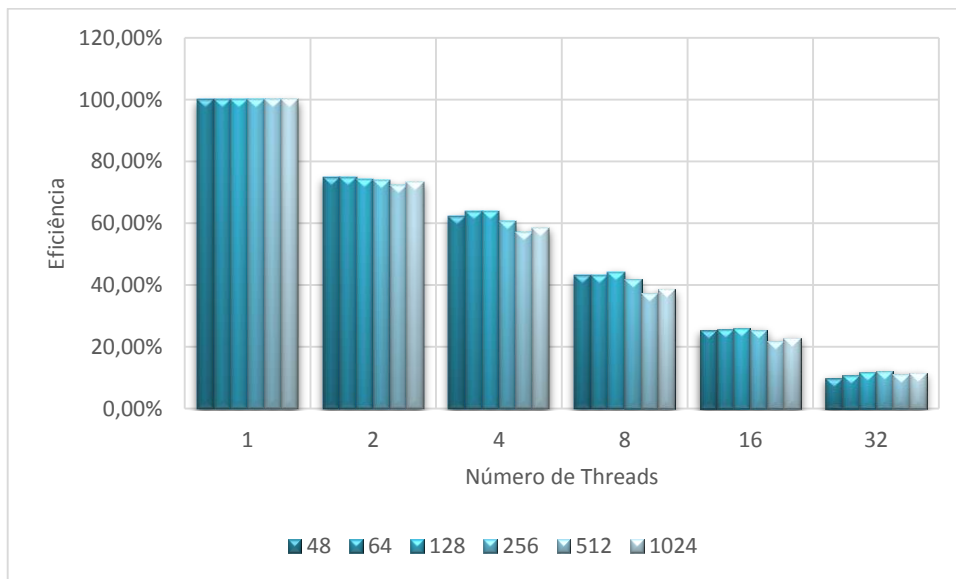


Figura 6.10: Eficiência da versão 1-OMP do novo *solver* em função do número de *threads* para diferentes tamanhos de bloco.

6.1.3. Segunda versão OpenMP

Na tentativa de aumentar a granularidade sem sacrificar o desempenho de acesso à memória, implementou-se uma segunda versão OpenMP do novo *solver* com paralelização na dimensão dos blocos. Executou-se na máquina Raven o termo de reatividade química para um passo de tempo e para a mesma grade de 200 x 200 x 38 pontos correspondente a 23.750 blocos para diferentes tamanhos de blocos. No novo pseudocódigo, apresentado a seguir, nota-se a criação de apenas uma única região paralela, correspondente ao laço dos blocos, sendo vários blocos alocados para cada *thread*. Conforme demonstrado por testes computacionais, o esquema mais eficiente para distribuição dos blocos no laço paralelo dos blocos foi SCHEDULE (DYNAMIC), com *chunk size* unitário, ou seja, alocando um bloco de pontos para cada *thread* de cada vez. Foram feitos diversos testes com *chunk sizes* diferentes, constatando-se que, para qualquer número de *threads*, o melhor seria o unitário.

Essa opção justifica-se pelo método de Rosenbrock: processa-se um bloco e, caso o erro médio dos pontos de grade do bloco seja excedido, reduz-se o tamanho do passo de integração e refaz-se completamente o cálculo do bloco, i.e. executam-se novamente todos os 4 estágios do método para todos os pontos de grade do bloco. Assim, blocos diferentes podem demandar tempo de processamento muito diferentes. Como esta versão OpenMP do novo *solver* aloca vários blocos para cada *thread*, pode ocorrer desbalanceamento de carga. O esquema de distribuição dinâmico de blocos para *threads* atribui um novo bloco para cada *thread* disponível, num esquema de rodízio. Isso garante que cada *thread* processe uma sequência de blocos ininterruptamente, mesmo que tenha que recalcular determinados blocos, diminuindo o desbalanceamento de carga.

INICIALIZE (OpenMP)

PARA CADA bloco **FAÇA** (laço dos blocos)

ENQUANTO (erro > tolerância) **FAÇA**

PARA CADA ponto de grade **FAÇA** (laço dos pontos de grade)

Calcule 1º estágio do Método de Rosenbrock

FIM (PARA CADA)

PARA CADA ponto de grade **FAÇA** (laço dos pontos de grade)

Calcule 2º estágio do Método de Rosenbrock

FIM (PARA CADA)

PARA CADA ponto de grade **FAÇA** (laço dos pontos de grade)

Calcule 3º estágio do Método de Rosenbrock

FIM (PARA CADA)

PARA CADA ponto de grade **FAÇA** (laço dos pontos de grade)

Calcule 4º estágio do Método de Rosenbrock

FIM (PARA CADA)

Calcula erro

FINALIZE (ENQUANTO)

FINALIZE (PARA CADA)

FINALIZE (OpenMP)

Essa segunda versão OpenMP (Fortran 90) do novo solver passa a ser referida como 2-OMP. O desempenho paralelo da execução do termos de reatividade química com essa versão do novo solver pode ser observado nas Figuras 6.11, 6.12 e 6.13, que mostram respectivamente o tempo de processamento, o *speed-up* e a eficiência para diferentes tamanhos de blocos. Nota-se que menores tempos são obtidos com menores tamanhos de bloco, embora a

diferença entre blocos de 48 e blocos de 64 pontos seja mínima. Além disso, os menores tempos de processamento foram obtidos com o maior número de *threads* (32) e as eficiências, embora decrescentes com o aumento do número de *threads*, não caíram muito abaixo dos 50% (47% para 32 *threads* e bloco de 64 pontos). Assim, consegue-se a versão 2-OMP do novo *solver* permite diminuir muito o tempo de processamento do termo de reatividade. No caso de 32 *threads* e bloco de 64 pontos, o tempo sequencial de 470 s é reduzido 16 vezes para cerca de 30 s. Numa eventual execução na Tupã (nós com 24 núcleos), estima-se uma redução de 12 vezes.

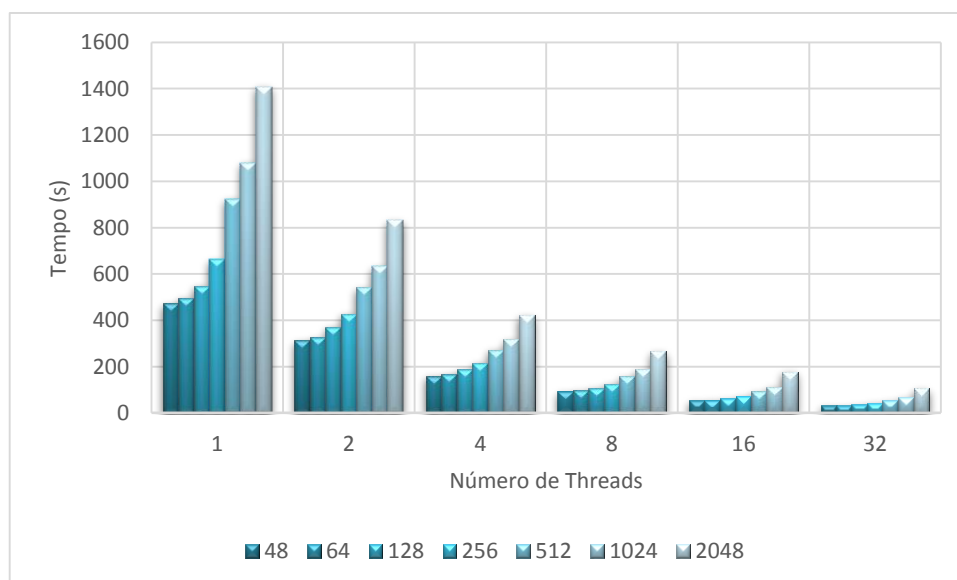


Figura 6.11: Tempo de execução da versão 2-OMP do novo *solver* em função do número de *threads* para diferentes tamanhos de bloco.

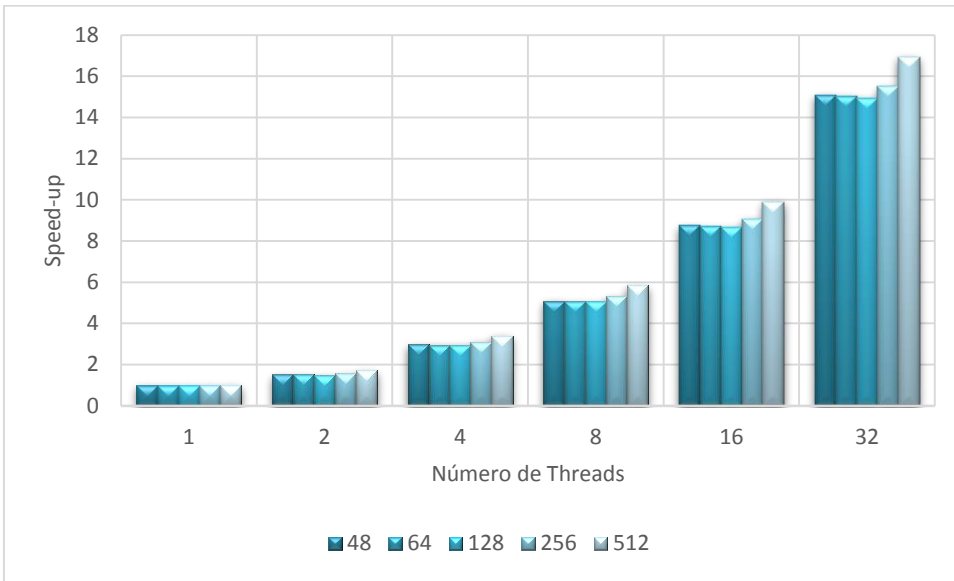


Figura 6.12: *Speed-up* da versão 2-OMP do novo *solver* em função do número de *threads* para diferentes tamanhos de bloco.

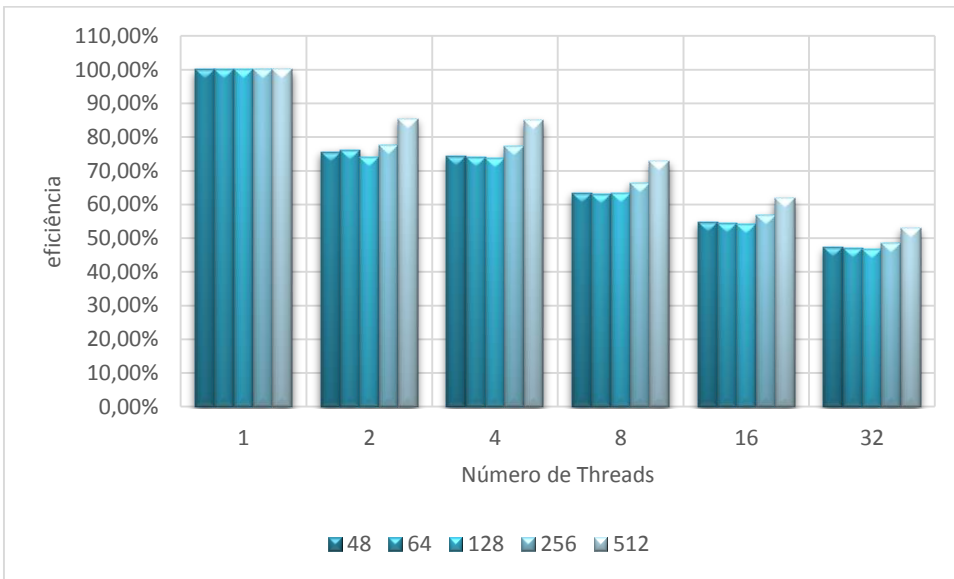


Figura 6.13: Eficiência da versão 2-OMP do novo *solver* em função do número de *threads* para diferentes tamanhos de bloco.

6.1.4. Balanceamento de carga

Comentou-se nas seções anteriores que o tempo de processamento para resolver o termo de reatividade química pode variar muito de um ponto de grade para outro em função do número de ajustes de passo necessários, conforme a Tabela 6.2. Esses ajustes de passo diminuem o passo de integração acarretando mais iterações em cada um dos estágios do método de Rosenbrock, que foi implementado levando-se em conta o erro médio para todos os pontos do bloco. Caso este erro exceda um limiar, o passo é diminuído e todos os 4 estágios do método de Rosenbrock são recalculados para todos os pontos de grade do bloco. A Figura 6.14 ilustra a grande variabilidade dos tempos de execução dos blocos, que variam pelo menos uma ordem de grandeza.

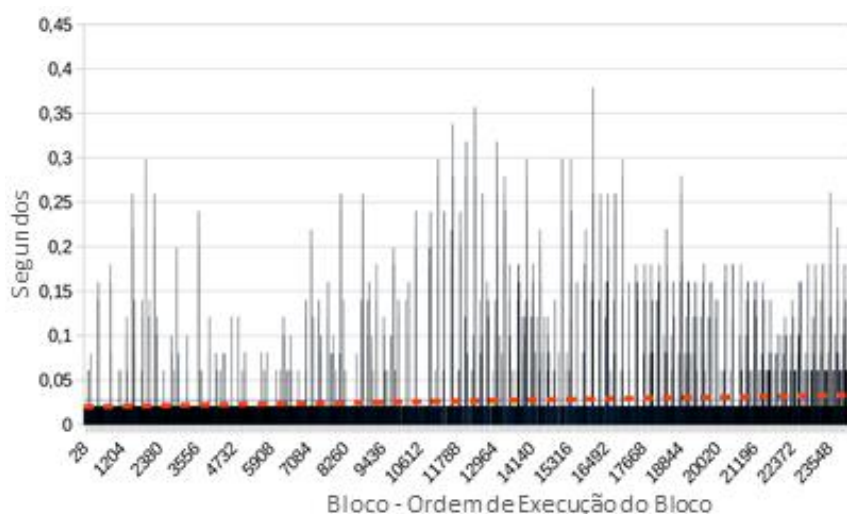


Figura 6.14: Tempos de processamento dos blocos para a grade completa do CCATT-BRAMS.

Isso acarreta um eventual desbalanceamento de carga que foi parcialmente neutralizado na versão 2-OMP com a adoção no laço paralelo do OpenMP da cláusula SCHEDULE(DYNAMIC), conforme comentado na seção anterior. O desempenho paralelo melhor da versão 2-OMP foi melhor que o da versão 1-

OMP, o que pode ser atribuído à maior granularidade da versão 2-OMP, mesmo considerando-se o mesmo tamanho de bloco.

Graham (1969) referiu-se à paralelização de tarefas do tipo "*Bag of Tasks*" alegando que um melhor desempenho paralelo poderia ser obtido ordenando-se as tarefas em ordem do custo computacional e executando-as em ordem decrescente de custo. Isso pode ser justificado porque ao se distribuir tarefas com custos semelhantes entre os processos ou *threads* melhora-se o balanceamento de carga. Obviamente, essa abordagem pressupõe a possibilidade de ordenar as tarefas, o que nem sempre é possível. Entretanto, no problema em questão neste trabalho, a ordenação dos blocos abrange primeiramente regiões com poucas emissões e que implicam em blocos de processamento rápido e termina com regiões em que pode se esperar o oposto, devido à atividade antropogênica (muitas emissões, muitas espécies químicas presentes, etc.). Assim, testou-se o desempenho invertendo a ordem de percorrimento dos blocos no laço paralelo da versão 2-OMP. Houve uma pequena melhora do desempenho paralelo em relação ao teste anterior, considerando-se a execução para a mesma grade de 23.750 blocos de tamanho 640, conforme pode ser visto nas Figuras 6.15, 6.16 e 6.17, que mostram respectivamente o tempo de processamento, o speed-up e a eficiência. Esta última chegou a superar 60% para 32 *threads*.

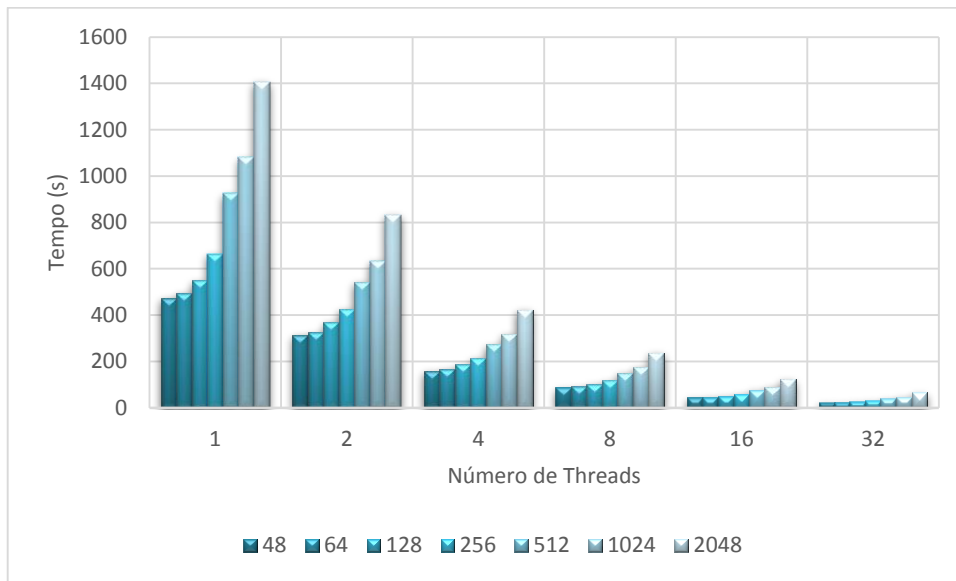


Figura 6.15: Tempo de execução da versão 2-OMP do novo *solver* em função do número de *threads* para o laço dos blocos percorrido na ordem original (curva azul) e na ordem inversa (curva vermelha)

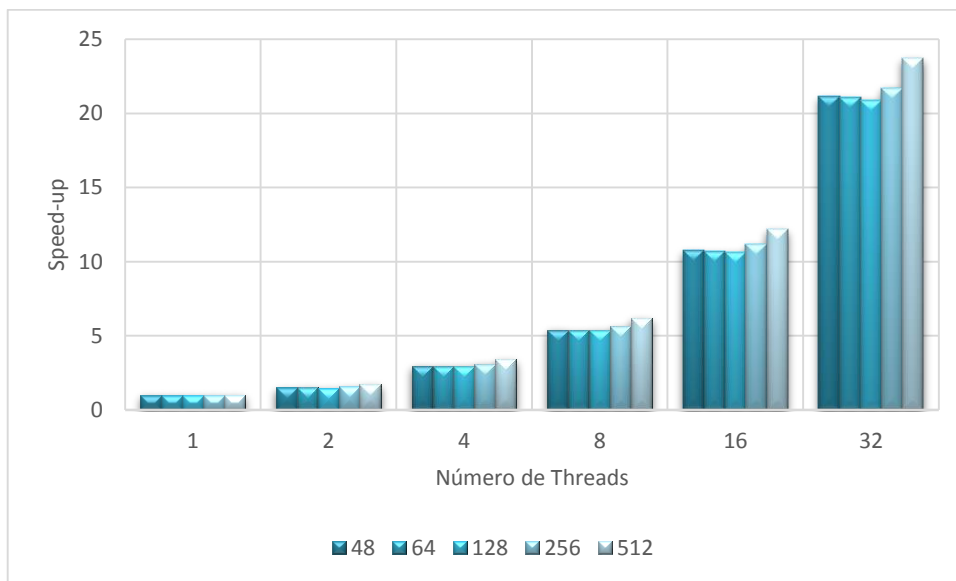


Figura 6.16: Speed-up da versão 2-OMP do novo *solver* em função do número de *threads* para o laço dos blocos percorrido na ordem original (curva azul) e na ordem inversa (curva vermelha)

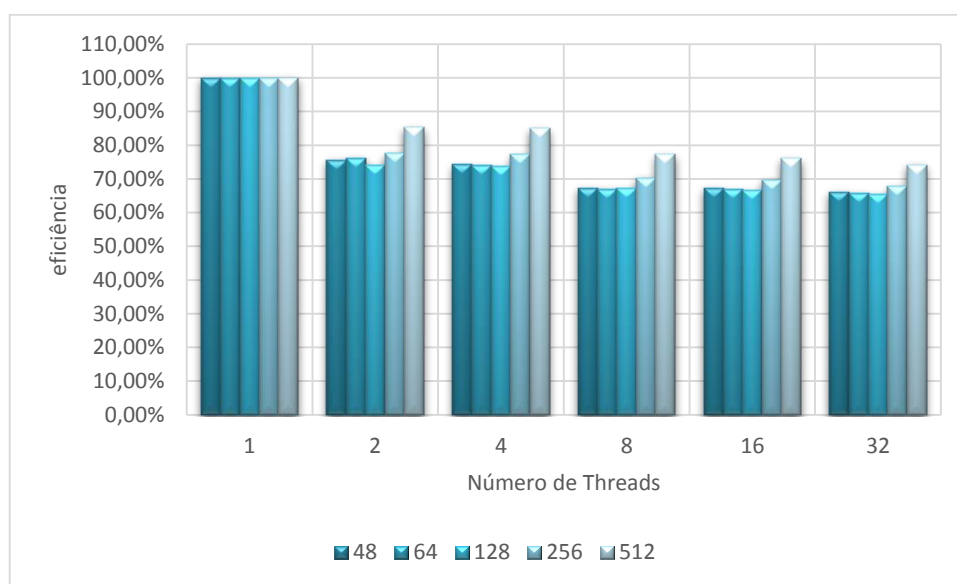


Figura 6.17: Eficiência da versão 2-OMP do novo *solver* em função do número de *threads* para o laço dos blocos percorrido na ordem original (curva azul) e na ordem inversa (curva vermelha)

6.2. Paralelização com OpenACC

A execução do termo de reatividade química com o novo *solver* baseou-se na versão 1-OMP, em que regiões paralelas foram criadas para os laços de pontos do bloco, relativos aos 4 estágios do método de Rosenbrock, conforme exposto no pseudocódigo a seguir. Essas regiões paralelas são executadas na GPGPU. Estes testes foram executados nos nós monoprocesados da Raven, que dispõe de uma GPGPU, para a grade completa de 23.750 blocos e para um tamanho de bloco de 64 pontos. Utilizou-se o compilador OpenACC da Cray para Fortran 90, que implementa a versão corrente do padrão OpenACC (2.0). Constatou-se que o padrão OpenACC ainda está relativamente imaturo, mas aparentemente está sendo continuamente aperfeiçoado. Não foi possível implementar uma versão OpenACC baseada na versão 2-OMP devido à complexa estrutura de dados envolvida que teria que ser copiada para a memória da GPGPU, aumentando muito o tempo de processamento. Além disso, houve restrições relativas aos tipos de variáveis definidas no OpenACC.

PARA CADA bloco **FAÇA** (laço dos blocos)

ENQUANTO (erro > tolerância) **FAÇA**

INICIALIZE (ACC DATA)

INICIALIZE (ACC PARALLEL)

PARA CADA ponto de grade **FAÇA** (laço dos pontos de grade)

Calcule 1º estágio do Método de Rosenbrock

FIM (PARA CADA)

FINALIZE (ACC PARALLEL)

INICIALIZE (ACC PARALLEL)

PARA CADA ponto de grade **FAÇA** (laço dos pontos de grade)

Calcule 2º estágio do Método de Rosenbrock

FIM (PARA CADA)

FINALIZE (ACC PARALLEL)

INICIALIZE (ACC PARALLEL)

PARA CADA ponto de grade **FAÇA** (laço dos pontos de grade)

Calcule 3º estágio do Método de Rosenbrock

FIM (PARA CADA)

FINALIZE (ACC PARALLEL)

INICIALIZE (ACC PARALLEL)

PARA CADA ponto de grade **FAÇA** (laço dos pontos de grade)

Calcule 4º estágio do Método de Rosenbrock

FIM (PARA CADA)

FINALIZE (ACC PARALLEL)

FINALIZE (ACC DATA)

Calcula erro

FINALIZE (ENQUANTO)

FINALIZE (PARA CADA)

Embora semelhante à programação em OpenMP, portar códigos para OpenACC exige o gerenciamento adicional das cópias de dados da CPU para a GPGPU e vice-versa, nem sempre trivial. Conforme já comentado, algumas variáveis do método Rosenbrock não são alteradas ao longo de todos os estágios, como no caso do vetor de projeções p , calculado apenas no primeiro estágio. Assim, tais variáveis devem persistir de uma região paralela que é fechada para uma nova região paralela que é aberta. A diretiva DATA do OpenACC provê essa funcionalidade, preservando essas variáveis como se estivessem numa espécie de área comum. Por outro lado, as diretivas COPYIN e COPYOUT permitem cópias de dados, respectivamente no início da região paralela (da CPU para a GPGPU) e no final desta (da GPGPU para a CPU), enquanto que o UPDATE possibilita cópias da CPU para a GPGPU e vice-versa em qualquer trecho da região paralela. A título de exemplo apresenta-se a seguir um trecho de código em Fortran 90 com algumas diretivas OpenACC.

```

!$ACC DATA COPYIN (DLmat(:, :, :), index_g_maxblock, dt_chem_i) &
!$ACC& CREATE ( v(:), S(:, :), p_fact2(:, :, :), B &
!$ACC& , DLk1(:, :), DLk2(:, :), DLk3(:, :), DLk4(:, :)) &
!$ACC& , DLb1(:, :), DLb2(:, :), DLb3(:, :), DLb4(:, :))
!$ACC UPDATE DEVICE (DLb1(:, :))
!$ACC PARALLEL LOOP PRIVATE(ijk, Ji, Jj, linha, S, v, B)
...
!$ACC END PARALLEL LOOP
!$ACC UPDATE SELF (DLk1(:, :))

```

Constatou-se que a execução com OpenACC ocasionou uma diminuição de cerca de 40% em relação à execução sequencial do termo de reatividade química com o novo *solver*. A figura 6.18 a seguir compara o desempenho da

versão paralelizada com OpenACC com a versão 2-OMP executada com diferentes números de *threads* num nó biprocessado da Raven. Esta redução de 40%, que corresponde a diminuir o tempo de processamento da versão sequencial em aproximadamente 1,6 vezes representa um baixo desempenho paralelo dessa versão OpenACC, o que pode ser explicado pela baixa granularidade, pois são abertas regiões paralelas na dimensão do laço de pontos do bloco.

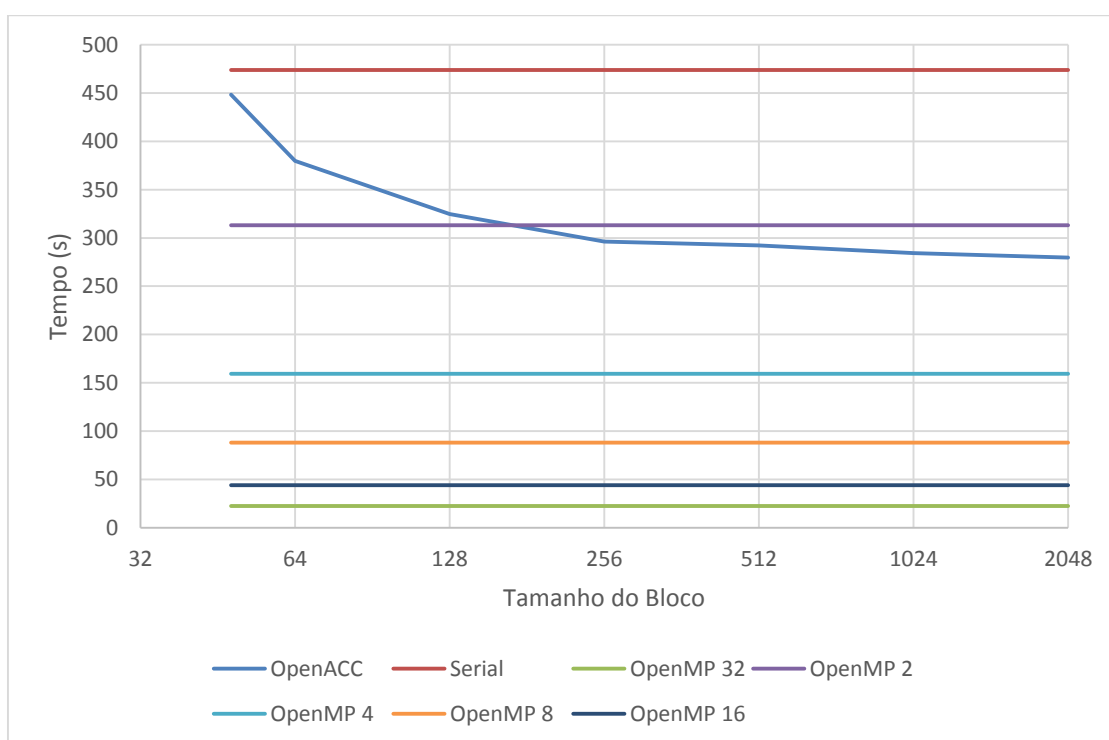


Figura 6.18: Tempos de execução do termo de reatividade química com o novo *solver* para diferentes tamanhos de bloco para a versão sequencial, a versão OpenACC e para a versão 2-OMP executada com 2, 4, 8, 16 e 32 *threads*.

Finalmente, essa comparação é repetida na Figura 6.19, agora incluindo-se os tempos do *solver* original para diferentes tamanhos de bloco.

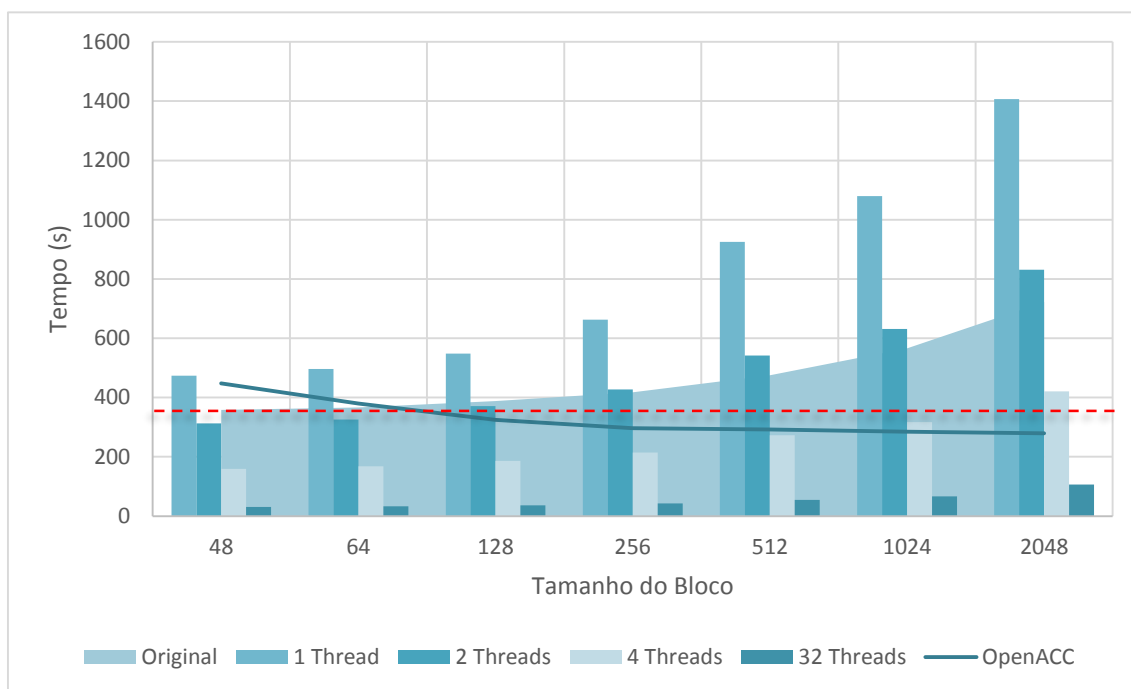


Figura 6.19: Tempos de execução do termo de reatividade química com o novo *solver* para diferentes tamanhos de bloco para a versão sequencial, a versão OpenACC (linha azul cheia) e para a versão 2-OMP executada com 2, 4, 8, 16 e 32 *threads* (colunas azuis) e com o *solver* original (área sombreada azul). A linha vermelha tracejada mostra o melhor tempo do *solver* original, relativo ao tamanho de bloco 48.

6.3. Diferenças numéricas nos resultados

Naturalmente, além da análise de desempenho computacional, é preciso verificar que o novo *solver* proposto para resolução do termo de reatividade química fornece resultados compatíveis (concentrações das espécies químicas) com aqueles do *solver* original. A comparação entre essas concentrações foi feita para as 47 espécies químicas consideradas na Tabela 6.4, onde aparecem as concentrações máximas encontradas bem como suas coordenadas em termos de pontos de grade (i,j,k) para todos os 23.750 blocos de pontos da grade considerados. Além dos valores máximos, aparecem também os valores médios para a toda grade das concentrações de cada espécie. A diferença encontrada corresponde ao 19º elemento, grifada na

tabela 6.4 em vermelho, mas ainda assim envolvendo concentrações baixíssimas, da ordem de 10^{-11} . Todos os outros valores da comparação entre ambos os *solvers*, original e novo, apresentam valores semelhantes. Neste caso, conclui-se que o novo *solver* possui bom desempenho numérico, mesmo sem pré-condicionamento de matrizes ou muitas iterações.

Note-se que o novo *solver* é mais preciso numericamente que o *solver* original.

Tabela 6.4: Concentrações das 47 espécies químicas obtidas pelo novo *solver* e pelo *solver* original. As colunas referem-se às concentrações máximas encontradas, às coordenadas (i,j,k) da grade onde estas ocorreram, e às concentrações médias em toda a grade, sendo cada linha correspondente a uma espécie química (concentrações em parte por bilhão ou em moléculas/cm³ segundo a espécie)

NOVO SOLVER					SOLVER ORIGINAL				
Máximo	i	j	k	Média	Máximo	i	j	k	Média
2,12E+02	24	113	161	4,22E+01	2,12E+02	24	113	161	4,22E+01
2,01E+01	22	10	111	2,75E+00	2,01E+01	22	10	111	2,75E+00
5,77E+01	1	123	72	1,16E-03	5,77E+01	1	123	72	1,16E-03
1,06E+02	5	167	11	1,55E-01	1,06E+02	5	167	11	1,55E-01
2,86E-01	10	114	147	4,30E-03	2,86E-01	10	114	147	4,30E-03
6,99E+00	13	67	115	3,62E-03	6,99E+00	13	67	115	3,62E-03
8,79E-02	2	123	72	9,50E-05	8,79E-02	2	123	72	9,50E-05
7,26E+01	24	49	163	5,63E-01	7,26E+01	24	49	163	5,63E-01
2,30E-01	3	63	140	5,00E-04	2,30E-01	3	63	140	5,00E-04
8,67E+02	4	17	134	2,75E-01	8,67E+02	4	17	134	2,75E-01
2,67E+01	32	40	177	2,24E-01	2,67E+01	32	40	177	2,24E-01
9,32E+02	5	130	164	7,60E+01	9,32E+02	5	130	164	7,60E+01
0,00E+00	1	1	1	0,00E+00	0,00E+00	1	1	1	0,00E+00
0,00E+00	1	1	1	0,00E+00	0,00E+00	1	1	1	0,00E+00
0,00E+00	1	1	1	0,00E+00	0,00E+00	1	1	1	0,00E+00
0,00E+00	1	1	1	0,00E+00	0,00E+00	1	1	1	0,00E+00
7,14E+07	2	124	191	7,04E+08	7,14E+07	2	124	191	7,04E+08
1,06E+01	5	91	159	1,12E-01	1,06E+01	5	91	159	1,12E-01
6,59E-08	16	15	68	1,23E-11	0,00E+00	1	1	1	0,00E+00
1,21E-04	38	26	166	9,71E-07	1,21E-04	38	26	166	9,71E-07
2,14E-02	6	159	104	1,50E-03	2,14E-02	6	159	104	1,50E-03

1,27E+03	5	30	10	8,68E+02	1,27E+03	5	30	10	8,68E+02
9,61E+00	5	130	164	2,41E-01	9,61E+00	5	130	164	2,41E-01
7,24E+01	5	19	37	8,01E-01	7,24E+01	5	19	37	8,01E-01
3,32E+01	5	130	164	5,79E-01	3,32E+01	5	130	164	5,79E-01
2,72E+02	5	135	140	2,41E+00	2,72E+02	5	135	140	2,41E+00
2,25E+01	5	19	37	2,25E-02	2,25E+01	5	19	37	2,25E-02
2,11E+01	5	92	159	7,78E-01	2,11E+01	5	92	159	7,78E-01
1,78E+01	5	130	164	2,73E-01	1,78E+01	5	130	164	2,73E-01
2,53E+01	24	113	161	2,22E+00	2,53E+01	24	113	161	2,22E+00
1,76E+01	11	83	88	3,95E-01	1,76E+01	11	83	88	3,95E-01
2,59E+01	38	139	110	6,69E-01	2,59E+01	38	139	110	6,69E-01
2,37E+01	24	49	163	2,30E-01	2,37E+01	24	49	163	2,30E-01
5,62E+00	11	55	73	1,54E+00	5,62E+00	11	55	73	1,54E+00
1,87E+01	38	59	97	1,19E+00	1,87E+01	38	59	97	1,19E+00
8,58E+00	5	130	164	1,44E-01	8,58E+00	5	130	164	1,44E-01
3,16E+01	5	130	164	1,39E+00	3,16E+01	5	130	164	1,39E+00
9,52E+07	10	51	149	1,13E+06	9,52E+07	10	51	149	1,13E+06
3,07E+07	5	130	164	2,74E+08	3,07E+07	5	130	164	2,74E+08
3,55E+07	5	130	164	2,29E+08	3,55E+07	5	130	164	2,29E+08
6,83E+08	6	159	104	4,48E+06	6,83E+08	6	159	104	4,48E+06
4,01E+03	10	108	152	6,07E+00	4,01E+03	10	108	152	6,07E+00
2,88E+01	2	71	47	9,72E-03	2,88E+01	2	71	47	9,72E-03
2,94E+06	6	90	101	3,80E+03	2,94E+06	6	90	101	3,80E+03
3,29E+08	10	51	149	2,94E+06	3,29E+08	10	51	149	2,94E+06
9,47E+08	2	78	52	2,53E+06	9,47E+08	2	78	52	2,53E+06
1,15E+08	2	124	191	1,31E+06	1,15E+08	2	124	191	1,31E+06

Foram também medidas as diferenças numéricas dos resultados do novo *solver* ao se variar o tamanho de bloco, em relação ao bloco de 64 pontos, sendo apenas encontradas diferenças da ordem de 10^{-11} . E também foram medidas as diferenças entre os resultados da versão sequencial e das versões paralelas do novo *solver*, que resultaram em diferenças da ordem de 10^{-8} .

7 CONCLUSÕES

Este trabalho propôs, implementou e testou um novo *solver* baseado em estimação linear ótima para resolução do termo de reatividade química do modelo ambiental CCATT-BRAMS codificado em Fortran 90 e paralelizado com a biblioteca de comunicação por troca de mensagens MPI. O *solver* original, correntemente empregado, foi implementado de maneira eficiente com uso da biblioteca Sparse 1.3 em linguagem C, mas não é conveniente para paralelização. O novo *solver* proposto é simples, preciso e converge rapidamente, além de ser implementado em pouquíssimas linhas de código, o que o torna apto para paralelização.

A execução do modelo ambiental para uma resolução espacial alta e com a resolução do termo de reatividade química a cada passo de tempo do modelo para atualizar as concentrações das espécies químicas (gases traço instáveis da atmosfera) torna-se inviável devido ao tempo de processamento requerido. Correntemente, esse termo é executado a cada 4 passos de tempo do modelo. O novo *solver* foi paralelizado para execução em nós computacionais de memória compartilhada com o padrão OpenMP de paralelismo de *threads* e também com o padrão OpenACC de paralelismo de *threads* para execução em GPGPU.

Os resultados mostram a viabilidade da paralelização com OpenMP para arquiteturas correntes de supercomputadores como o Tupã do CPTEC/INPE. No caso do OpenACC, os resultados foram mais modestos, mas passíveis de serem melhorados. Assim, uma versão OpenMP do novo *solver* poderia ser implementado operacionalmente, acoplada ao CCATT-BRAMS, possibilitando a atualização das concentrações das espécies químicas com maior frequência e melhorando as previsões desse modelo ambiental.

Como trabalhos futuros, pretende-se explorar essa possibilidade de operacionalização do novo *solver*, gerar uma nova versão OpenACC desse *solver* que tenha maior granularidade, ao paralelizar o laço dos blocos (como

no OpenMP), em vez do laço dos pontos de grade, como na sua versão corrente. Eventualmente, a versão OpenMP poderia ser também aperfeiçoada.

REFERÊNCIAS BIBLIOGRÁFICAS

- ALTERMANN, W.; KAZMIERCZAK, J. Archean microfossils: a reappraisal of early life on Earth. **Research in Microbiology**, v. 154, n. 9, p. 611-617, 2003.
- ANDREAE M. O.; CRUTZEN P. J. Atmospheric aerosols: biogeochemical sources and role in atmospheric chemistry. **Science**, v. 276, n. 5315, p. 1052-1058, 1997.
- ARBEX, M. A.; CANÇADO, J. E. D.; PEREIRA, L. A. A.; BRAGA, A. L. F.; SALDIVA, P. H. N. Queima de biomassa e efeitos sobre a saúde. **Jornal Brasileiro de Pneumologia**, v. 30, n.2, p. 158-175, 2004.
- BREMER, H.; KAR, J.; Drummond, J. R.; NICHITU, F.; ZOU, J.; LIU, J.; GILLE, J. C.; DEETER, M. N.; FRANCIS, G. ZISKIN, D.; WARNER, J.. Carbon monoxide from biomass burning in the tropics and it's impact on the tropospheric ozone. **Journal of Geophysical Research**, v. 109, n. D12304, 2004.
- BRICKUS, L. S. R.; AQUINO NETO, F. R. A qualidade do ar de interiores e a química. **Química Nova**, São Paulo , v. 22, n. 1, 1999.
- COX, P. M.; HARRIS, P. P.; HUNTINGFORD, C.; BETTS, R. A.; COLLINS, M.; JONES, C. D.; JUPP, T. E.; MARENGO, J. A.; NOBRE, C. A. Increasing risk of Amazonian drought due to decreasing aerosol pollution. **Nature**, v. 453, p. 212-215, 2008.
- CRASSIER, V.; SUHRE, K.; TULET, P.; ROSSET, R. Development of a reduced chemical scheme for use in mesoscale meteorological models. **Atmospheric Environment**, v. 34, p. 2633-2644, 2000,.
- CUNHA, M. C. **Métodos numéricos**. 2. Ed. Campinas: UNICAMP. ISBN 978-85-268-0877-5.
- CURTISS C. F., HIRSCHFELDER J. O. Integration of stiff equations. **Proceedings of the National Academy of Sciences of the United States of America – PNAS**, v. 38, p. 235-243, 1952
- ELLER, P. **Development and acceleration of parallel chemical transport models**. 2009, 133 p. Dissertação em ciência da computação e aplicações - Faculty of the Virginia Polytechnic Institute and State University, Blacksburg, EUA, 2009.

FAZENDA, A. L.; PANETTA, J.; ENARI, E. H. Towards production code effective portability among vector machines and microprocessor-based architectures. In: INTERNATIONAL SYMPOSIUM ON COMPUTER ARCHITECTURE AND HIGH PERFORMANCE COMPUTING, 18., 2006, Ouro Preto – MG. **Proceedings...** Ouro Preto, 17–20 October, 2006.

FREITAS S. R.; LONGO, K. M.; SILVA DIAS, M.; CHATFIELD, R.; SILVA DIAS, P.; ARTAXO, P.; ANDREAE M. O.; GRELL, G.; RODRIGUES, L. F.; FAZENDA, A.; PANETTA J. The coupled aerosol and tracer transport model to the brazilian developments on the regional atmospheric modeling system: Part 1, **Atmospheric Chemistry and Physics Discussions**, v. 9, p. 2843-2861, 2009.

FREITAS S. R.; LONGO, K. M.; SILVA DIAS, M.; CHATFIELD, R.; SILVA DIAS, P.; ARTAXO, P.; ANDREAE M. O.; GRELL, G.; RODRIGUES, L. F.; FAZENDA, A.; PANETTA J. The coupled aerosol and tracer transport model to the brazilian developments on the regional atmospheric modeling system. **Atmos. Chem. Phys. Discuss.**, v. 7, p. 8525–8569, 2007.

FREITAS, S. R.; LONGO, K.; SILVA DIAS, M.; SILVA DIAS, P.; CHATFIELD, R.; FAZENDA, A.; RODRIGUES, L. F. The coupled aerosol and tracer transport model to the brazilian developments on the regional atmospheric modeling system: validation using direct and remote sensing observations. In: INTERNATIONAL CONFERENCE ON SOUTHERN HEMISPHERE METEOROLOGY AND OCEANOGRAPHY (ICSHMO), 8., 2006, Foz do Iguaçu. **Proceedings...** São José dos Campos: INPE, 2006. p. 101-107. CD-ROM. ISBN 85-17-00023-4.

GALANTER, M.; LEVY II, H.; CARMICHAEL, G. R. Impacts of biomass burning on tropospheric CO, NO_x, and O₃. **Journal of Geophysical Research**, v.105, n. D5, p. 6633-6653, 2000.

GILAT A., SUBRAMANIAN V. **Métodos numéricos para engenheiros e cientistas**. [S.I.]: Bookman, 2008, ISBN 978-85-7780-205-0.

GOVETT M.; TIERNEY, C.; MIDDLECOFF, J.; HENDERSON, T. Using graphical processing units (GPUs) for next generation weather and climate prediction models. **CAS Presentation**, 2009.

GRAHAN, R. L. Bounds on multiprocessing timing anomalies. **SIAM Journal on Applied Mathematics**, v. 17, n. 2, p. 416-429, 1969.

HAIRER E., WANNER G.. **Solving ordinary differential equations II: stiff and differential algebraic problems** 2. ed. Springer-Verlag, 1996. Springer Series in Comput. Mathematics.

KIRK, D. B.; W. HWU. **Programando para processadores paralelos: uma abordagem prática à programação de GPU**. Elsevier, 2011, ISBN 978-85-3521-4188-4.

KOGGE, P. M.; DYSART, T. M. Using the TOP500 to trace and project technology and architecture trends. In: INTERNATIONAL CONFERENCE FOR HIGH PERFORMANCE COMPUTING, NETWORKING, STORAGE AND ANALYSIS, 2011, Seattle. **Proceedings...** Ney York: ACM, 2011.

LAMBERT, D. J. **Numerical methods for ordinary differential systems**. New York: Wiley, 1992. ISBN 978-0-471-92990-1.

LINFORD, J. C.; VACHHARAJANI, M.; MICHALAKES, J.; SANDU, A. Multi-core acceleration of chemical kinetics for simulation and prediction, In: ACM/IEEE CONFERENCE ON HIGH PERFORMANCE COMPUTING, 2009, Portland, Oregon, USA. **Proceedings...** Portland: ACM/IEEE, 2009

LONGO, K.; FREITAS, S. R.; PIRRE, M.; MARÉCAL, V.; RODRIGUES, L. F.; PANETTA, J.; ALONSO, M. F.; ROSÁRIO, N. E.; MOREIRA, D. S.; GÁCITA, M. S.; ARTETA, J.; FONSECA R.; STOCKLER, R.; KATSURAYAMA, D. M.; FAZENDA, A.; BELA, M. The Chemistry CATT-BRAMS Model (CCATT-BRAMS 4.5): a regional atmospheric model system for integrated air quality and weather forecasting and research; **Geoscientific Model Development Discussions** , v. 6, p. 1389-1405, 2013.

LYNCH, P. The origins of computer weather prediction and climate modeling. **Journal of Computational Physics**, v. 227, p. 3431-3444, 2008.

MOREIRA, D. S.; FREITAS, S. R.; BONATTI, J. P.; MERCADO, L. M.; ROSÁRIO, N. M. É.; LONGO, K. M.; MILLER, J. B.; GLOOR, M.; GATTI, L. V. Coupling between the JULES land-surface scheme and the CCATT-BRAMS atmospheric chemistry model (JULES-CCATT-BRAMS1.0): applications to numerical weather forecasting and the CO2 budget in South America. **Geoscientific Model Development Discussions**, v. 6. n. 1. p. 453-494, 2013.

MORRIS, V.; CLEMENTE-COLÓN, P.; NALLI N. R.; JOSEPH E.; ARMSTRONG R. A.; DETRÉS, Y.; GOLDBERG, M. D.; MINNETT P. J.; LUMPKIN R. Measuring Trans-Atlantic Aerosol Transport From África. In: **EOS Trans. AGU**, v. 87(50), p. 565-571, 2006, doi: 10.1029/2006EO500001.

NVIDIA Corporation. NVIDIA CUDA **compute unified device architecture**: programming guide version 2.0. Santa Clara, 2008.

OLIVIER, J.G.J.; JANSSES-MAENHOUT, G.; MUNTEAN, M.; PETERS, J.H.A.W. **Trends in global CO2 emissions - 2014 report..** 2014. Netherlands: JRC report 93171 / PBL report 1490, 2014, ISBN 978-94-91506-87-1.

OPENACC. **The OpenACC application programming interface.** OpenACC, 2011. Disponível em: <http://www.openacc.org>.

OpenMP. **Architecture review board, openmp application program interface.** OpenMP, 2014. Disponível em: <http://openmp.org>

PINTO R. L. U. F.; RIOS NETO A. An optimal linear estimation approach to solve linear algebraic equations, **Journal of Computational and Applied Mathematics**, North-Holland, v. 33, p.261-268, 1990.

PIELKE, R. A.; COTTON, W. R.; WALKO, R. L.; TREMBACK, C. J.; LYONS, W. A.; GRASSO, L. D.; NICHOLLS, M. E.; MORAN, M. D.; WESLEY, D. A.; LEE, T. J.; COPELAND, J. H. A comprehensive meteorological modeling system—RAMS. **Meteorol. Atmos. Phys**, v. 49, p. 69–91, 1992.

PRESS, W. H.; TEUKOLSKY, S. A.; VETTERLING, W. T.; FLANNERY, B. P.; **Numerical recipes in FORTRAN: the art of scientific computing.** 2. ed. Cambridge University Press New York, 1992. p. 727-745. ISBN 0521437164.

ROSENBROCK, H.H. Some general implicit processes for the numerical solution of differential equations. **Computation Journal** , v. 5, p.329–330, 1963.

SC11; OPENACC Announcement. In: INTERNATIONAL CONFERENCE FOR HIGH PERFORMANCE COMPUTING, NETWORKING, STORAGE AND ANALYSIS, 2011, Seattle. **Proceedings...** Ney York: ACM, 2011.

VERWER, J.; SPEE, E., BLOM, J.; HUNDSDORFER, W. A second order Rosenbrock method applied to photochemical dispersion problems; **SIAM J. Sci. Comput.**, v. 20, p. 1456–1480, 1999, doi:10.1137/S1064827597326651.

WARD, D. E.; SUSOTT, R. A.; KAUFFMAN, J. B.; BABBITT, R. E.; CUMMINGS, D. L.; DIAS, B.; HOLBEN, B. N.; KAUFMAN, Y. J.; RASMUSSEN, R. A.; SETZER, A. W. Smoke and fire characteristics for cerrado and deforestation burns in Brazil: BASE-B experiment. **Journal of Geophysical Research**, v..97, n D13, p. 14,601-14,619, 1992.

ZHANG, H.; LINFORD, J. C.; SANDU, A.; SANDER, R. Chemical mechanism solvers in air quality models. **Atmosphere**, v. 2, p. 510-532, 2013, doi:10.3390/atmos2030510.