

Johnny Moreira Gomes

**Técnicas Computacionais para a Solução Numérica de Modelos Cardíacos
Baseados em Cadeias de Markov**

Dissertação apresentada ao Programa de Pós-graduação em Modelagem Computacional, da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do grau de Mestre em Modelagem Computacional.

Orientador: Prof. D.Sc. Rodrigo Weber dos Santos

Juiz de Fora

2015

Johnny Moreira Gomes,

Técnicas Computacionais para a Solução Numérica de Modelos Cardíacos Baseados em Cadeias de Markov/
Johnny Moreira Gomes. – Juiz de Fora: UFJF/MMC, 2015.

XVI, 103 p.: il.; 29,7cm.

Orientador: Rodrigo Weber dos Santos

Dissertação (mestrado) – UFJF/MMC/Programa de Modelagem Computacional, 2015.

Referências Bibliográficas: p. 98 – 102.

1. Eletrofisiologia Cardíaca. 2. Métodos Numéricos.
3. Cadeias de Markov. 4. Método de Uniformização.
5. Passo de Tempo Adaptativo. I. Weber dos Santos, Rodrigo. II. Universidade Federal de Juiz de Fora, MMC, Programa de Modelagem Computacional.

Johnny Moreira Gomes

**Técnicas Computacionais para a Solução Numérica de Modelos Cardíacos
Baseados em Cadeias de Markov**

Dissertação apresentada ao Programa de Pós-graduação em Modelagem Computacional, da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do grau de Mestre em Modelagem Computacional.

Aprovada em 24 de Fevereiro de 2015.

BANCA EXAMINADORA

Prof. D.Sc. Rodrigo Weber dos Santos - Orientador
Universidade Federal de Juiz de Fora

Prof. D.Sc. Bernardo Martins Rocha
Universidade Federal de Juiz de Fora

Prof^a. D.Sc. Ana Paula Couto da Silva
Universidade Federal de Minas Gerais

Prof. D.Sc Rafael Alves Bonfim de Queiroz
Universidade Federal de Juiz de Fora

*Dedico este trabalho à minha
família, que me forneceu todo o
apoio e suporte que se possa
imaginar.*

AGRADECIMENTOS

Agradeço à minha mãe Marisa; ao meu Pai Rognei; ao meu padrasto Lula; aos meus irmãos Jéssica, Rognei Jr., Ana Cecília e Rafael; às minhas avós Aparecida e Terezinha; à minha tia Marilda, meu tio Marcus e minha prima Laura; e a tantos outros familiares que contribuíram para a minha formação como pessoa e que sempre me propiciaram um ambiente repleto de apoio, de compreensão e de momentos memoráveis. Agradeço especialmente à minha namorada linda Danielle pela paciência, pelo companheirismo e por todo o auxílio que me forneceu durante toda esta jornada. Agradeço aos meus grandes amigos do CTU: Thales, Octávio, Thiago e Luis Paulo por serem sempre os caras com quem posso contar. Agradeço aos meus amigos e colegas da Modelagem Computacional, ao meu orientador Rodrigo e aos meus professores da Engenharia Computacional por terem me incentivado com conhecimentos e experiências incríveis. Agradeço também ao Programa de Pós-Graduação em Modelagem Computacional e às instituições que tornaram possível a conclusão desta etapa em minha carreira acadêmica: FAPEMIG, CAPES e Universidade Federal de Juiz de Fora.

*‘Deixe o futuro dizer a verdade,
e avaliar cada um de acordo com
seus trabalhos e suas conquistas.’*
Nikola Tesla

RESUMO

Este trabalho compara diferentes esquemas numéricos para a solução de modelos modernos para a eletrofisiologia de miócitos cardíacos. Apresentamos o *Método de Uniformização* - amplamente utilizado para a solução de problemas estocásticos em ciência da computação - e mostramos que, quando aplicado na resolução numérica de modelos cardíacos baseados em Cadeias de Markov de Tempo contínuo, aumenta substancialmente a estabilidade numérica em relação a métodos explícitos tradicionalmente utilizados, como o *Método de Rush-Larsen* e o *Método de Euler Explícito*. A formulação em Cadeias de Markov para estruturas subcelulares - como os canais iônicos - permite a descrição detalhada do comportamento elétrico de células cardíacas para importantes aplicações experimentais, como a simulação dos efeitos de drogas e toxinas sobre a atividade elétrica da membrana celular. No entanto, as equações diferenciais associadas às Cadeias de Markov para canais iônicos frequentemente trazem problemas de estabilidade numérica, que limitam fortemente o passo de tempo utilizado por esquemas explícitos. Com a utilização do Método de Uniformização foi possível aumentar significativamente a magnitude dos passos de tempo utilizados em simulações de três modelos da eletrofisiologia cardíaca baseados em Cadeias de Markov. Neste trabalho mostramos como é possível associar o Método de Uniformização a outros esquemas explícitos para a solução numérica de tais modelos, e como tais técnicas melhoram significativamente o desempenho computacional em relação a métodos explícitos tradicionais. Além disso, propomos extensões do método de Rush-Larsen e do método de Uniformização com segunda ordem de precisão para o desenvolvimento de esquemas explícitos de passo de tempo adaptativo, visando reduzir ainda mais o custo computacional em simulações com tolerância numérica estrita.

Palavras-chave: Eletrofisiologia Cardíaca. Métodos Numéricos. Cadeias de Markov. Método de Uniformização. Passo de Tempo Adaptativo.

ABSTRACT

This work compares different numerical schemes for the solution of modern electrophysiology models for cardiac myocytes. We present the *Uniformization Method* - frequently applied to stochastic problems in computer science - which significantly increase the numerical stability when used for the solution of cardiac models based on Continuous Time Markov Chains, with respect to traditional explicit schemes such as *Rush-Larsen Method* and *Foward Euler Method*. The Markov Chains formulation for subcellular structures, e.g. ionic channels, enables an accurate description of the electrical behaviour of cardiac cells for important experimental applications, for instance the simulation of the effects of drugs or toxins on the electrical activity of the cell's membrane. However, the differential equations associated with the Markov Chains for ionic channels frequently cause problems of numerical stability, which severely limits the time step used by explicit schemes. By using the Uniformization Method we could significantly increase the time steps size in simulations of three models of cardiac electrophysiology based on Markov Chains. In this work we show how the Uniformization Method can be used along with other foward numerical schemes for the solution of these models, and how these techniques significantly improve the computational performance with respect to traditional numerical methods. In adition, we propose extensions of the Rush-Larsen method and the Uniformization method with second-order accuracy for developing foward time-adaptive techniques, aiming to reduce the computational cost of simulations with strict numerical tolerances.

Keywords: Cardiac Electrophysiology. Numerical Methods. Markov Chains. Uniformization Method. Adaptive Time Step.

SUMÁRIO

Lista de Acrônimos	16
1 Introdução	17
1.1 Justificativa	17
1.2 Objetivos	20
1.3 Sumário	21
2 Modelagem da Eletrofisiologia Cardíaca	22
2.1 Introdução	22
2.1.1 <i>Estruturas Subcelulares</i>	24
2.2 Potencial de Ação e Excitabilidade	26
2.3 Modelo Elétrico para a Membrana Celular	28
2.4 Modelos para a Corrente Iônica	30
2.5 Modelos para os Canais Iônicos	31
2.5.1 <i>Modelo de Dois Estados</i>	31
2.5.2 <i>Modelo de Subunidades</i>	33
2.5.3 <i>Cadeias de Markov</i>	35
2.6 O Modelo de Hodgkin e Huxley (1952)	37
2.7 O Modelo de Bondarenko <i>et al.</i> (2004)	41
2.8 Modelo Generalizado para a Membrana Celular	43
3 Métodos Numéricos	46
3.1 Método de Euler Explícito	46
3.2 Método Runge-Kutta de 2ª Ordem	47
3.3 Método de Euler Adaptativo	47
3.4 Método de Rush-Larsen	48
3.5 Método de Uniformização	49
3.6 Método de Sundnes <i>et al.</i>	50
3.7 Métodos de Rush-Larsen de 2ª Ordem	52

4	Métodos Propostos e Implementações	54
4.1	Métodos de Precisão Numérica Estendida	54
4.1.1	<i>Método de Rush-Larsen Explícito de 2ª Ordem (RL2)</i>	54
4.1.2	<i>Método SAST2 Simplificado</i>	55
4.1.3	<i>Método de Uniformização de 2ª Ordem (UNI2)</i>	56
4.2	Métodos Híbridos	57
4.2.1	<i>Método Euler + UNI</i>	58
4.2.2	<i>Método RL + UNI</i>	58
4.2.3	<i>Método SAST1 + UNI</i>	58
4.3	Métodos com Discretização Temporal Particionada (DTP)	58
4.4	Métodos Adaptativos	60
4.4.1	<i>Método de Rush-Larsen Adaptativo</i>	60
4.4.2	<i>Métodos Híbridos de Passo Adaptativo</i>	61
4.5	Otimizações para o Método de Uniformização	63
4.5.1	<i>Truncamento da Série</i>	64
4.5.2	<i>Matrizes Esparsas</i>	65
4.6	Implementação dos Métodos Adaptativos	68
4.7	Experimentos Computacionais	69
4.7.1	<i>Modelos Utilizados</i>	69
4.7.2	<i>Protocolos de Estímulo Elétrico</i>	70
4.7.3	<i>Ferramentas Computacionais</i>	71
4.7.4	<i>Ambiente Computacional</i>	71
4.8	Métricas	71
4.8.1	<i>Erros Numéricos e Tolerâncias</i>	72
4.8.2	<i>Tempos de Computação</i>	72
4.8.3	<i>Utilização de Memória</i>	72
4.8.4	<i>Passos de Tempo</i>	73
4.8.5	<i>Análise de Rigidez</i>	74
5	Resultados e Discussão	76
5.1	Resultados de Convergência	76
5.2	Utilização de Memória	77
5.3	Tempos de Computação	79

5.3.1	<i>Métodos com Discretização Temporal Uniforme</i>	79
5.3.2	<i>Métodos com Discretização Temporal Particionada</i>	82
5.3.3	<i>Métodos de Passo Adaptativo</i>	84
5.4	Discussão	88
5.4.1	<i>Métodos Híbridos</i>	88
5.4.2	<i>Métodos Adaptativos</i>	93
5.4.3	<i>Utilização de Memória</i>	94
6	Conclusões e Trabalhos Futuros	96
APÊNDICES		102

LISTA DE ILUSTRAÇÕES

2.1	Estrutura do coração de um mamífero (adaptado de Guyton (2005)).	23
2.2	Origem e propagação do estímulo elétrico no coração de um mamífero (adaptado de Constanzo (2007)).	24
2.3	Estrutura básica da membrana celular e fluxo de íons através de um canal iônico (extraído de Campos (2008)).	25
2.4	Dinâmica de abertura e fechamento de canais iônicos sensíveis ao potencial transmembrânico (extraído de Aires (2007)).	25
2.5	Potencial de ação medido sobre a membrana sarcoplasmática de um miócito cardíaco (adaptado de (Sachse, 2004)).	27
2.6	Visualização das quatro fases de um Potencial de Ação (adaptado de Costa (2011)).	28
2.7	Modelo elétrico para a membrana celular.	29
2.8	Modelo de dois estados.	32
2.9	Espaço de estados para o modelo de duas subunidades distintas.	33
2.10	Espaço de estados para o modelo de duas subunidades idênticas.	34
2.11	Cadeia de Markov com os estados aberto (O), fechado (C) e inativo (I).	35
2.12	Cadeia de Markov para os Canais Iônicos da <i>Corrente Rápida de Sódio</i> (Bondarenko <i>et al.</i> , 2004).	37
2.13	Esquematisação da membrana celular e dos canais iônicos considerados no modelo de Hodgkin e Huxley (1952).	38
2.14	Potencial de ação do modelo de Hodgkin e Huxley (1952) (adaptado de Keener e Sneyd (1998)).	39
2.15	Variação das condutividades g_{na} e g_k do modelo Hodgkin-Huxley durante um potencial de ação (adaptado de Keener e Sneyd (1998)).	40
2.16	Ilustração esquemática dos fluxos iônicos e compartimentos celulares considerados pelo modelo de Bondarenko <i>et al.</i> (Imagem extraída de Bondarenko <i>et al.</i> (2004)).	43
4.1	Cadeia de Markov para a corrente de Ca^{2+} tipo L do modelo de Bondarenko <i>et al.</i> (Imagem extraída de Bondarenko <i>et al.</i> (2004)).	65

4.2	Cadeia de Markov para a corrente de I_{Kr} do modelo de Bondarenko <i>et al.</i> (Imagem extraída de Bondarenko <i>et al.</i> (2004)).	67
5.1	Gráfico h <i>versus</i> tempo para o método SAST1 + UNI / ADP ($\tau = 0.02$), aplicado ao modelo de Bondarenko <i>et al.</i>	86
5.2	Gráfico h <i>versus</i> tempo para o método SAST1 + UNI / ADP ($\tau = 0.001$), aplicado ao modelo de Winslow <i>et al.</i>	87
5.3	Gráfico h <i>versus</i> tempo para o método SAST1 + UNI / ADP ($\tau = 0.006$), aplicado ao modelo de Iyer <i>et al.</i>	88
5.4	Variável rígida P_{O_1} <i>versus</i> tempo durante um PA do modelo de Bondarenko <i>et al.</i>	92
5.5	Variável rígida $[Ca^{2+}]_{SS}$ <i>versus</i> tempo durante um PA do modelo de Winslow <i>et al.</i>	93

LISTA DE TABELAS

2.1	Diferenças entre as concentrações dos íons Na^+ e K^+ nos meios intracelular e extracelular de um axônio gigante de lula (Hodgkin e Huxley, 1952).	26
4.1	Esquemas numéricos com discretização particionada ($h_1 < h_2$)	59
5.1	Taxas de convergência para o método RL2 obtidas com o modelo proposto por Tusscher e Panfilov (2006).	76
5.2	Taxas de convergência para o método UNI2 obtidas com o modelo proposto por Bondarenko <i>et al.</i> (2004).	77
5.3	Picos de memória alocada no <i>Heap</i> durante simulações de modelos da eletrofisiologia cardíaca.	78
5.4	<i>Overheads</i> de utilização de memória por parte do método de uniformização. .	79
5.5	Resultados de Simulação ($t = 0s$ até $t = 1s$) - Bondarenko <i>et al.</i>	79
5.6	Resultados de Simulação ($t = 0s$ até $t = 1s$) - Winslow <i>et al.</i>	80
5.7	Resultados de Simulação ($t = 0s$ até $t = 1s$) - Iyer <i>et al.</i>	80
5.8	Resultados de Simulação ($t = 0s$ até $t = 10s$) - Tusscher-Panfilov.	80
5.9	Ganhos de desempenho do método SAST1 + UNI em relação aos métodos de Euler e de Rush-Larsen.	80
5.10	Resultados de simulação com discretização particionada ($t = 0s$ até $t = 1s$) - Bondarenko <i>et al.</i>	82
5.11	Resultados de simulação com discretização particionada ($t = 0s$ até $t = 1s$) - Winslow <i>et al.</i>	82
5.12	Resultados de simulação com discretização particionada ($t = 0s$ até $t = 1s$) - Iyer <i>et al.</i>	82
5.13	Resultados de simulação com discretização particionada ($t = 0s$ até $t = 10s$) - Tusscher-Panfilov.	82
5.14	Ganhos de desempenho utilizando discretização particionada.	83
5.15	Resultados de simulações para métodos adaptativos - Tusscher e Panfilov (2006). .	84
5.16	Resultados de Simulação para Métodos Adaptativos ($t = 0s$ até $t = 1s$) - Bondarenko <i>et al.</i>	85

5.17 Resultados de Simulação para Métodos Adaptativos ($t = 0s$ até $t = 1s$) - Winslow <i>et al.</i>	86
5.18 Resultados de Simulação para Métodos Adaptativos ($t = 0s$ até $t = 1s$) - Iyer <i>et al.</i>	87
5.19 Ganhos de desempenho em relação ao método de Euler por parte das variações do método SAST1 + UNI.	94
A.1 Lista de métodos numéricos existentes na literatura.	103
A.2 Lista de esquemas numéricos propostos.	103

Lista de Acrônimos

CM	Cadeia de Markov
DTP	Discretização Temporal Particionada
EDO	Equação Diferencial Ordinária
HH	Referência a variáveis ou equações do tipo "Hodgkin-Huxley"
MK	Referência a variáveis ou equações referentes a Cadeias de Markov
NL	Referência a variáveis ou equações do tipo "totalmente não-linear"
OMS	Organização Mundial da Saúde
PA	Potencial de Ação
RAM	<i>Random Access Memory</i>

1 Introdução

A modelagem da fisiologia cardíaca é hoje uma área de pesquisa extremamente ativa, envolvendo múltiplas áreas do conhecimento científico. Depois da publicação de Hodgkin e Huxley (1952) os modelos celulares têm evoluído constantemente. O desenvolvimento recente da genética tem auxiliado a formulação destes modelos através da identificação das proteínas que formam os diversos canais iônicos que permeiam a membrana celular (Schram *et al.*, 2002).

Os canais iônicos possuem um papel essencial na geração do potencial de ação (PA). Pesquisas recentes têm permitido a compreensão minuciosa acerca dos canais iônicos, das suas funções e das conformações das proteínas que compõem sua estrutura. Muitas doenças alteram as propriedades eletrofisiológicas das correntes iônicas associadas a canais iônicos, por exemplo, de células do coração. Essas alterações podem ser descritas matematicamente por cadeias de Markov.

Modelos da eletrofisiologia cardíaca são compostos por sistemas de equações diferenciais de difícil resolução numérica (Rush e Larsen, 1978). As equações derivadas dos modelos de Markov aumentam a complexidade do sistema como um todo, demandando a utilização de pequenos passos de tempo em métodos numéricos condicionalmente estáveis - por exemplo, o método de Euler -, os quais são largamente utilizados para a simulação de tecidos cardíacos, devido à sua intrínseca utilização de memória reduzida. Portanto, é de grande importância a busca por técnicas numéricas que venham a aumentar o desempenho de métodos numéricos para os modelos baseados em cadeias de Markov.

1.1 Justificativa

As doenças cardíacas são responsáveis por um terço do total de mortes registradas em todo o mundo (OMS). Acredita-se que mais de 300 mil pessoas morrem anualmente no Brasil vítimas de patologias relacionadas principalmente à atividade elétrica do coração. Muitos esforços têm sido feitos para compreender as causas das doenças cardíacas, de

modo a proporcionar, para a comunidade científica, sólidas bases para o desenvolvimento de novas formas de tratá-las e de detectá-las.

Para o cumprimento de tais objetivos, faz-se necessária uma profunda compreensão acerca do funcionamento do coração. Entretanto, a sua grande complexidade, desde o nível molecular até o nível dos tecidos, torna a tarefa extremamente difícil. Todavia, alguns de seus aspectos podem ser melhor compreendidos através da modelagem computacional (Martin, 1968), que permite aos pesquisadores realizarem um vasto número de experimentos em um curto período de tempo, sem que sejam necessários estudos *in vivo* ou *in vitro*. Espera-se que, a longo prazo, possa-se vislumbrar um cenário em que todo o sistema circulatório humano poderá ser simulado computacionalmente, permitindo aos cientistas desenvolverem e testarem medicamentos contra as mais diversas doenças que atacam o coração através de experimentos virtuais - *in silico* -, ou seja, sem riscos e em tempo menor do que em situações de experimentação com cobaias.

A utilização de modelos computacionais para a descrição matemática de complexos sistemas naturais foi incluída pela Sociedade Brasileira de Computação (SBC) na lista dos Grandes Desafios da Computação para o período 2006/2016 (Carvalho *et al.*, 2006). O Comitê de Pesquisa em Computação do Reino Unido e a Sociedade Britânica de Computação estabeleceram em 2004 (Hoare e Milner, 2004) a personificação computacional de animais, plantas e organismos unicelulares como um dos Grandes Desafios da Computação para os próximos anos. As sociedades de computação inglesa e brasileira estão cientes de que a complexidade envolvida na modelagem dos sistemas naturais irá impor a necessidade de melhorias nas atuais técnicas de solução numérica e de um grande suporte computacional para sua simulação.

Em virtude das grandes demandas computacionais inerentes à simulação dos complexos sistemas naturais, faz-se necessária a utilização de técnicas eficientes para a solução numérica de sistemas de equações diferenciais.

Sobre a simulação da eletrofisiologia do órgão cardíaco, o elevado número de células presentes nos tecidos do coração implica na simulação de malhas computacionais com elevada quantidade de nós, sendo que cada um dos mesmos representa um diferente

sistema de equações diferenciais ordinárias a ser resolvido numericamente. Deste modo, é de extrema importância a utilização de métodos numéricos com baixa demanda de memória RAM no processo de solução de cada um desses sistemas. Por isso, métodos condicionalmente estáveis, como o método de Euler, o método de Runge-Kutta e o método de Rush-Larsen (Rush e Larsen, 1978) são amplamente utilizados em simulação eletrofisiológica de tecidos cardíacos.

Tais técnicas, por sua natureza, limitam o tamanho dos passos de tempo tomados no processo de simulação, devido às suas condições de estabilidade. Tal limitação é acentuada com a presença de equações do tipo *stiff* (rígidas), que são equações diferenciais ordinárias em que as complicações de instabilidade por tamanho do passo de tempo são consideravelmente elevadas. O método de Rush-Larsen (Rush e Larsen, 1978) busca atenuar o problema de instabilidade sobre determinados tipos de equações presentes em modelos eletrofisiológicos. Quando tais equações constituem a parte mais rígida do sistema como um todo, como no caso do modelo proposto por Tusscher e Panfilov (2006), o método permite a utilização de passos de tempo consideravelmente maiores, diminuindo o tempo de computação de forma acentuada (MacLachlan *et al.*, 2005).

Determinados modelos celulares de eletrofisiologia cardíaca, como o modelo para miócitos de camundongos proposto por Bondarenko *et al.* (2004) e o modelo para células cardíacas humanas proposto por Iyer *et al.* (2004), utilizam modelos de Markov (Jensen, 1953) para descrever o comportamento de canais iônicos. Tal modelagem permite uma descrição mais completa do funcionamento de tais estruturas, fornecendo a possibilidade da simulação dos efeitos de drogas e de toxinas sobre tais canais e, conseqüentemente, sobre a atividade elétrica do coração.

A presença de equações derivadas de cadeias de Markov em modelos celulares de eletrofisiologia cardíaca frequentemente faz com que o tamanho do passo de tempo utilizado em simulação seja limitado pelas condições de estabilidade de tais equações, pois as mesmas geralmente figuram-se como a parte mais rígida do modelo (MacLachlan *et al.*, 2005), que é o caso dos modelos descritos em Bondarenko *et al.* (2004) e Iyer *et al.* (2004). Conseqüentemente, o método de Rush-Larsen, quando aplicado a tais modelos,

passa a não oferecer ganho em desempenho computacional.

A técnica de uniformização (Melamed e Yadin, 1984) é um método de estabilidade superior a métodos explícitos tradicionais para a solução numérica de sistemas de equações diferenciais ordinárias derivadas de cadeias de Markov. Segundo recentes experimentos (Gomes *et al.*, 2015), sua utilização, em conjunto com o método de Euler, em modelos de eletrofisiologia cardíaca, permite considerável aumento no tamanho dos passos de tempo e, por consequência, a diminuição nos tempos de computação.

Abordagens como o método de Euler e o método de Runge-Kutta de 2^a ordem, utilizando-se passo de tempo adaptativo, também resultaram em ganhos de desempenho na simulação de vários modelos desta natureza (Campos *et al.*, 2013). Portanto, a pesquisa sobre a utilização do método de uniformização em conjunto com essas demais técnicas pode vir a trazer resultados bastante relevantes no que se refere ao aumento de desempenho no processo de simulação numérica de tecidos cardíacos.

1.2 Objetivos

Conforme descrito anteriormente, a modelagem matemática da eletrofisiologia de células cardíacas tem por objetivo principal fornecer a pesquisadores da área ferramentas computacionais capazes de auxiliar na busca por novas técnicas de tratamento e de diagnóstico de doenças cardíacas. A pesquisa por técnicas eficientes para a solução numérica de tais modelos é fundamental para que tais ferramentas possam ser geradas e efetivamente utilizadas, uma vez que a complexidade envolvida é considerável, especialmente para o caso dos modelos baseados em Cadeias de Markov.

Neste trabalho propomos a utilização do método de uniformização em conjunto com métodos amplamente utilizados na comunidade de modelagem cardíaca, visando melhorar o desempenho de simulações envolvendo modelos recentes baseados em Cadeias de Markov. Além disso, propomos extensões explícitas com precisão numérica de segunda ordem para os métodos de Rush-Larsen e de uniformização, com a finalidade de desenvolver esquemas de passo adaptativo para os métodos híbridos anteriormente citados, buscando obter ganhos de desempenho computacional enquanto os erros numéricos são

mantidos sob controle.

1.3 Sumário

Nesta seção, descreveremos brevemente a organização do conteúdo deste trabalho através dos capítulos que compõem o presente documento.

Neste primeiro capítulo apresentamos uma introdução, onde os aspectos gerais e objetivos do trabalho foram apresentados, bem como a justificativa para a pesquisa desenvolvida na área do conhecimento em questão.

No capítulo 2, iremos apresentar o conteúdo teórico básico para a formulação de modelos eletrofisiológicos para células cardíacas, introduzindo conceitos fundamentais para a formulação geral das equações envolvidas.

No capítulo 3 serão apresentados os métodos numéricos da literatura que foram utilizados neste trabalho para a solução dos sistemas de EDOs não-lineares derivados de modelos da eletrofisiologia cardíaca.

O capítulo 4 descreverá os novos métodos desenvolvidos neste trabalho e tópicos selecionados com relação à implementação de tais métodos, bem como o ambiente computacional, os modelos utilizados e as métricas empregadas no processo de experimentação computacional.

O capítulo 5 apresenta os resultados obtidos na simulação dos modelos citados utilizando os métodos descritos nos capítulos 3 e 4. Além disso, é apresentada uma análise sobre o comportamento dos métodos em estudo quando aplicados aos diferentes modelos, com base nos resultados obtidos.

No capítulo 6 apresentamos uma conclusão geral acerca do trabalho realizado, limitações dos métodos propostos e possíveis trabalhos futuros.

2 Modelagem da Eletrofisiologia Cardíaca

A modelagem da eletrofisiologia cardíaca parte, fundamentalmente, de formulações matemáticas para fenômenos evidenciados no nível celular. Neste capítulo apresentaremos conceitos básicos em relação à eletrofisiologia de células excitáveis e estabeleceremos os passos fundamentais para a formulação matemática de modelos que descrevem o comportamento elétrico das estruturas envolvidas.

2.1 Introdução

O coração é um órgão muscular, cuja principal função é fornecer oxigênio e nutrientes para as células por meio do bombeamento de sangue para todo o corpo do indivíduo. Tal processo se dá pelo mecanismo periódico de contração (sístole) e de relaxamento (diástole) do músculo cardíaco (Constanzo, 2007).

A Figura 2.1 ilustra a estrutura básica do coração de um mamífero, composta por dois átrios e dois ventrículos. Os átrios são as estruturas que recebem o sangue, enquanto os ventrículos são as estruturas que expulsam o sangue para o restante do organismo.

A visão geral do ciclo cardíaco é relativamente simples: o átrio direito recebe o sangue desoxigenado do corpo, leva-o para o ventrículo direito, de onde será levado para os pulmões com a finalidade de receber oxigênio. Após esse processo, o sangue retorna ao coração pelo átrio esquerdo e é levado ao ventrículo esquerdo para, finalmente, ser bombeado para o restante do corpo.

Os tecidos musculares que compõem o músculo cardíaco são compostos de células musculares denominadas *miócitos*. Tais células, ao serem submetidas a um pequeno estímulo elétrico, sofrem um processo de variação temporal do seu potencial transmembrânico, que é a diferença de potencial elétrico entre os meios intracelular e extracelular. Esse percurso do potencial transmembrânico é denominado potencial de

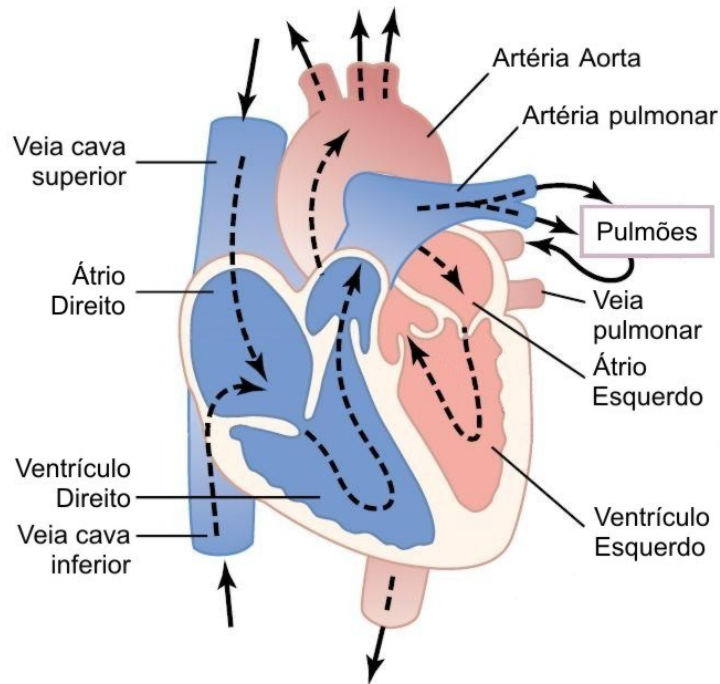


Figura 2.1: Estrutura do coração de um mamífero (adaptado de Guyton (2005)).

ação (PA), o qual é dividido nas fases de despolarização e repolarização. Na situação de repouso, o potencial transmembrânico é negativo. Com a aplicação de um estímulo elétrico, tal potencial varia rapidamente para um valor positivo, o que caracteriza a fase de despolarização. Posteriormente, o potencial de repouso será retomado através de uma diminuição lenta e gradativa do potencial transmembrânico, o que consiste na fase de repolarização.

Os miócitos cardíacos são considerados células excitáveis e contráteis, sendo que o processo de contração é sempre precedido pelo advento do potencial de ação. Para garantir o sincronismo entre miócitos no processo de contração, o PA é propagado entre todos os miócitos cardíacos, tendo sua origem no chamado *nodo sinoatrial* (veja a Figura 2.2). O sinal elétrico percorre o órgão cardíaco através de fibras de alta condutividade, sendo que o mesmo atinge os ventrículos em tempo posterior à sua chegada aos átrios, de modo que os ventrículos sofram contração após o sangue ter sido devidamente bombeado pelos átrios (Jenkins *et al.*, 2002).

Os processos biofísicos envolvidos na geração e na propagação do PA celular são de elevada complexidade e de natureza não linear, sendo sua compreensão fundamental para o

entendimento e para a exploração dos complexos fenômenos da fisiologia celular (Campos, 2008). Portanto, diversos modelos computacionais para a descrição do PA em células cardíacas vêm sendo formulados desde a segunda metade do século XIX, com o objetivo de auxiliar no processo de compreensão do funcionamento do coração e na busca por novas técnicas de tratamento e de diagnóstico de doenças cardíacas.

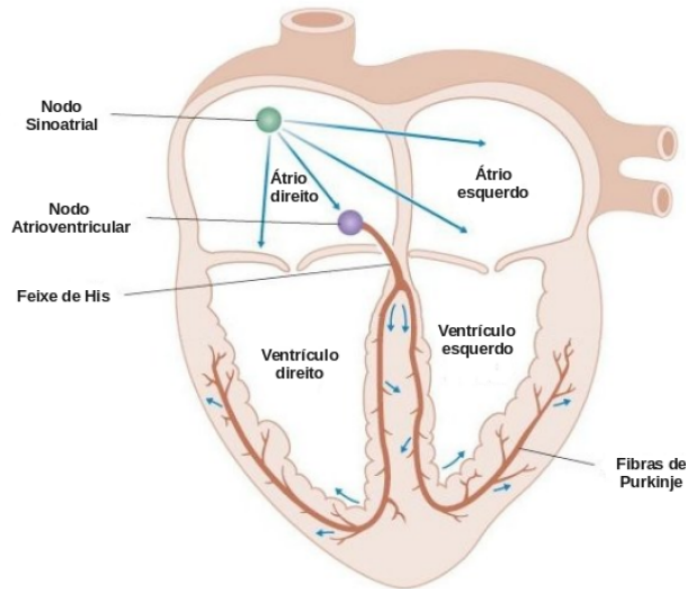


Figura 2.2: Origem e propagação do estímulo elétrico no coração de um mamífero (adaptado de Constanzo (2007)).

2.1.1 Estruturas Subcelulares

Conforme mostrado em Alberts *et al.* (2003), a célula tem seu interior delimitado por uma membrana que controla o fluxo das substâncias que entram e saem do citoplasma. A membrana é constituída de uma bicamada fosfolipídica contínua que mantém uma relação ambivalente com a água (Figura 2.3). Ambos os meios intracelular e extracelular são soluções aquosas compostas por sais dissolvidos, principalmente NaCl e KCl, os quais se dissociam em íons Na^+ , K^+ e Cl^- . A bicamada fosfolipídica age como uma barreira ao livre fluxo desses íons, mantendo assim uma diferença de concentração e de potencial elétrico entre os meios.

O transporte de íons e de moléculas entre os meios intracelular e extracelular pode ocorrer por meio de processos ativos (com gasto energético) e passivos (sem gasto

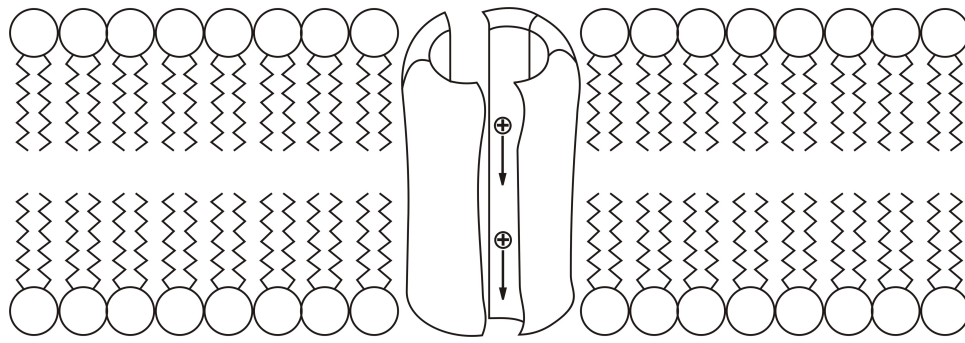


Figura 2.3: Estrutura básica da membrana celular e fluxo de íons através de um canal iônico (extraído de Campos (2008)).

energético) (Alberts *et al.*, 2003). O transporte passivo de íons ocorre pelo processo de difusão através de estruturas altamente seletivas denominadas canais iônicos. Os canais iônicos são altamente sensíveis a determinadas variáveis do sistema celular, sendo que a maior parte tem seu comportamento regulado pelo potencial transmembrânico. A Figura 2.4 ilustra determinado canal iônico tendo seu estado de condutividade alterado pelo potencial transmembrânico.

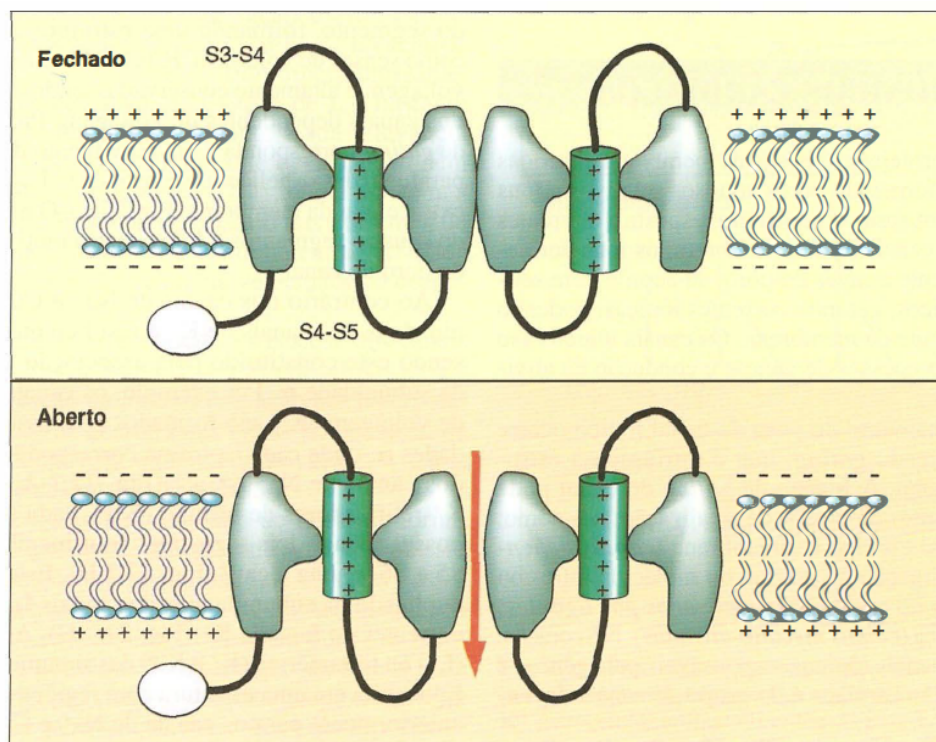


Figura 2.4: Dinâmica de abertura e fechamento de canais iônicos sensíveis ao potencial transmembrânico (extraído de Aires (2007)).

2.2 Potencial de Ação e Excitabilidade

Conforme visto na seção anterior, a membrana plasmática admite a existência de fluxos iônicos através de estruturas proteicas presentes em toda a sua extensão. A regulação do potencial transmembrânico (a diferença de potencial entre os meios intracelular e extracelular) pelos canais iônicos é uma das funções mais importantes da célula (Alberts *et al.*, 2003). Vários tipos de células, como neurônios e células musculares, usam este potencial como forma de comunicação intercelular. Assim, o funcionamento do sistema nervoso e da contração muscular, por exemplo, dependem da geração e da propagação de sinais elétricos, isto é, do potencial de ação.

Para que possamos estudar os sinais elétricos nas células, precisamos classificá-las em dois grupos distintos: células excitáveis e células não-excitáveis. Muitas células mantêm um potencial de equilíbrio estável. Para algumas delas, se correntes elétricas são aplicadas em um período curto de tempo, o potencial retorna diretamente para o equilíbrio depois que a corrente é removida. Tais células são chamadas de não-excitáveis.

Por outro lado, as chamadas células excitáveis, ao receberem uma corrente elétrica de intensidade suficiente através de sua membrana, percorrem um longo caminho de variação do seu potencial transmembrânico, ou seja, o potencial de ação (PA). Tais células, quando na situação de repouso, possuem um potencial transmembrânico negativo, em torno de $-80mV$ para células nervosas. Essa diferença de potencial é mantida devido a uma grande diferença de concentração iônica entre os meios intracelular e extracelular. A Tabela 2.1 mostra as concentrações de repouso dos íons de sódio e de potássio em um axônio gigante de lula.

Tabela 2.1: Diferenças entre as concentrações dos íons Na^+ e K^+ nos meios intracelular e extracelular de um axônio gigante de lula (Hodgkin e Huxley, 1952).

Íon	Conc. meio intracelular (mM)	Conc. meio extracelular (mM)
K^+	397	20
Na^+	50	497

A Figura 2.5 ilustra esquematicamente um potencial de ação medido na membrana de um miócito cardíaco. Um miócito cardíaco em estado de repouso é submetido

a uma corrente elétrica de estímulo. Imediatamente após tal evento, o potencial transmembrânico sofre uma rápida elevação, ou seja, sofre o processo de despolarização. Após uma rápida queda no potencial, uma fase relativamente longa, denominada fase platô, é iniciada. Finalmente, a repolarização leva a célula novamente ao potencial de repouso.

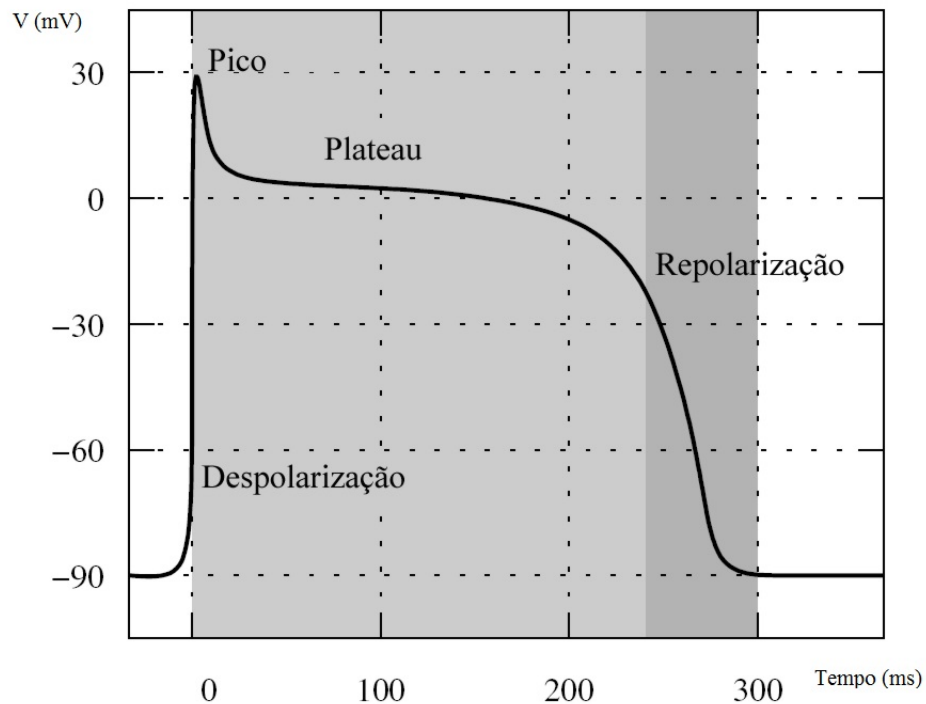


Figura 2.5: Potencial de ação medido sobre a membrana sarcoplasmática de um miócito cardíaco (adaptado de (Sachse, 2004)).

O PA é dividido em quatro fases distintas (Aires, 2007), as quais são mostradas na Figura 2.6. A fase de repouso é aquela em que a membrana diz-se polarizada, com o potencial transmembrânico constante em um valor menor que zero. A fase de despolarização, conforme descrito anteriormente, ocorre imediatamente após à aplicação de uma corrente elétrica que eleva o potencial para um valor acima do chamado *potencial limiar*. Nessa fase, a membrana subitamente torna-se fortemente permeável aos íons de sódio, os quais dirigem-se para o interior da célula em um grande fluxo que eleva o potencial transmembrânico a um valor positivo. Na fase de repolarização, ocorre a inativação dos canais de sódio e um considerável aumento na condutividade dos canais de potássio, gerando um fluxo de potássio para o meio extracelular, o que leva ao lento

restabelecimento do potencial de repouso. A fase de hiperpolarização ocorre quando a fase de repolarização leva o potencial a um valor inferior ao potencial de repouso.

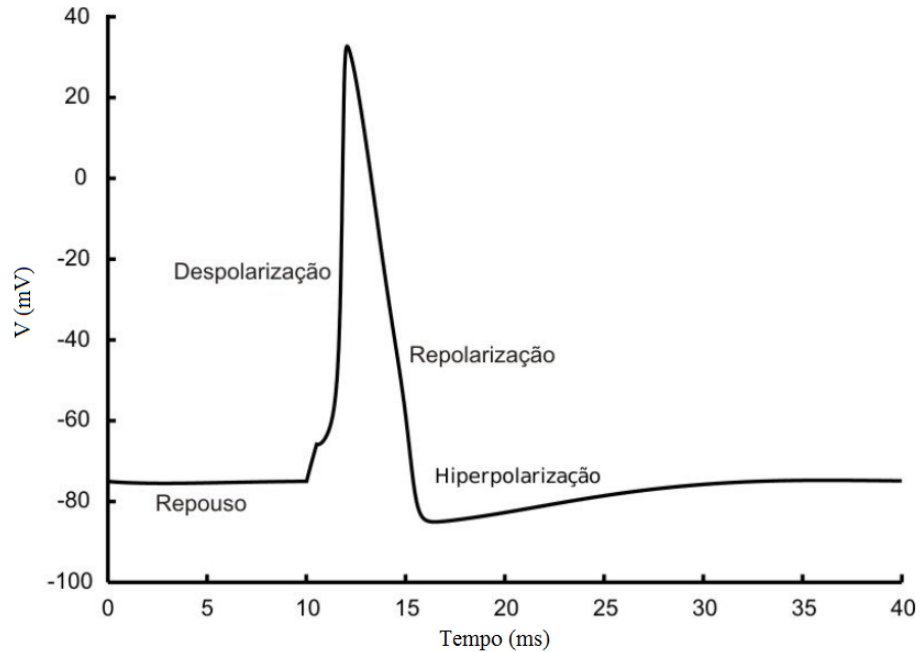


Figura 2.6: Visualização das quatro fases de um Potencial de Ação (adaptado de Costa (2011)).

2.3 Modelo Elétrico para a Membrana Celular

A membrana celular, sendo composta por uma camada separadora e isolante elétrica, além de estruturas pelas quais ocorrem fluxos de cargas (ou correntes elétricas), pode ser modelada por um circuito elétrico composto por um capacitor e várias resistências em paralelo (Figura 2.7). Conforme veremos adiante, as resistências associadas aos canais iônicos são funções não lineares da diferença de potencial entre os meios intracelular e extracelular (Hodgkin e Huxley, 1952). Por definição, tal diferença de potencial é chamada potencial transmembrânico, o qual é dado por $V = V_{Intracelular} - V_{Extracelular}$.

A partir do modelo elétrico apresentado na Figura 2.7, podemos extrair uma equação diferencial para o potencial transmembrânico, a qual é obtida da lei de Kirchhoff para correntes elétricas, ou seja, o somatório de correntes que saem de um nó pertencente a um circuito elétrico é nula.

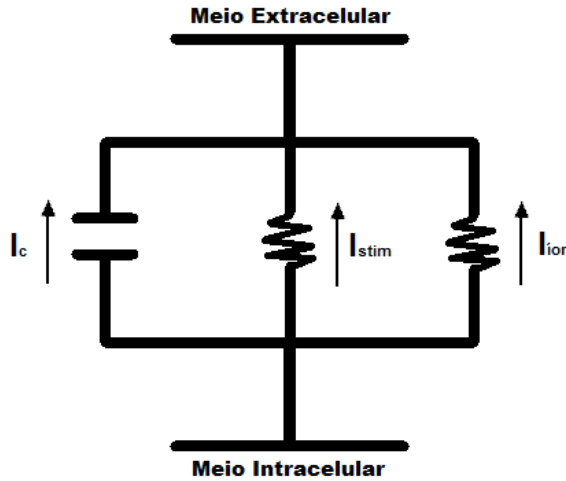


Figura 2.7: Modelo elétrico para a membrana celular.

Para um capacitor de capacitância C_m , sabe-se que $q = C_m V_c$, onde q é a carga elétrica acumulada no capacitor e V_c é a tensão sobre os seus terminais. Derivando-se ambos os membros de tal equação em relação ao tempo, obtemos a Equação 2.1. Como $\frac{dq}{dt} = I_c$, tem-se a Equação 2.2.

$$\frac{dq}{dt} = C_m \frac{dV_c}{dt} \quad (2.1)$$

$$I_c = C_m \frac{dV_c}{dt} \quad (2.2)$$

Aplicando-se a lei de Kirchhoff para correntes elétricas sobre o nó inferior do circuito da Figura 2.7, obtemos a Equação 2.3, onde C_m é a capacitância elétrica da membrana plasmática, I_{ion} é o somatório das correntes iônicas que atravessam os canais iônicos e I_{stim} é uma corrente de origem externa, denominada *corrente de estímulo*.

$$\frac{dV}{dt} = -\frac{1}{C_m} (I_{ion} + I_{stim}) \quad (2.3)$$

As diferenças nas concentrações fazem com que os íons se movam no sentido contrário ao dado pelo gradiente de concentração. Em contrapartida, a força do campo elétrico gerada pelo potencial transmembrânico dirige os íons no sentido contrário ao movimento

de difusão. Um equilíbrio será alcançado quando o fluxo difusivo dos íons se igualar ao fluxo devido à diferença de potencial elétrico. O valor do potencial transmembrânico para tal equilíbrio, relativo a um íon isolado, é dado pela equação de Nernst (Equação 2.4), onde R é a constante dos gases (8,314 J/Kmol), T é a temperatura absoluta, z é a valência do íon, F é a constante de Faraday ($9,648 \times 10^4$ C/mol), c_e e c_i denotam as concentrações externa e interna do íon, respectivamente.

$$v_{eq} = \frac{RT}{zF} \ln \left(\frac{c_e}{c_i} \right) \quad (2.4)$$

A Equação 2.4 é válida para o caso em que um único tipo de íon pode atravessar a membrana. Neste caso, na situação de equilíbrio, ou seja, $V = v_{eq}$, o fluxo do íon em questão através da membrana será nulo. Para o caso em que a membrana é permeável para dois ou mais íons diferentes, é utilizada a formulação de Goldman-Hodgkin-Katz (Keener e Sneyd, 1998), também chamada de equação GHK. A título de exemplo, considerando-se os íons K^+ e Na^+ , a equação GHK é dada por

$$v_{eq} = -\frac{RT}{F} \ln \left(\frac{P_{Na}[Na^+]_i + P_K[K^+]_i}{P_{Na}[Na^+]_e + P_K[K^+]_e} \right), \quad (2.5)$$

onde P_K e P_{Na} são as permissividades da membrana em relação aos íons K^+ e Na^+ , respectivamente.

2.4 Modelos para a Corrente Iônica

Os modelos para a relação corrente-voltagem (I-V) em determinado canal iônico mais utilizados em formulações de eletrofisiologia cardíaca são o modelo linear (Equação 2.6) e a equação GHK, cuja demonstração pode ser encontrada em Keener e Sneyd (1998). O modelo linear para o íon S , dado pela Equação 2.6, estabelece que a corrente iônica I_S é função linear da diferença entre o potencial transmembrânico e o potencial de repouso E_S do íon S , sendo g a condutância do canal iônico em questão.

$$I_S = g(V - E_S) \quad (2.6)$$

A equação GHK estabelece uma relação não-linear entre a corrente iônica e o potencial transmembrânico, sendo obtida a partir da hipótese de campo elétrico constante sobre a membrana sarcoplasmática (Keener e Sneyd, 1998). Sendo I_s a corrente relativa ao íon S , P_S a permissividade da membrana ao íon S , c_i e c_e as concentrações interna e externa, respectivamente, do íon S , temos que a equação GHK, para a situação de todos os canais iônicos abertos, é dada por:

$$I_s = P_s \frac{z^2 F^2}{RT} V \frac{(c_i - c_e) e^{\frac{-zFV}{RT}}}{1 - e^{\frac{-zFV}{RT}}} \quad (2.7)$$

Geralmente, a condutância g associada a uma corrente iônica é função do potencial transmembrânico ou de determinadas concentrações iônicas, sendo usualmente descrita através de modelos para o comportamento do canal iônico associado. Tais modelos são descritos na próxima seção.

2.5 Modelos para os Canais Iônicos

A ampla variedade de canais iônicos presentes na membrana celular demanda a realização de experimentos sofisticados para se avaliar quantitativamente seu comportamento. A metodologia mais utilizada atualmente é denominada *patch-clamp* (Keener e Sneyd, 2002), na qual se mede correntes iônicas através dos canais sob valores constantes de potencial transmembrânico. Através de observações experimentais, verifica-se que os canais iônicos alternam-se entre estados condutivos e não condutivos, o que leva à formulação de diversos tipos de modelos estocásticos para os mesmos.

2.5.1 Modelo de Dois Estados

Muitos modelos para o comportamento de canais iônicos descrevem um espaço de estados discreto para os mesmos, de modo que algumas configurações permitem a passagem de corrente iônica e outras não. O modelo mais simples para ilustrar esse conceito é o chamado *modelo de dois estados*, que estabelece dois possíveis estados para um canal iônico: aberto (O) e fechado (C) (Figura 2.8). Sendo n a proporção de canais no estado

O , pode-se formular a EDO (2.8) - a qual pode ser chamada de *equação do tipo Hodgkin-Huxley* (HH) - em função das taxas de transição $\alpha(V)$ (C para O) e $\beta(V)$ (O para C), as quais usualmente são modeladas como funções não lineares de V .

$$\frac{dn}{dt} = \alpha(V)(1 - n) - \beta(V)n \quad (2.8)$$

Fazendo-se

$$n_{\infty}(V) = \frac{\alpha(V)}{\alpha(V) + \beta(V)} \quad (2.9)$$

e

$$\tau_n(V) = \frac{1}{\alpha(V) + \beta(V)} \quad (2.10)$$

temos que a equação diferencial (2.8) toma a forma:

$$\frac{dn}{dt} = \frac{n_{\infty}(V) - n}{\tau_n(V)} \quad (2.11)$$

A Equação (2.11) torna conveniente o ajuste de parâmetros a ser realizado, uma vez que as funções $n_{\infty}(V)$ e $\tau_n(V)$, respectivamente chamadas de *valor assintótico de n* e *constante de tempo de n* , são mais facilmente obtidas através de procedimentos experimentais (Keener e Sneyd, 1998).

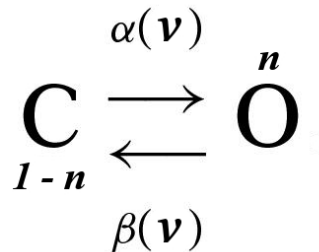


Figura 2.8: Modelo de dois estados.

Seja $g_{max,S}$ a condutância para a população de canais iônicos do íon S no caso de todos os canais estarem no estado O . Se n , dado por (2.8), é a proporção dos canais que encontram-se no estado O , temos que I_S é dada pela Equação (2.12) (utilizando-se o

modelo linear para a relação entre corrente e potencial transmembrânico).

$$I_S = ng_{max,S}(V - E_S) \quad (2.12)$$

2.5.2 Modelo de Subunidades

Podemos assumir, como uma generalização do modelo de dois estados para o canal iônico, a existência de múltiplas subunidades idênticas como componentes estruturais do canal, sendo cada uma delas sujeita aos estados *aberto* (O) ou *fechado* (C). Suponha, por exemplo, que determinado canal iônico seja composto de duas subunidades. Deste modo, o canal pode assumir qualquer um dos quatro estados E_{00} , E_{11} , E_{10} e E_{01} , onde os índices denotam as diferentes subunidades, com 1 e 0 denotando a subunidade aberta e fechada, respectivamente. No entanto, tal espaço de quatro estados somente é utilizado quando consideramos que as subunidades são distintas entre si, sendo que as taxas de transição α e β são características de cada subunidade (veja a Figura 2.9). Assumindo as subunidades idênticas entre si, podemos concluir que os estados E_{01} e E_{10} são equivalentes, de modo que é possível uma representação que gere um espaço de estados mais compacto.

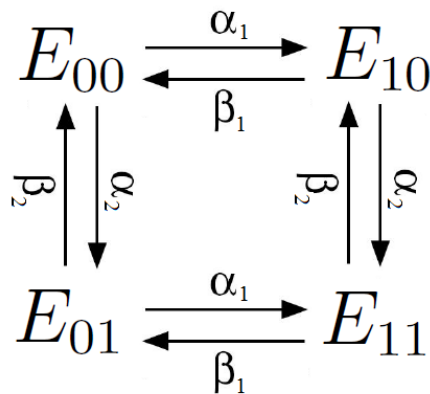


Figura 2.9: Espaço de estados para o modelo de duas subunidades distintas.

Considerando E_i o estado em que exatamente i subunidades encontram-se abertas, nosso exemplo tem seu espaço de estados reduzido para E_0 , E_1 e E_2 . A Figura 2.10 mostra o espaço de estados com as transições possíveis e suas respectivas taxas, sendo α a taxa de transição $C \rightarrow O$ de uma subunidade e β a taxa de transição $O \rightarrow C$ de uma

subunidade do canal iônico.

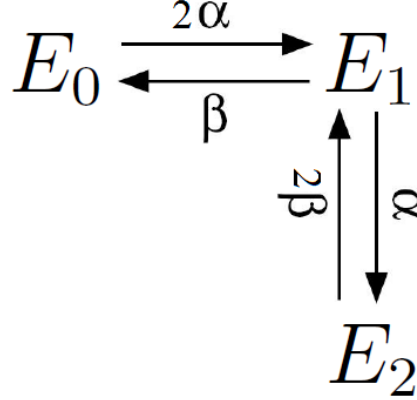


Figura 2.10: Espaço de estados para o modelo de duas subunidades idênticas.

Ambos os modelos representados nas Figuras 2.9 e 2.10 tratam-se de cadeias de Markov (veja a Seção 2.5.3), que resultam em equações com certa complexidade de resolução. Pode-se mostrar (Keener e Sneyd, 1998) que, sendo x_0 e x_2 as variáveis adimensionais associadas aos estados E_0 e E_2 , respectivamente, temos que $x_0 = (1 - n)^2$ e que $x_2 = n^2$, onde n é a variável adimensional dada pela Equação 2.8. De modo geral, tem-se que a condutância máxima em um canal iônico que contém k subunidades idênticas é dada por $n^k g_{max}$, onde n satisfaz a equação diferencial 2.8.

Observa-se em determinados canais iônicos, por exemplo os chamados *canais de corrente rápida de sódio*, uma rápida resposta condutiva à elevação do potencial transmembrânico, seguida de uma lenta fase de transição para o estado não-condutivo. Tais fases, para os canais de sódio, são chamadas, respectivamente, de *ativação* e *inativação*. Conforme proposto por Hodgkin e Huxley (1952), consideram-se três subunidades idênticas m para a ativação e uma subunidade h relacionada à inativação, sendo as equações resultantes dadas por:

$$\begin{aligned}
 I_S &= m^3 h g_{max} (V - E_S) \\
 \frac{dm}{dt} &= \frac{m_\infty(V) - m}{\tau_m(V)} \\
 \frac{dh}{dt} &= \frac{h_\infty(V) - h}{\tau_h(V)}
 \end{aligned} \tag{2.13}$$

2.5.3 Cadeias de Markov

Uma cadeia de Markov é uma forma particular de um processo estocástico em que considera-se um conjunto discreto de estados (ou seja, uma variável aleatória discreta) tal que a predição de estados futuros independe dos estados anteriores. Diversos dos novos modelos para células excitáveis têm empregado cadeias de Markov como alternativa ao modelo de subunidades para descrever o comportamento dos canais iônicos (Bondarenko *et al.*, 2004; Iyer *et al.*, 2004; Clancy e Rudy, 2002; S. Wang *et al.*, 1997), pois no último, o processo de ativação e o processo de inativação são modelados de forma independente. Contudo, experimentos revelam que a inativação de um canal iônico é um processo intrinsecamente acoplado à ativação (Aldrich *et al.*, 1983; Armstrong e Bezanilla, 1977), o que pode ser levado em consideração na modelagem de canais iônicos por cadeias de Markov.

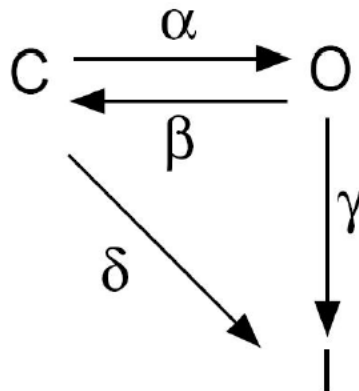


Figura 2.11: Cadeia de Markov com os estados aberto (O), fechado (C) e inativo (I).

A título de exemplo, consideremos um simples modelo de Markov, no qual o canal iônico pode estar em um dos três estados: aberto (O), fechado (C) ou inativo (I). Consideraremos neste exemplo que, uma vez no estado inativo, o canal não pode retornar para os estados aberto ou fechado. A Figura 2.11 ilustra o conjunto de estados e as possíveis transições, com suas respectivas taxas.

Ainda para o exemplo considerado, sendo i , o e c as proporções de canais iônicos do íon S nos estados inativo, aberto e fechado, respectivamente, a corrente iônica I_S para tal íon pode ser modelada pelo conjunto de EDOs (2.14), onde α , β , γ e δ são as taxas de

transição entre os estados, representadas na Figura (2.11).

$$\begin{aligned}
 I_S &= og_{S,max}(V - E_S) \\
 \frac{dc}{dt} &= -(\alpha + \delta)c + \beta o \\
 \frac{do}{dt} &= \alpha c - (\beta + \gamma)o \\
 \frac{di}{dt} &= \delta c + \gamma o
 \end{aligned} \tag{2.14}$$

Os estados inativo e fechado produzem o mesmo efeito de não permitirem a passagem de íons, ou seja, conferem estados não condutivos para o canal iônico em questão. Contudo, a transição do estado aberto para o estado inativo é geralmente favorecida em situações de valores positivos para o potencial transmembrânico, enquanto a transição do estado aberto para o estado fechado ocorre com mais frequência em potenciais mais negativos (Hille, 2001).

A biologia molecular tem sido fundamental na compreensão de fenômenos fisiológicos e na obtenção de informações acerca da estrutura e da função dos canais iônicos. Graças a tal área do conhecimento, é possível capturar detalhes sobre a conformação proteica dos canais iônicos e as alterações morfológicas promovidas por drogas ou por doenças genéticas.

As alterações das propriedades eletrofisiológicas decorrentes das doenças cardíacas podem se refletir em mudanças nas taxas de transição de modelos de Markov para os canais iônicos. Portanto, a utilização de tal formalismo na modelagem do comportamento de canais iônicos pode vir a ser adequada em futuros estudos sobre consequências fisiológicas associadas a patologias cardíacas, bem como no processo de desenvolvimento de novos medicamentos para o tratamento de tais doenças. Maiores detalhes sobre a relação entre a estrutura molecular dos canais iônicos e as cadeias de Markov podem ser encontrados em Hille (2001).

Modelos baseados em cadeias de Markov para os canais iônicos podem assumir elevada complexidade, levando em consideração um grande número de estados e,

consequentemente, demandando o ajuste de uma grande quantidade de parâmetros (taxas de transição), os quais podem ser determinados somente com a utilização das mais avançadas técnicas de medição experimental. Ilustramos essa possível complexidade com a Figura 2.12, que mostra a cadeia de Markov para os canais iônicos da chamada *corrente rápida de sódio*, considerada no modelo para miócitos de camundongo proposto por Bondarenko *et al.* (2004).

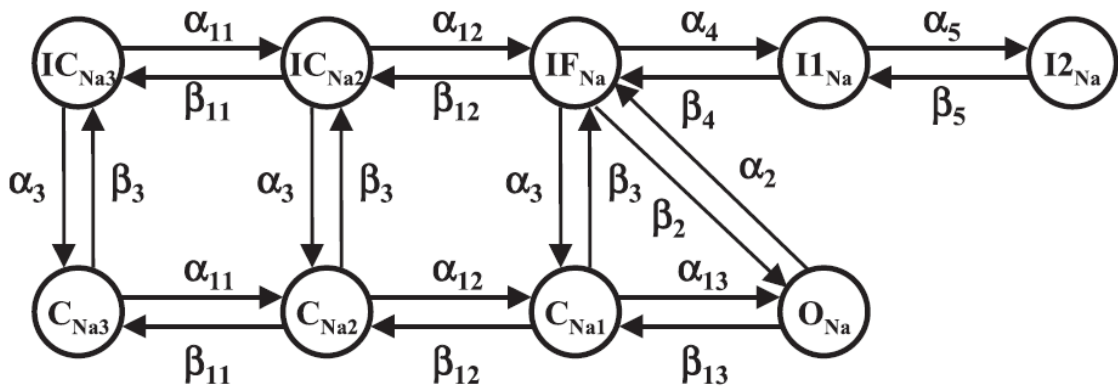


Figura 2.12: Cadeia de Markov para os Canais Iônicos da *Corrente Rápida de Sódio* (Bondarenko *et al.*, 2004).

2.6 O Modelo de Hodgkin e Huxley (1952)

O modelo de Hodgkin e Huxley descreve a eletrofisiologia da membrana do axônio gigante de lula e foi desenvolvido a partir de medidas do comportamento elétrico passivo e ativo da célula (Hodgkin e Huxley, 1952). A base da descrição do potencial de ação proposto por Hodgkin e Huxley é o comportamento dos canais de sódio e de potássio, conforme ilustrado pela Figura 2.13.

Conforme descrevemos em seções anteriores, o Potencial de Ação é dividido nas fases de repouso, despolarização, repolarização e hiperpolarização. Os principais componentes responsáveis pelo comportamento do potencial transmembrânico em tais fases são os canais iônicos de sódio e de potássio, os quais sofrem variações em suas condutividades durante a trajetória do PA. Esta relação entre potencial de ação e as variações das condutividades dos canais iônicos de sódio e de potássio pode ser observada nas Figuras 2.14 e 2.15.

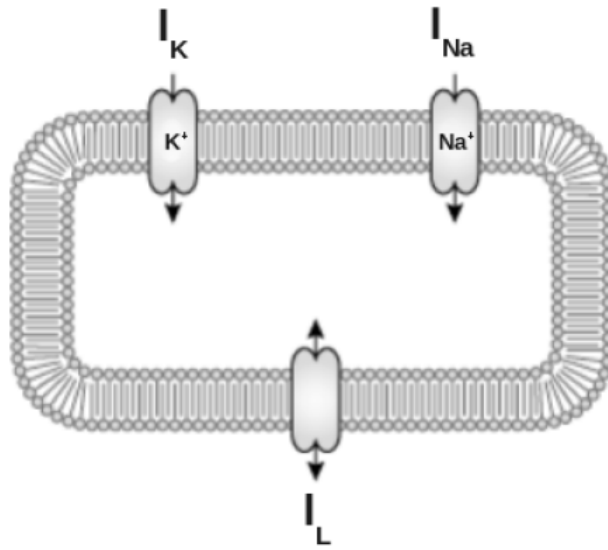


Figura 2.13: Esquematização da membrana celular e dos canais iônicos considerados no modelo de Hodgkin e Huxley (1952).

Conforme discutido em seções anteriores, é possível a obtenção de uma equação diferencial para o potencial transmembrânico V em função da corrente de estímulo e do somatório de correntes que atravessam a membrana, conforme mostrado na equação abaixo:

$$\frac{dV}{dt} = -\frac{1}{C_m}(I_m + I_{stim}) \quad (2.15)$$

onde C_m é a capacitância da membrana, I_m é a corrente transmembrânica e I_{stim} é uma corrente de estímulo. O modelo de Hodgkin e Huxley leva em consideração apenas as correntes de sódio e de potássio, sendo sua corrente transmembrânica, portanto, dada por

$$I_m = I_{Na} + I_K + I_l ,$$

sendo I_{Na} a corrente de sódio, I_K a corrente de potássio e I_l uma corrente de fuga. A corrente de fuga I_l é um somatório de diferentes correntes iônicas, majoritariamente de cloro. As equações para as correntes iônicas são dadas por

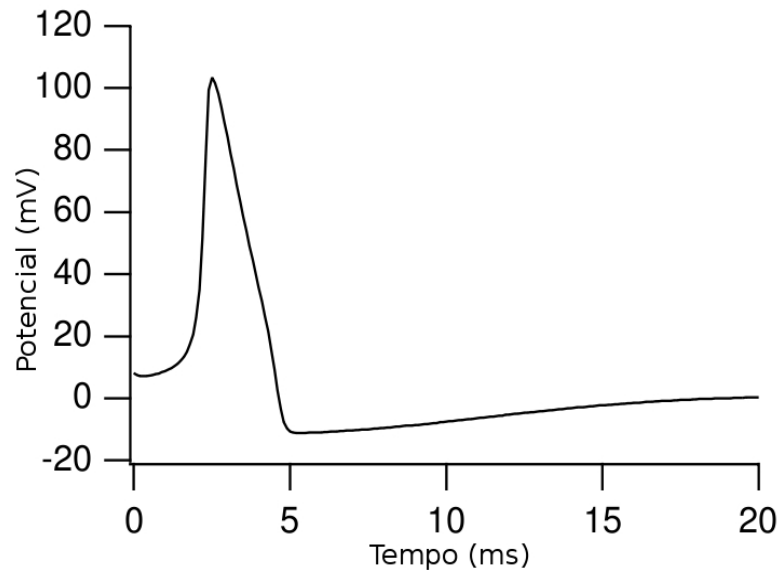


Figura 2.14: Potencial de ação do modelo de Hodgkin e Huxley (1952) (adaptado de Keener e Sneyd (1998)).

$$I_{Na} = g_{Na}(V_m - E_{Na}) \quad (2.16)$$

$$I_K = g_K(V_m - E_K) \quad (2.17)$$

$$I_l = g_l(V_m - E_l) ,$$

onde g_{Na} , g_K e g_l são as condutâncias associadas às correntes e E_{Na} , E_K e E_l são os potenciais de Nernst relativos aos respectivos íons envolvidos. Assume-se que a condutividade g_l é constante e que as demais condutividades variam com tempo e são dependentes do potencial transmembrânico. Além disso, o modelo considera as concentrações iônicas constantes e, consequentemente, os potenciais de Nernst constantes.

A condutância associada à corrente de sódio g_{Na} é dada por

$$g_{Na} = m^3 h \bar{g}_{Na} ,$$

onde $\bar{g}_{Na}$ é a condutividade máxima de sódio, m é a variável adimensional associada às subunidades de ativação do canal iônico e h é a variável adimensional de inativação. As taxas de transição α_m , β_m , α_h e β_h são funções não lineares do potencial transmembrânico.

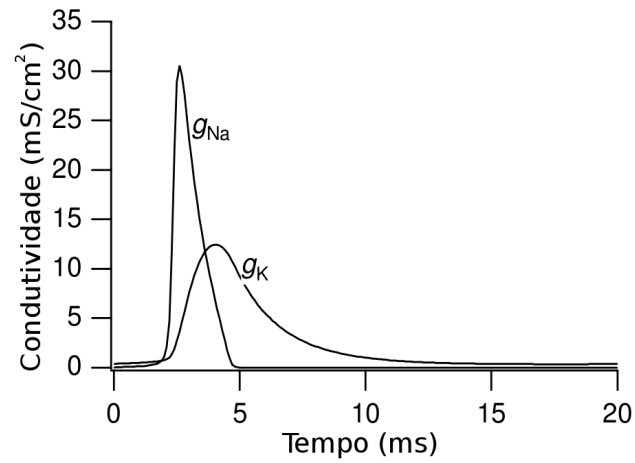


Figura 2.15: Variação das condutividades g_{Na} e g_K do modelo Hodgkin-Huxley durante um potencial de ação (adaptado de Keener e Sneyd (1998)).

As equações diferenciais para as variáveis m e h são

$$\begin{aligned}\frac{dm}{dt} &= \alpha_m(1 - m) - \beta_m m \\ \frac{dh}{dt} &= \alpha_h(1 - h) - \beta_h h .\end{aligned}$$

A condutância associada à corrente de potássio g_K é dada por

$$g_K = \bar{g}_K n^4 ,$$

onde $\bar{g}_K$ representa a condutividade máxima de potássio e n é a proporção de subunidades de canais de potássio no estado condutivo, sendo controlada pelas taxas de transição α_n e β_n :

$$\frac{dn}{dt} = \alpha_n(1 - n) - \beta_n n \quad (2.18)$$

As funções não lineares para as taxas de transição do modelo são dadas por:

$$\alpha_m = 0.1 \frac{25 - V_m}{\exp\left(\frac{25 - V_m}{10}\right) - 1}, \quad (2.19)$$

$$\beta_m = 4 \exp\left(\frac{-V_m}{18}\right), \quad (2.20)$$

$$\alpha_h = 0.07 \exp\left(\frac{-V_m}{20}\right), \quad (2.21)$$

$$\beta_h = \frac{1}{\exp\left(\frac{30 - V_m}{10}\right) + 1}, \quad (2.22)$$

$$\alpha_n = 0.01 \frac{10 - V_m}{\exp\left(\frac{10 - V_m}{10}\right) - 1}, \quad (2.23)$$

$$\beta_h = 0.125 \exp\left(\frac{-V_m}{80}\right). \quad (2.24)$$

2.7 O Modelo de Bondarenko *et al.* (2004)

O modelo de eletrofisiologia cardíaca proposto por Bondarenko *et al.* (2004) descreve quantitativamente a atividade iônica em miócitos ventriculares oriundos do ápice e do septo cardíacos de camundongos. O modelo busca explicar as variações regionais observadas na fase de repolarização dos miócitos de camundongos a partir das diferentes expressões das correntes de potássio em tais regiões. Através de um conjunto de 45 EDOs, o modelo simula correntes iônicas, bombas iônicas e a homeostase celular para a reprodução do potencial de ação. Os fluxos iônicos considerados pelo modelo estão representados na Figura 2.16, os quais determinam a seguinte EDO para o potencial transmembrânico:

$$\begin{aligned} -C_m \frac{dV}{dt} = & I_{CaL} + I_{p(Ca)} + I_{NaCa} + I_{Cab} + I_{Na} + I_{Nab} \\ & + I_{NaK} + I_{Ktof} + I_{K,tos} + I_{K1} + I_{Ks} + I_{Kur} \\ & + I_{Kss} + I_{Kr} + I_{Cl,Ca} + I_{app} \end{aligned} \quad (2.25)$$

Algumas das equações para o comportamento de canais iônicos são baseadas no formalismo de Hodgkin e Huxley (1952), porém estão presentes no modelo cadeias de Markov que modelam, por exemplo, as correntes iônicas $I_{Ca,L}$, I_{Na} e I_{Kr} . A cadeia de Markov para a corrente I_{Na} (*fast Na^+ current*) foi mostrada na Figura 2.12. O conjunto completo de equações que modelam a corrente I_{Na} encontra-se abaixo:

$$I_{Na} = G_{Na} O_{Na} (V - E_{Na}) \quad (2.26)$$

$$E_{Na} = \frac{RT}{F} \ln \frac{0.9[Na^+]_o + 0.1[K^+]_o}{0.9[Na^+]_i + 0.1[K^+]_i} \quad (2.27)$$

$$C_{Na3} = 1 - (O_{Na} + C_{Na1} + C_{Na2} + IF_{Na} + I1_{Na} + I2_{Na} + IC_{Na2} + IC_{Na3}) \quad (2.28)$$

$$\begin{aligned} \frac{dC_{Na2}}{dt} = & \alpha_{Na11} C_{Na3} - \beta_{Na11} C_{Na2} + \beta_{Na12} C_{Na1} - \alpha_{Na12} C_{Na2} \\ & + \alpha_{Na3} IC_{Na2} - \beta_{Na3} C_{Na2} \end{aligned} \quad (2.29)$$

$$\begin{aligned} \frac{dC_{Na1}}{dt} = & \alpha_{Na12} C_{Na2} - \beta_{Na12} C_{Na1} + \beta_{Na13} O_{Na} - \alpha_{Na13} C_{Na1} \\ & + \alpha_{Na3} IF_{Na} - \beta_{Na3} C_{Na1} \end{aligned} \quad (2.30)$$

$$\frac{dO_{Na}}{dt} = \alpha_{Na13} C_{Na1} - \beta_{Na13} O_{Na} + \beta_{Na2} IF_{Na} - \alpha_{Na2} O_{Na} \quad (2.31)$$

$$\begin{aligned} \frac{dIF_{Na}}{dt} = & \alpha_{Na2} O_{Na} - \beta_{Na2} IF_{Na} + \beta_{Na3} C_{Na1} - \alpha_{Na3} IF_{Na} \\ & + \beta_{Na4} I1_{Na} - \alpha_{Na4} IF_{Na} + \alpha_{Na12} IC_{Na2} - \beta_{Na12} IF_{Na} \end{aligned} \quad (2.32)$$

$$\frac{dI1_{Na}}{dt} = \alpha_{Na4} IF_{Na} - \beta_{Na4} I1_{Na} + \beta_{Na5} I2_{Na} - \alpha_{Na5} I1_{Na} \quad (2.33)$$

$$\frac{dI2_{Na}}{dt} = \alpha_{Na5} I1_{Na} - \beta_{Na5} I2_{Na} \quad (2.34)$$

$$\begin{aligned} \frac{dIC_{Na2}}{dt} = & \alpha_{Na11} IC_{Na3} - \beta_{Na11} IC_{Na2} + \beta_{Na12} IF_{Na} - \alpha_{Na12} IC_{Na2} \\ & + \beta_{Na3} C_{Na2} - \alpha_{Na3} IC_{Na2} \end{aligned} \quad (2.35)$$

$$\frac{dIC_{Na3}}{dt} = \beta_{Na11} IC_{Na2} - \alpha_{Na11} IC_{Na3} + \beta_{Na3} C_{Na3} - \alpha_{Na3} IC_{Na3} \quad (2.36)$$

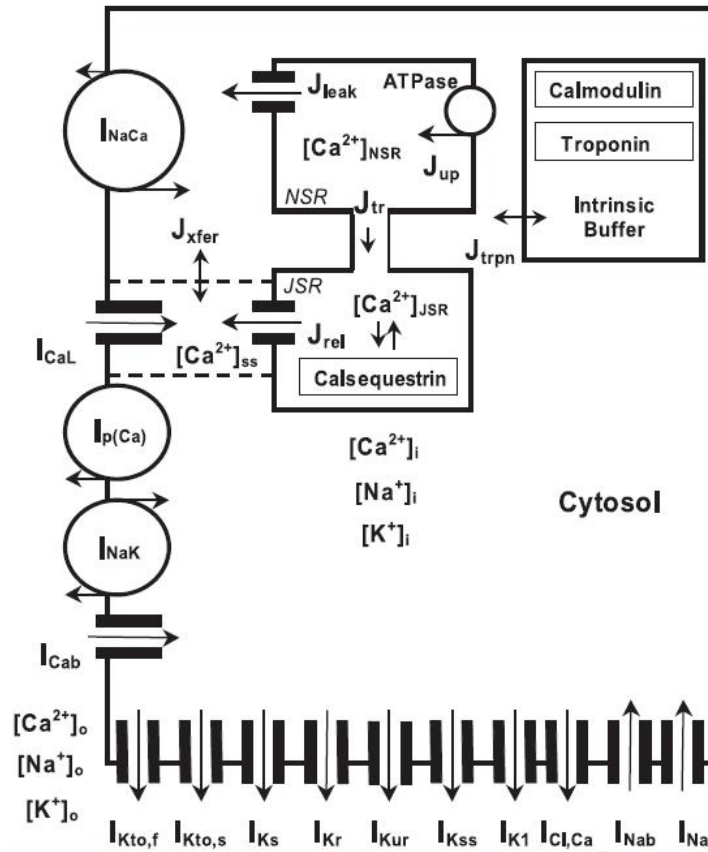


Figura 2.16: Ilustração esquemática dos fluxos iônicos e compartimentos celulares considerados pelo modelo de Bondarenko *et al.* (Imagem extraída de Bondarenko *et al.* (2004)).

2.8 Modelo Generalizado para a Membrana Celular

Para facilitar referências futuras, iremos propor uma notação generalizada para as variáveis associadas aos modelos de eletrofisiologia celular utilizados neste trabalho. Seja um modelo genérico cujas variáveis associadas a EDOs (variáveis de estado) são representadas pelo vetor $\vec{Y}$, com N_{var} componentes. Iremos categorizar cada equação diferencial em uma das três possíveis classes: equação não-linear (NL); equação do tipo Hodgkin-Huxley (HH) ou *quasi-linear* (tais equações são associadas às *gate variables*); e equação de cadeia de Markov (MK).

A equação para o potencial transmembrânico desse modelo é a primeira das equações NL e tem a forma da Equação (2.37), onde I_i é a corrente iônica de índice i que atravessa os canais da membrana, C_m é a capacitância da membrana, V é o potencial transmembrânico e N_{ion} é o número de correntes iônicas do modelo. As variáveis não-lineares restantes y_{nl}^j

têm equações na forma da Equação (2.38), onde f_{nl}^j são funções completamente não-lineares e j varia de 2 até N_{nl} , o qual é o número de equações do tipo NL.

Considere, ainda, que tenhamos N_{hh} *gate variables* no modelo, cada uma delas associada a uma equação do tipo HH. Seja y_{hh}^j uma dessas variáveis, com $j = 1, 2, \dots, N_{hh}$. Suas equações correspondentes são descritas pela Equação (2.39), as quais podem ser consideradas equações *quasi-lineares*, apesar do fato de α^j e β^j serem funções não-lineares do potencial transmembrânico.

Finalmente, as variáveis de cadeias de Markov, denotadas por y_{mk}^j , são descritas pela Equação (2.40), onde N_{mk} é o número dessas variáveis e $j = 1, 2, \dots, N_{mk}$. Na Equação (2.40), α_j^i é a taxa de transição do estado y_{mk}^i para o estado y_{mk}^j em uma cadeia de Markov. Claramente, se y_{mk}^i e y_{mk}^j pertencem a diferentes cadeias de Markov, então $\alpha_i^j = \alpha_j^i = 0$.

$$\frac{dy_{nl}^1}{dt} = \frac{dV}{dt} = -\frac{1}{C_m} \sum_{i=1}^{N_{ion}} I_i \quad (2.37)$$

$$\frac{dy_{nl}^j}{dt} = f_{nl}^j(\vec{Y}, t), \quad j = 2, \dots, N_{nl} \quad (2.38)$$

$$\frac{dy_{hh}^j}{dt} = f_{hh}^j(\vec{Y}, t) = \alpha^j(1 - y_{hh}^j) - \beta^j y_{hh}^j, \quad j = 1, \dots, N_{hh} \quad (2.39)$$

$$\frac{dy_{mk}^j}{dt} = f_{mk}^j(\vec{Y}, t) = -y_{mk}^j \sum_{i=1, i \neq j}^{N_{mk}} \alpha_i^j + \sum_{i=1, i \neq j}^{N_{mk}} \alpha_j^i y_{mk}^i, \quad j = 1, \dots, N_{mk} \quad (2.40)$$

Após todas essas classificações explícitas, neste documento poderemos também generalizar nosso modelo descrevendo-o por

$$\frac{dy^j}{dt} = f^j(\vec{Y}, t), \quad j = 1, \dots, N_{var} \quad (2.41)$$

onde a derivada no tempo da variável genérica y^j é $f^j(\vec{Y}, t)$. As equações descritas acima também podem ser referenciadas em formato vetorial, conforme mostrado pelas equações abaixo:

$$\vec{Y}'_{nl} = \vec{F}_{nl}(\vec{Y}, t) , \quad (2.42)$$

$$\vec{Y}'_{hh} = \vec{F}_{hh}(\vec{Y}, t) , \quad (2.43)$$

$$\vec{Y}'_{mk} = \vec{F}_{mk}(\vec{Y}, t) , \quad (2.44)$$

onde $\vec{Y}_{nl}$, $\vec{Y}_{hh}$ e $\vec{Y}_{mk}$ são os vetores correspondentes às variáveis do tipo NL, HH e MK, respectivamente. Os vetores $\vec{F}_{nl}$, $\vec{F}_{hh}$ e $\vec{F}_{mk}$ representam as funções de lado direito para tais grupos de variáveis. Para o sistema de equações completo, utilizamos a seguinte equação vetorial:

$$\vec{Y}' = \vec{F}(\vec{Y}, t) \quad (2.45)$$

onde $\vec{F}$ é a função vetorial que representa o lado direito do sistema de EDOs.

3 Métodos Numéricos

Este capítulo é destinado à apresentação dos métodos numéricos existentes na literatura que foram aplicados neste trabalho para a realização de experimentos computacionais e como base para a geração de novos esquemas numéricos para a solução de modelos da eletrofisiologia de células cardíacas.

3.1 Método de Euler Explícito

O método de Euler explícito é um método de 1ª ordem para a solução aproximada de equações diferenciais ordinárias. Considere a EDO $\frac{dy}{dt} = f(y, t)$, onde $f(y, t)$ é uma função dada e $y(t)$ é uma função a ser determinada (ou aproximada). Expandindo-se $y(t)$ em uma série de Taylor em torno de t , tem-se

$$y(t+h) = y(t) + hf(y(t), t) + \frac{h^2}{2!}f^{(1)}(y(t), t) + \frac{h^3}{3!}f^{(2)}(y(t), t) + \dots, \quad (3.1)$$

onde h é um valor de pequena magnitude denominado *passo*. Tomando-se uma aproximação para $y(t+h)$ a partir do truncamento de termos cuja potência de h seja superior a 2, ou seja, termos de ordem superior a 2, temos

$$\bar{y}(t+h) = y(t) + hf(y(t), t) = y(t+h) + O(h^2). \quad (3.2)$$

Sejam $y_0 = y(t_0)$ e $y_1 = \bar{y}(t_0 + h)$, para $n = 2, 3, \dots$ é estabelecido pelo método de Euler que

$$y_n = y_{n-1} + hf(y_{n-1}, t_{n-1}), \quad (3.3)$$

onde $t_n = t_0 + nh$. Pode-se demonstrar (Burden e Faires, 2008) que o erro cometido ao aproximar-se $y(t_0 + nh)$ por y_n é da ordem de h . Além disso, pode-se verificar que o método de Euler explícito possui estabilidade condicionada ao passo utilizado e às

características da EDO a ser aproximada, sendo classificado, portanto, como um método condicionalmente estável. Em notação vetorial, o método de Euler explícito é dado por:

$$\vec{Y}_{n+1} = \vec{Y}_n + h\vec{F}(\vec{Y}_n, t_n) . \quad (3.4)$$

3.2 Método Runge-Kutta de 2ª Ordem

Os métodos de Runge-Kutta - apresentados detalhadamente por Holmes (2007) - são métodos explícitos, condicionalmente estáveis, também para a solução numérica de EDOs, com formulações para diferentes ordens de precisão. Neste trabalho, utilizaremos a versão de 2ª ordem, doravante referida por RK2 ou método de Heun, a qual é ilustrada pelas equações abaixo, onde t_0 é o tempo inicial adotado.

$$\begin{aligned} t_n &= nh + t_0 \\ y_0 &= y(t_0) \\ f_n &= f(y_n, t_n) \\ y_{n+1} &= y_n + \frac{h}{2}(f_n + f(y_n + hf_n, t_n + h)) \end{aligned} \quad (3.5)$$

3.3 Método de Euler Adaptativo

O método de Euler adaptativo (ou Euler / ADP) trata-se de uma extensão do método de Euler para a utilização de passos de tempo adaptativos (Campos *et al.*, 2013). Tal método utiliza, a cada passo, avaliações da solução pelo método de Euler e pelo método de Heun para estimar erros numéricos locais e adaptar o tamanho do passo de tempo de acordo com uma tolerância predeterminada. Os itens a seguir descrevem as operações executadas a cada passo da simulação:

1. Dado $\vec{Y}_n$, calcule $\vec{Y}_{n+1} = \vec{Y}_n + h\vec{F}(\vec{Y}_n, t_n)$.
2. Calcule $\vec{Y}_{n+1} = \vec{Y}_n + \frac{h}{2}(\vec{F}(\vec{Y}_n, t_n) + \vec{F}(\vec{Y}_{n+1}, t_n + h))$. Note que o cálculo de

$\vec{F}(\vec{Y}_{n+1}, t_n + h)$ será reutilizado na primeira etapa do próximo passo.

3. Calcule o máximo erro relativo encontrado entre as duas soluções:

$$erro_{local} = \max_{1 \leq j \leq N_{var}} \left| \frac{y_{n+1}^j - \bar{y}_{n+1}^j}{\bar{y}_{n+1}^j} \right| \quad (3.6)$$

4. Dada uma tolerância de erro local τ , se $erro_{local} \leq \tau$, então o passo de tempo será atualizado e dado por $h_{new} = h \sqrt{\frac{0.5 \times \tau}{erro_{local}}}$, respeitando um valor máximo (h_{max}) e um valor mínimo (h_{min}) predefinidos. Se $erro_{local} > \tau$, então o passo atual é descartado e calculado novamente com o passo de tempo reduzido à metade.

3.4 Método de Rush-Larsen

O método proposto por Rush e Larsen (Rush e Larsen, 1978) (denotaremos este método por RL) é largamente utilizado em *solvers* numéricos para modelos de eletrofisiologia que contém as chamadas *gating variables* provenientes da formulação de (Hodgkin e Huxley, 1952) para canais iônicos. As equações associadas a tais variáveis possuem a forma da Equação (2.39) e são comumente chamadas de equações *quasi-lineares*.

Esse método assume que os coeficientes α^j e β^j da Equação (2.39) são aproximadamente constantes em um pequeno intervalo de tempo, embora sejam funções não lineares do potencial transmembrânico. Portanto, o método consiste na linearização local das equações *quasi-lineares*, onde as mesmas são calculadas pela Equação (3.7), a qual é a solução exata das EDOs lineares resultantes.

$$y_{hh,n+1}^j = (y_{hh,n}^j - \frac{\alpha^j}{\alpha^j + \beta^j})e^{-(\alpha^j + \beta^j)h} + \frac{\alpha^j}{\alpha^j + \beta^j} \quad (3.7)$$

As demais equações do modelo, ou seja, aquelas não *quasi-lineares*, são resolvidas pelo método de Euler explícito, fazendo com que o método resultante seja de primeira ordem, com maior estabilidade para modelos com um grande número de equações *quasi-lineares* (MacLachlan *et al.*, 2005).

3.5 Método de Uniformização

O método de uniformização (Melamed e Yadin, 1984) é tradicionalmente utilizado para a análise da confiabilidade de redes e sistemas computacionais. Trata-se de um método com boas condições de estabilidade para a solução numérica de EDOs associadas a cadeias de Markov. Esse método será referenciado neste texto como UNI.

Uma *cadeia de Markov* (Jensen, 1953) é um processo estocástico definido em um conjunto discreto de estados onde a probabilidade condicional associada a um estado futuro depende apenas do estado presente. Uma cadeia de Markov pode ser de tempo discreto (a variável tempo é finita ou enumerável) ou de tempo contínuo (a variável tempo é contínua). Do ponto de vista estatístico, o método de Uniformização baseia-se na transformação de uma cadeia de Markov de tempo contínuo (CMTC) em uma cadeia de Markov de tempo discreto (CMTD) submetida ao processo de Poisson (Reibman e Trivedi, 1988).

Seja A a matriz de transições de uma CMTC dada por $[A]_{ij} = \alpha_i^j$ para $i \neq j$ e

$$[A]_{jj} = - \sum_{i=1, i \neq j}^{N_{mk}} \alpha_j^i,$$

onde α_i^j são as taxas de transição definidas na Seção 2.8. Considere ainda que $\vec{Y}_{mk}(t)$ seja o vetor de probabilidades associado a essa CMTC. Se considerarmos os termos $[A]_{ij}$ constantes desde o instante 0 até o instante t , o sistema de EDOs associado à CMTC, $\vec{Y}'_{mk}(t) = A\vec{Y}_{mk}(t)$, será linear. Portanto, a solução do sistema será dada por $\vec{Y}_{mk}(t) = e^{tA}\vec{Y}_{mk}(0)$, onde a exponencial matricial é definida pela série de Taylor da seguinte maneira:

$$e^{tA} = \sum_{i=0}^{\infty} \frac{(tA)^i}{i!}$$

A utilização direta da matriz A leva a erros numéricos e a problemas de convergência para o cálculo da série, uma vez que os termos diagonais de A são negativos e os demais termos são positivos. Além disso, pode haver termos com magnitude maior do que 1, exigindo a normalização de A . A técnica de uniformização define $A^* = A/q + I$, onde

$q \geq \max_{1 \leq i \leq n} |[A]_{ii}|$ e n é o número de estados da CMTC. Como $A = q(A^* - I)$, segue que $\vec{Y}_{mk}(t) = e^{qt(A^* - I)} \vec{Y}_{mk}(0) = e^{-qt} e^{qtA^*} \vec{Y}_{mk}(0)$, então a solução resultante utilizando-se a série de Taylor é dada pela equação abaixo (Sidje e Stewart, 1996).

$$\vec{Y}_{mk}(t) = e^{-qt} \left[\sum_{i=0}^{\infty} \frac{(qtA^*)^i}{i!} \right] \vec{Y}_{mk}(0)$$

A formulação final para o tempo t até $t + h$ é ilustrada pelas Equações (3.8), (3.9) e (3.10), onde $A(t)$ é a matriz de transição da CMTC avaliada no tempo t e $\vec{Y}_{mk}(t)$ é o vetor de probabilidades correspondente, contendo as variáveis da CMTC como suas componentes. O truncamento do somatório na Equação (3.8) é determinado pelo parâmetro N , o qual pode ser determinado pela Equação (3.11), dada uma tolerância λ . O valor λ é um limitante superior para o erro absoluto de truncamento obtido em cada componente de $\vec{Y}_{mk}(t + h)$ (Melamed e Yadin, 1984).

$$\vec{Y}_{mk,n+1} = e^{-q(t_n)h} \left[\sum_{i=0}^{N(t_n)} \frac{(A^*(t_n)q(t_n)h)^i}{i!} \right] \vec{Y}_{mk,n} \quad (3.8)$$

$$A^*(t_n) = \frac{A(t_n)}{q(t_n)} + I \quad (3.9)$$

$$q(t_n) \geq \max_{1 \leq i \leq N_{mk}} |[A(t_n)]_{ii}| \quad (3.10)$$

$$\lambda \leq 1 - e^{-q(t_n)h} \sum_{i=0}^N \frac{(q(t_n)h)^i}{i!} \quad (3.11)$$

3.6 Método de Sundnes *et al.*

Em 2009, Sundnes *et al.* propuseram uma extensão do método de Rush-Larsen na qual equações semelhantes a (3.7) são utilizadas para a solução de todas as EDOs em modelos de eletrofisiologia (Sundnes *et al.*, 2009). A partir de uma linearização local de todas as EDOs do sistema e através de uma simplificação por diagonalização do Jacobiano, pode-se chegar à equação (3.12), onde $t_n = t_0 + nh$, y_n^j representa a aproximação da j -ésima variável do sistema avaliada no tempo t_n , $\vec{Y}_n = (y_n^1, y_n^2, \dots, y_n^m)$ é o vetor aproximado de variáveis do sistema no instante t_n e $f^j(\vec{Y}, t)$ é o lado direito da EDO correspondente à

variável y^j , ou seja, $\frac{dy^j}{dt} = f^j(\vec{Y}, t)$. A derivada parcial presente na formulação, denotada por k_n , pode ser obtida por uma diferença finita para trás.

$$\begin{aligned} y_{n+1}^j &= y_n^j + \frac{f^j(\vec{Y}_n, t_n)}{k_n} (e^{hk} - 1) \\ k_n &= \frac{\partial f^j(\vec{Y}_n, t)}{\partial y^j} \end{aligned} \quad (3.12)$$

A aplicação da Equação (3.12) sobre todas as EDOs do sistema constitui a versão de 1ª ordem do método de Sundnes, a qual chamaremos de método SAST1. Pode-se facilmente verificar que a Equação (3.12) aplicada às equações do tipo *quasi-linear* - ilustradas pela Equação (2.39) - é equivalente à formulação (3.7), o que dispensa, para tais equações, o cálculo numérico da derivada parcial de $f^j(k_n)$, a qual pode ser obtida analiticamente.

Originalmente, o método proposto por Sundnes *et al.* (2009) consiste na realização dos dois seguintes passos principais:

1. Seja $\vec{Y}_n$ o vetor solução calculado mais recentemente, calculemos $\vec{Y}_{n+1/2}$ através das seguintes equações, para $i = 1, 2, \dots, m$:

$$y_{n+1/2}^i = y_n^i + \frac{f^i(\vec{Y}_n, t_n)}{k_n} (e^{\frac{h}{2}k} - 1) \quad (3.13)$$

$$k_n = \frac{\partial f^j(\vec{Y}_n, t)}{\partial y^j} \quad (3.14)$$

2. Para cada uma das variáveis y^j , tomemos o vetor $\vec{W}^j = (y_{n+1/2}^1, y_{n+1/2}^2, \dots, y_n^j, \dots, y_{n+1/2}^m)$ e calculemos o valor de y_{n+1}^j através das equações:

$$y_{n+1}^j = y_n^j + \frac{f^j(\vec{W}^j, t_n)}{\bar{k}_n} (e^{h\bar{k}_n} - 1) \quad (3.15)$$

$$\bar{k}_n = \frac{\partial f^j(\vec{W}^j, t)}{\partial y^j} \quad (3.16)$$

O método descrito pelos dois passos anteriores será referenciado neste trabalho por SAST2 o qual possui precisão de segunda ordem, conforme mostrado experimentalmente em Sundnes *et al.* (2009).

3.7 Métodos de Rush-Larsen de 2ª Ordem

Nesta seção apresentamos uma coleção de extensões de 2ª ordem para o método de Rush-Larsen, proposta por Perego e Veneziani (2009). Considere uma equação do tipo HH escrita na seguinte forma:

$$\frac{dy_{hh}^j}{dt} = a^j y_{hh}^j + b^j, \quad (3.17)$$

onde

$$a^j = -(\alpha^j + \beta^j), \quad b^j = \alpha^j. \quad (3.18)$$

Lembramos que α^j e β^j são funções não lineares do potencial da transmembrânico V e que são definidos pelas Equações (2.39). A família de métodos apresentada em Perego e Veneziani (2009) utiliza uma abordagem de 2ª ordem, como o RK2, para as Equações do tipo NL (2.37), (2.38) e a formulação dada pela Equação (3.19) para as Equações HH (2.39).

$$y_{n+1}^j = e^{a_{n+1/2}^j h} \left(y_n^j + \frac{b_{n+1/2}^j}{a_{n+1/2}^j} \right) - \frac{b_{n+1/2}^j}{a_{n+1/2}^j} \quad (3.19)$$

Tal formulação, assim como no método de Rush-Larsen original, assume que a^j e b^j são contantes em um pequeno intervalo de tempo. No entanto, os valores adotados para a^j e b^j em cada passo de tempo são tomados no ponto médio entre $t = t_n$ e $t = t_{n+1}$, ou seja, no instante $t = t_{n+1/2}$. Os valores $a_{n+1/2}^j$ e $b_{n+1/2}^j$ são estimados por uma combinação linear de valores discretos para a^j e b^j , como mostrado nas equações:

$$\begin{aligned}
a_{n+1/2}^j &= c_{-1}a_{n+1}^j + c_0a_n^j + c_1a_{n-1}^j, \\
b_{n+1/2}^j &= c_{-1}b_{n+1}^j + c_0b_n^j + c_1b_{n-1}^j .
\end{aligned} \tag{3.20}$$

Para garantir precisão de segunda ordem, c_{-1} , c_0 e c_1 são restritos pelas relações

$$\begin{aligned}
c_{-1} &= \theta/2 + 1/4 , \\
c_0 &= 1 - \theta , \\
c_1 &= \theta/2 - 1/4 ,
\end{aligned} \tag{3.21}$$

onde θ é um parâmetro que pode ser escolhido com base em questões de estabilidade e de eficiência. Conforme a escolha do parâmetro θ , podem ser gerados esquemas explícitos ou implícitos. Em Perego e Veneziani (2009) são apresentadas versões implícitas, explícitas e esquemas do tipo preditor-corretor com passo de tempo adaptativo. Neste trabalho será apresentado um esquema de passo adaptativo baseado no método de Rush-Larsen com baixo custo computacional por passo de tempo.

4 Métodos Propostos e Implementações

Neste capítulo apresentamos os novos esquemas numéricos desenvolvidos neste trabalho e detalhes importantes acerca de suas respectivas implementações. Adicionalmente, descreveremos os experimentos computacionais realizados neste trabalho, juntamente com as métricas utilizadas para as medições de desempenho.

4.1 Métodos de Precisão Numérica Estendida

4.1.1 Método de Rush-Larsen Explícito de 2ª Ordem (RL2)

A partir da família de extensões do método de Rush-Larsen descrita na Seção 3.7 - Equações (3.19), (3.20) e (3.21) - escolhemos $\theta = 1/2$, obtendo:

$$c_{-1} = 1/2 , \quad (4.1)$$

$$c_0 = 1/2 , \quad (4.2)$$

$$c_1 = 0 , \quad (4.3)$$

de modo que

$$\begin{aligned} a_{n+1/2}^j &= (1/2)a_{n+1}^j + (1/2)a_n^j, \\ b_{n+1/2}^j &= (1/2)b_{n+1}^j + (1/2)b_n^j . \end{aligned} \quad (4.4)$$

Desta maneira, produzimos um método implícito, uma vez que o cálculo de $a_{n+1/2}^j$ e $b_{n+1/2}^j$ dependerá dos valores desconhecidos a_{n+1}^j e b_{n+1}^j , respectivamente. Neste trabalho

propomos a utilização de aproximações para os valores de a_{n+1}^j e de b_{n+1}^j com base em uma abordagem semelhante ao método de Heun. O método resultante é explícito e será referenciado neste trabalho por RL2, sendo o mesmo descrito pelos passos abaixo:

1. Dado $\vec{Y}_n$, calcule $\vec{Y}_{n+1}$ pelo método de Rush-Larsen original.
2. Dado $\vec{Y}_{n+1}$, calcule as aproximações $\bar{a}_{n+1}^j$ e $\bar{b}_{n+1}^j$ para a_{n+1}^j e b_{n+1}^j , respectivamente, pela Eq. (3.18). Então calcule
$$a_{n+1/2}^j = \frac{1}{2}\bar{a}_{n+1}^j + \frac{1}{2}a_n^j \text{ e } b_{n+1/2}^j = \frac{1}{2}\bar{b}_{n+1}^j + \frac{1}{2}b_n^j.$$
3. Calcule $\vec{Y}_{hh,n+1}$ usando a Eq. (3.19). Se $a_{n+1/2}^j$ é aproximadamente nulo, então utilize $y_{hh,n+1}^j = y_{hh,n}^j + \frac{h}{2}(f_{hh}^j(\vec{Y}_n, t_n) + f_{hh}^j(\vec{Y}_{n+1}, t_{n+1}))$.
4. Calcule $\vec{Y}_{nl,n+1}$ por $y_{nl,n+1}^j = y_{nl,n}^j + \frac{h}{2}(f_{nl}^j(\vec{Y}_n, t_n) + f_{nl}^j(\vec{Y}_{n+1}, t_{n+1}))$.

4.1.2 Método *SAST2 Simplificado*

Conforme veremos em capítulos posteriores, o método SAST2 - descrito na seção 3.6 - apresenta características de estabilidade consideravelmente superiores aos métodos SAST1, Euler, Heun, RL e RL2, permitindo a utilização de passos de tempo significativamente maiores que tais métodos para os modelos baseados em cadeias de Markov. No entanto, seu custo computacional por passo de tempo é severamente mais elevado que o custo apresentado pelo método SAST1. Com a finalidade de desenvolvermos um esquema adaptativo baseado nos métodos SAST1 e Uniformização, produzimos uma versão de segunda ordem do método SAST1 baseada no método de Heun, de forma exatamente análoga ao método RL2. Damos a tal método o nome de *SAST2 Simplificado*.

O método *SAST2 Simplificado* apresenta um custo por passo de tempo equiparável ao método SAST1. No entanto, de acordo com experimentos numéricos realizados, suas restrições de estabilidade são praticamente idênticas às aquelas apresentadas pelo método SAST1. Embora tal método, quando utilizado isoladamente, apresente um desempenho inferior em relação ao método SAST2 (por questões de estabilidade), seu custo por passo de tempo consideravelmente mais baixo o torna mais indicado para o desenvolvimento de esquemas de passo adaptativo baseados no método SAST1.

4.1.3 Método de Uniformização de 2ª Ordem (UNI2)

O método de Uniformização, assim como o método de Rush-Larsen, baseia-se em um processo de linearização local e na obtenção de uma solução exponencial. Para o método de Rush-Larsen, admitimos que uma equação diferencial ordinária seja linear e de coeficientes constantes para um pequeno intervalo de tempo. Portanto, a solução local aplicada pelo método de Rush-Larsen é a solução exata de uma equação diferencial da forma

$$y' = ay + b ,$$

a qual é dada por

$$y(t) = e^{at} \left(y_0 + \frac{b}{a} \right) - \frac{b}{a} ,$$

onde y_0 é o valor de y para $t = 0$ e os coeficientes a e b são constantes. Se o termo b for nulo, temos que a solução exata da equação resultante será dada por

$$y(t) = y_0 e^{at} . \quad (4.5)$$

No caso do método de Uniformização, a linearização local se dá para um sistema de EDOs da forma

$$\frac{d\vec{Y}_{mk}}{dt} = A(\vec{Y}, t) \vec{Y}_{mk} ,$$

de modo que, se considerarmos $A(\vec{Y}, t)$ constante para um pequeno intervalo de tempo, teremos

$$\frac{d\vec{Y}_{mk}}{dt} = A \vec{Y}_{mk} .$$

Um sistema diferencial matricial da forma $\frac{d\vec{Y}}{dt} = A \vec{Y}$ possui solução exponencial dada por

$$\vec{Y}(t) = \vec{Y}_0 e^{At} .$$

Tal solução assemelha-se àquela apresentada pela Equação (4.5), diferenciando-se pela presença da exponencial matricial, a qual é definida através da série de Taylor (veja a Seção 3.5). Com base em tais semelhanças, podemos propor uma extensão para o método de Uniformização baseada na utilização da matriz de transição avaliada no instante intermediário entre o passo atual e o passo subsequente. Em outras palavras, a cada passo de tempo, efetuamos:

$$\vec{Y}_{mk,n+1} = \vec{Y}_{mk,n} e^{A(t_{n+1/2})h}, \quad (4.6)$$

onde h é o passo de tempo utilizado. Uma vez conhecida a matriz $A(t_{n+1/2})$, podemos completar o passo de uniformização utilizando-se as Equações (3.8), (3.9) e (3.11), substituindo-se $A^*(t_n)$ por $A^*(t_{n+1/2})$, $A(t_n)$ por $A(t_{n+1/2})$ e $q(t_n)$ por $q(t_{n+1/2})$.

De modo semelhante ao método RL2, neste trabalho propomos a utilização da aproximação

$$A(t_{n+1/2}) = \frac{A(t_n) + \bar{A}(t_{n+1})}{2},$$

onde $\bar{A}(t_{n+1})$ é uma aproximação para $A(t_{n+1})$ obtida através de um passo preliminar utilizando o método de uniformização tradicional (1ª ordem). Tal esquema numérico proposto também assemelha-se ao método de Heun e será referenciado neste trabalho como UNI2. Posteriormente mostraremos resultados experimentais que indicam uma convergência numérica de 2ª ordem para este método.

4.2 Métodos Híbridos

Nesta seção descreveremos os métodos híbridos gerados a partir da associação do método de uniformização com outros métodos tradicionais apresentados no capítulo anterior. É importante destacar que o método de uniformização pode ser aplicado somente às Equações do tipo (2.40), de modo que precisamos associá-lo com outro método numérico para que o sistema de equações do modelo seja resolvido por completo.

4.2.1 Método Euler + UNI

O método híbrido Euler + UNI consiste na utilização do método de uniformização para as variáveis de cadeias de Markov ($\vec{Y}_{mk}$), ou seja, aquelas determinadas pelas Equações (2.40), e na aplicação do método de Euler para as demais variáveis do modelo.

4.2.2 Método RL + UNI

O esquema híbrido RL + UNI é bastante similar ao método Euler + UNI, sendo que a única diferença encontra-se na utilização do método de Rush-Larsen em substituição ao método de Euler.

4.2.3 Método SAST1 + UNI

Aplicando-se o método de Uniformização para as variáveis de Markov definidas pela Eq. (2.40) e o método SAST1 para as demais equações do sistema, temos o método híbrido SAST1 + UNI. Tal método possui um elevado custo computacional por passo de tempo mas, conforme será visto adiante, a estabilidade numérica de resolução é aumentada de forma significativa.

4.3 Métodos com Discretização Temporal Particionada (DTP)

Determinados métodos dentre os apresentados em seções anteriores apresentam diferentes características numéricas para conjuntos distintos de equações em modelos cardíacos: o método de Rush-Larsen apresenta condições de estabilidade menos restritivas que o método de Euler para as Equações do tipo Hodgkin-Huxley (2.39) (Perego e Veneziani, 2009); analogamente, o método de Euler apresenta estabilidade numérica consideravelmente inferior ao método de uniformização para a solução das Equações de Markov (2.40) (Melamed e Yadin, 1984).

Com base nessas observações, desenvolvemos alguns experimentos utilizando um passo

de tempo diferenciado para a solução das equações para as quais o método utilizado apresenta melhor estabilidade numérica. Em termos práticos, suponha que para o método híbrido Euler + UNI utilizemos um passo de tempo h_2 para as equações de Markov e um passo de tempo h_1 para as demais equações do sistema, a serem resolvidas com o método de Euler explícito, de modo que $h_2 > h_1$. Geramos *solvers* com tal processo de discretização particionada para determinados métodos apresentados no capítulo anterior, de acordo com a Tabela 4.1. Conforme veremos mais adiante, em alguns casos tal técnica apresentou consideráveis ganhos de desempenho em relação às versões de passo único correspondentes.

<i>Solver</i>	Equações NL (2.38)		Equações HH (2.39)		Equações MK (2.40)	
	Método	Δt	Método	Δt	Método	Δt
Euler + UNI / DTP	Euler	h_1	Euler	h_1	UNI	h_2
RL / DTP	Euler	h_1	RL	h_2	Euler	h_1
RL + UNI / DTP	Euler	h_1	RL	h_2	UNI	h_2
SAST1 / DTP	SAST1	h_1	RL	h_2	SAST1	h_1
SAST1 + UNI / DTP	SAST1	h_1	RL	h_2	UNI	h_2

Tabela 4.1: Esquemas numéricos com discretização particionada ($h_1 < h_2$)

Suponha que estejamos utilizando um método M1 para o conjunto de variáveis $\vec{Y}_{h_1}$, com passo de tempo h_1 e um método M2 para o conjunto das variáveis restantes $\vec{Y}_{h_2}$, utilizando um passo de tempo $h_2 = D \times h_1$, onde D é um número inteiro. A técnica de discretização particionada utilizada neste trabalho é ilustrada pelo Algoritmo 1. Podemos notar que, a cada D passos de tempo realizados para as variáveis $\vec{Y}_{h_1}$, as variáveis $\vec{Y}_{h_2}$ são calculadas com um único passo de tempo $D \times h_1$.

De acordo com o Algoritmo 1, quando n for múltiplo de D , calculamos $\vec{Y}_{h_1, n+1}$ e $\vec{Y}_{h_2, n+D}$. Nos próximos $D-1$ passos $\vec{Y}_{h_1}$ é calculado com passo h_1 e $\vec{Y}_{h_2}$ não é atualizado. Quando $\vec{Y}_{h_2, n+D}$ é calculado, podemos dizer que a solução completa no instante atual

encontra-se sincronizada e seus valores podem ser salvos.

Algoritmo 1: Pseudocódigo ilustrando um método de discretização particionada:

$h_1 = h$ e $h_2 = D \times h$. A operação $a \% b$ denota o resto da divisão do inteiro a pelo inteiro b .

Entrada: t_{final}, h_1, D

início

$t \leftarrow 0$;

$n \leftarrow 0$;

enquanto $t < t_{final}$ **faça**

Calcule $\vec{Y}_{h_1, n+1}$ pelo método M1;

se $n \% D = 0$ **então**

Calcule $\vec{Y}_{h_2, n+D}$ pelo método M2;

senão se $(n + 1) \% D = 0$ **então**

Salve a solução atual (agora $\vec{Y}_{h_1}$ e $\vec{Y}_{h_2}$ estão sincronizados) ;

fim

$t \leftarrow t + h_1$;

$n \leftarrow n + 1$;

fim

fim

Saída: Aproximação numérica de $\vec{Y}(t)$ para $0 < t \leq t_{final}$.

4.4 Métodos Adaptativos

4.4.1 Método de Rush-Larsen Adaptativo

O Método de Rush-Larsen adaptativo (RL / ADP) é um esquema explícito proposto neste trabalho, baseado no método de Rush-Larsen e que adapta a magnitude do passo de tempo durante a execução. Tal abordagem busca reduzir o tempo de computação enquanto mantém os erros numéricos sob controle. Este método é muito similar ao *Euler adaptativo* apresentado na Seção 3.3, porém utiliza os métodos de Rush-Larsen de 1^a e de 2^a ordens. Cada passo do método é composto pelas seguintes operações:

1. Dado $\vec{Y}_n$, calcule $\vec{\bar{Y}}_{n+1}$ e $\vec{Y}_{n+1}$ como estabelecido pelo método RL2.
2. Calcule o máximo erro relativo encontrado entre as duas soluções:

$$erro_{local} = \max_{1 \leq j \leq N_{var}} \left| \frac{y_{n+1}^j - \bar{y}_{n+1}^j}{y_{n+1}^j} \right| \quad (4.7)$$

3. Dada uma tolerância de erro local τ , se $erro_{local} \leq \tau$, então o passo de tempo será atualizado e dado por $h_{new} = h \sqrt{\frac{0.5 \times \tau}{erro_{local}}}$, respeitando um valor máximo (h_{max}) e um valor mínimo (h_{min}) predefinidos. Se $erro_{local} > \tau$, então o passo atual é descartado e calculado novamente com o passo de tempo reduzido à metade.
4. A solução discreta considerada para o próximo passo será $\vec{\bar{Y}}_{n+1}$.

4.4.2 Métodos Híbridos de Passo Adaptativo

Nesta seção descreveremos os métodos híbridos baseados no método de Uniformização com passo de tempo adaptativo. Tais métodos utilizam a extensão do método de Uniformização para segunda ordem de precisão (UNI2), de modo a obtermos uma estimativa de erro local para o método de primeira ordem a cada passo de tempo executado. Suponha que estejamos utilizando um método explícito de 1ª ordem denominado M1 e sua extensão de 2ª ordem denominada M2, juntamente com o método de Uniformização, para gerar um esquema de passo adaptativo. O Algoritmo 2 descreve de forma generalizada como implementamos neste trabalho esquemas de passo adaptativo para um método M1 a ser

utilizado juntamente com o método de uniformização.

Algoritmo 2: Algoritmo genérico para os métodos híbridos de passo adaptativo. A

operação $a\%b$ denota o resto da divisão do inteiro a pelo inteiro b .

Entrada: t_{final} , $h_{inicial}$, τ

início

$t \leftarrow 0$;

$h \leftarrow h_{inicial}$;

$n \leftarrow 0$;

repita

repita

Calcule $\vec{Y}_{nl,n+1}$ e $\vec{Y}_{hh,n+1}$ pelo método M1;

Calcule $\vec{Y}_{mk,n+1}$ pelo método UNI1;

Calcule $\vec{Y}_{nl,n+1}$ e $\vec{Y}_{hh,n+1}$ pelo método M2;

Calcule $\vec{Y}_{mk,n+1}$ pelo método UNI2;

$erro_{local} = \max_{1 \leq j \leq N_{var}} \left| \frac{y_{n+1}^j - \bar{y}_{n+1}^j}{y_{n+1}^j} \right|$;

se $erro_{local} > \tau$ **então**

$h \leftarrow h/2$;

fim

até $erro_{local} \leq \tau$;

$h = h \sqrt{\frac{0.5 \times \tau}{erro_{local}}}$;

$t \leftarrow t + h$;

$n \leftarrow n + 1$;

até $t \geq t_{final}$;

fim

Saída: Aproximação numérica de $\vec{Y}(t)$ para $0 < t \leq t_{final}$.

A expressão para o novo passo de tempo apresentada no Algoritmo 2 é a mesma utilizada para o método RLTA e para o método de Euler Adaptativo (Campos *et al.*, 2013). Caso o erro local calculado seja superior à tolerância τ predefinida, descartamos os valores recém calculados para as variáveis do modelo e efetuamos novamente o cálculo

do passo corrente, utilizando-se a metade do passo de tempo original. Tal procedimento trata-se de uma heurística que busca evitar uma nova reiteração de passo.

Os métodos híbridos de passo adaptativo que implementamos neste trabalho são os descritos a seguir:

- Euler + UNI de Tempo Adaptativo (Euler + UNI / ADP): Método ilustrado pelo Algoritmo 2, com o método M1 sendo substituído pelo método de Euler explícito e M2 pelo método RK2. Este método assemelha-se ao método de Euler adaptativo, com a diferença de que o método de Uniformização é utilizado para as equações de Cadeias de Markov.
- RL + UNI de Tempo Adaptativo (RL + UNI / ADP): Método análogo ao anterior, porém com os métodos M1 e M2 sendo substituídos pelos métodos RL1 e RL2, respectivamente.
- SAST1 + UNI de Tempo Adaptativo (SAST1 + UNI / ADP): Método representado pelo Algoritmo 2 - como os dois anteriores -, com a utilização do método SAST1 no lugar de M1. Em substituição a M2 utilizamos o método SAST2 Simplificado (apresentado na seção 4.1.2).

4.5 Otimizações para o Método de Uniformização

Nesta seção iremos discutir duas otimizações importantes que foram aplicadas na implementação do método de uniformização utilizada neste trabalho. A primeira delas diz respeito ao *truncamento bilateral* efetuado sobre a série da formulação do método de uniformização, que permite a aproximação controlada da série pela utilização de apenas uma faixa de termos centrais. A segunda otimização que iremos abordar trata-se da utilização de estruturas esparsas para representar as matrizes de transição das cadeias de Markov envolvidas.

4.5.1 Truncamento da Série

Conforme descrito na Seção 3.5, a expressão (3.11) pode ser utilizada para a realização do truncamento *direito* sobre o somatório da Equação (3.8). Com isso, é possível dispensar o cálculo dos termos tais que $i > N(t_n)$, cuja magnitude é considerada desprezível. Além disso, é possível aplicar um truncamento sobre os termos *à esquerda* (termos de índices menores), uma vez que, para magnitudes suficientemente grandes de $q(t)h$, a distribuição dos termos de magnitude significativa passa a concentrar-se em posições intermediárias do somatório presente na Eq. (3.8), conforme descrito em Reibman e Trivedi (1988). Em outras palavras, os termos mais à esquerda - assim como os termos de índice grande - passam a contribuir de forma pouco significativa para o cômputo final do somatório. Supondo que, a cada passo n , tenhamos meios justificáveis para escolher índices $N_1(t_n)$ e $N_2(t_n) = N(t_n)$ tais que os únicos termos calculados para o somatório da Equação (3.8) sejam aqueles associados aos índices $\{N_1(t_n), N_1(t_n) + 1, \dots, N_2(t_n)\}$, temos a seguinte versão simplificada para o método de uniformização:

$$\vec{Y}_{mk,n+1} = e^{-q(t_n)h} \left[\sum_{i=N_1(t_n)}^{N_2(t_n)} \frac{(A^*(t_n)q(t_n)h)^i}{i!} \right] \vec{Y}_{mk,n} \quad (4.8)$$

$$A^*(t_n) = \frac{A(t_n)}{q(t_n)} + I \quad (4.9)$$

$$q(t_n) \geq \max_{1 \leq i \leq N_{mk}} |[A(t_n)]_{ii}| \quad (4.10)$$

Para determinarmos o valor de $N_1(t_n)$, utilizamos a abordagem descrita em Reibman e Trivedi (1988), que consiste na determinação do maior inteiro $N_1(t_n)$ tal que

$$1 - e^{-q(t_n)h} \sum_{i=0}^{N_1(t_n)-1} \frac{(q(t_n)h)^i}{i!} \leq \lambda/2. \quad (4.11)$$

Para o cálculo de $N_2(t_n)$ utilizamos a expressão do limitante de erro por truncamento para a segunda metade da tolerância λ predefinida, utilizando o menor valor inteiro $N_2(t_n)$ tal que

$$1 - e^{-q(t_n)h} \sum_{i=0}^{N_2(t_n)} \frac{(q(t_n)h)^i}{i!} \leq \lambda/2 . \quad (4.12)$$

4.5.2 Matrizes Esparsas

As matrizes de transição provenientes dos modelos de Markov geralmente apresentam grande esparsidade, ou seja, muitos de seus elementos são valores nulos. Do ponto de vista das cadeias de Markov, um elemento nulo na matriz de transição equivale à impossibilidade de transição direta entre dois determinados estados da cadeia de Markov em questão. O método de uniformização lida diretamente com as matrizes de transição das cadeias de Markov, de modo que tal característica de esparsidade pode ser explorada. Embora tais matrizes possuam, geralmente, dimensões relativamente pequenas (entre 4x4 e 11x11), é possível obter alguma economia de memória com a utilização de matrizes esparsas.

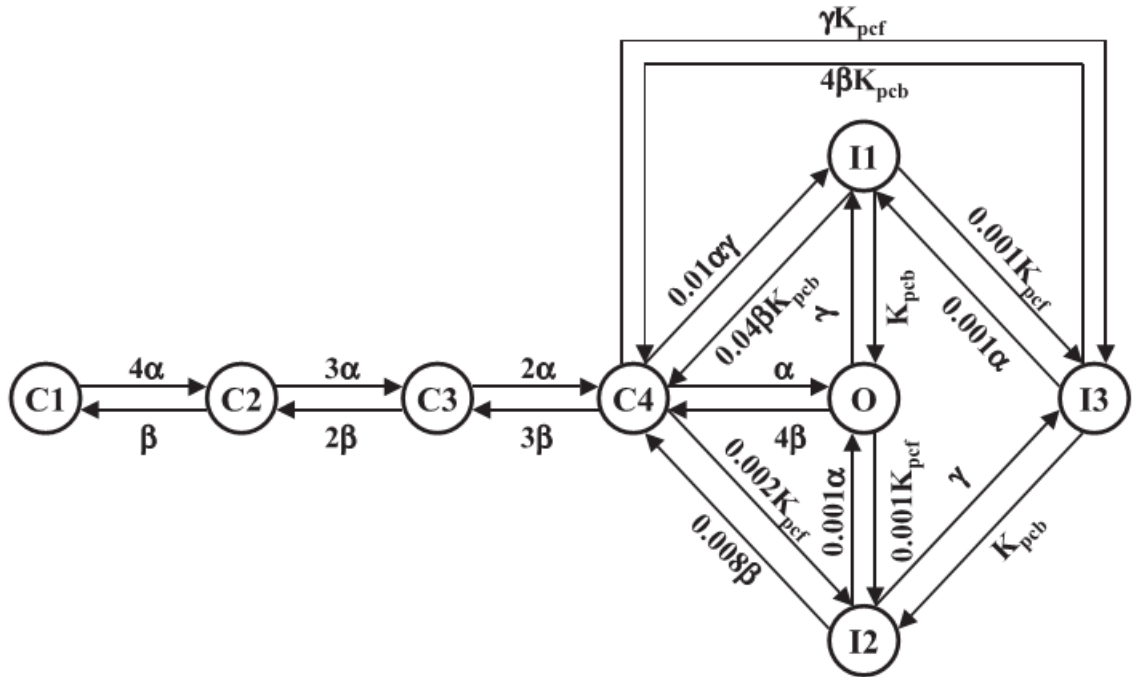


Figura 4.1: Cadeia de Markov para a corrente de Ca^{2+} tipo L do modelo de Bondarenko *et al.* (Imagem extraída de Bondarenko *et al.* (2004)).

A título de ilustração, considere a cadeia de Markov para os estados do canal iônico de Ca^{2+} tipo L do modelo de Bondarenko *et al.* (2004), mostrada na Figura 4.1. Tal cadeia

de Markov é constituída de 8 estados e sua matriz de transição associada tem dimensão 8×8 e é dada por:

$$\mathbf{A} = \begin{bmatrix} d_1 & \beta & 0 & 0 & 0 & 0 & 0 & 0 \\ 4\alpha & d_2 & 2\beta & 0 & 0 & 0 & 0 & 0 \\ 0 & 3\alpha & d_3 & 3\beta & 0 & 0 & 0 & 0 \\ 0 & 0 & 2\alpha & d_4 & 0.04\beta k_{pcb} & 0.008\beta & 4\beta k_{pcb} & 4\beta \\ 0 & 0 & 0 & 0.01\alpha\gamma & d_5 & 0 & 0.001\alpha & \gamma \\ 0 & 0 & 0 & 0.002k_{pcb} & 0 & d_6 & k_{pcb} & 0.001k_{pcb} \\ 0 & 0 & 0 & \gamma k_{pcb} & 0.001k_{pcb} & \gamma & d_7 & 0 \\ 0 & 0 & 0 & \alpha & k_{pcb} & 0.001\alpha & 0 & d_8 \end{bmatrix},$$

onde os elementos diagonais d_j são dados por

$$d_j = - \sum_{i=1, i \neq j}^8 [A]_{ij}.$$

Podemos notar que a matriz A possui apenas 30 elementos não-nulos dentre seus 64 elementos. Idealmente, poderíamos armazenar A utilizando-se cerca de metade da memória utilizada para armazenar a matriz completa. No entanto, algum espaço extra é necessário para que a estrutura original possa ser recuperada a partir de sua representação.

Como as estruturas das cadeias de Markov são conhecidas previamente a partir dos modelos, podemos criar matrizes esparsas estáticas, ou seja, estruturas em que os elementos nulos não irão alterar-se durante o seu tempo de vida. Deste modo, é possível dispensar a utilização de listas encadeadas - pois a quantidade de memória utilizada é conhecida no momento de criação do objeto -, o que permite a estrutura esparsa criada não apresentar prejuízos de desempenho na realização de algumas operações.

A estrutura utilizada para as matrizes esparsas neste trabalho é composta de um vetor de N_{mk} ponteiros, em que N_{mk} é o número de estados da cadeia de Markov em questão. Cada ponteiro corresponde a uma linha da matriz e aponta para uma estrutura contendo um vetor de valores reais (ponto flutuante) e um vetor de inteiros. O vetor de valores

reais apresenta em sequência os valores não nulos da linha associada, enquanto o vetor de inteiros especifica os índices de coluna na matriz original para cada valor armazenado no vetor de reais. Além disso, tal vetor de inteiros tem como primeiro elemento o número de elementos não nulos presentes na linha correspondente. Deste modo, se a mesma quantidade de bits for utilizada para representar valores inteiros e de ponto flutuante, teríamos que a quantidade de memória utilizada por essa estrutura seria da ordem de duas vezes o tamanho de memória necessário para armazenar os valores não nulos da matriz original.

Para as estruturas implementadas neste trabalho, utilizamos o tipo *double* da linguagem C++ para representar os valores reais não nulos da matriz original e o tipo *short int* para representar os índices de coluna. Uma variável do tipo *double* ocupa 64 bits em memória, enquanto variáveis do tipo *short int*, sob determinadas condições, ocupam 16 bits. Dessa forma, para a matriz A apresentada anteriormente, em teoria - desconsiderando a memória destinada aos ponteiros - utilizaríamos $(16 + 64)bits \times 30 + 8 \times 16bits = 316bytes$, contra $64bits \times 64 = 512bytes$ para o armazenamento completo da matriz, o que representa uma economia teórica de 38% em utilização de memória.



Figura 4.2: Cadeia de Markov para a corrente de I_{Kr} do modelo de Bondarenko *et al.* (Imagem extraída de Bondarenko *et al.* (2004)).

Para a cadeia de Markov referente à corrente I_{kr} do modelo de Bondarenko *et al.* (2004), mostrada na Figura 4.2, a economia de memória realizada com utilização da estrutura esparsa descrita é de aproximadamente 30%. Para a cadeia de Markov de 9 estados da Figura 2.11, a redução na utilização de memória seria cerca de 49%.

4.6 Implementação dos Métodos Adaptativos

Os métodos adaptativos Euler / ADP e RL / ADP apresentam custo por passo de tempo muito próximo àquele desempenhado pelas respectivas versões de passo fixo. Isso ocorre porque, para ambos os métodos, a cada passo faz-se necessária apenas uma avaliação de lado direito, uma vez que o cálculo de $\vec{F}(\vec{Y}_{n+1}, t_n + h)$ para a obtenção da solução local de 2ª ordem é reaproveitado no passo seguinte para o avanço de 1ª ordem.

No caso dos métodos híbridos adaptativos, também ocorre o reaproveitamento do lado direito e das matrizes de transição entre passos consecutivos. No entanto, o cômputo do somatório da Equação(4.8) faz-se necessário duas vezes a cada passo realizado. Para mostrarmos tal fato com maior clareza, descrevemos abaixo as etapas realizadas a cada passo por parte do método Euler + UNI / ADP (os métodos RL + UNI / ADP e SAST1 + UNI / ADP podem ser descritos de maneira análoga). Por razões de simplificação, suponha que o modelo em questão seja composto de apenas uma Cadeia de Markov com matriz de transições $A(\vec{Y}, t)$:

1. $\vec{F}_{nl}(\vec{Y}_n, t_n)$, $\vec{F}_{hh}(\vec{Y}_n, t_n)$ e $A(\vec{Y}_n, t_n)$ são conhecidos a partir do passo anterior.
2. Calculamos

$$\vec{Y}_{nl,n+1} = \vec{Y}_{nl,n} + h\vec{F}_{nl}(\vec{Y}_n, t_n)$$

e

$$\vec{Y}_{hh,n+1} = \vec{Y}_{hh,n} + h\vec{F}_{hh}(\vec{Y}_n, t_n) .$$

3. $\vec{Y}_{mk,n+1}$ é calculado pelo método de uniformização. A título de simplificação, escrevemos

$$\vec{Y}_{mk,n+1} = \vec{Y}_{mk,n} e^{hA(\vec{Y}_n, t_n)} .$$

4. Calculamos $\vec{F}_{nl}(\vec{Y}_{n+1}, t_{n+1})$, $\vec{F}_{hh}(\vec{Y}_{n+1}, t_{n+1})$ e $A(\vec{Y}_{n+1}, t_{n+1})$.
5. Calculamos

$$\vec{Y}_{nl,n+1} = \vec{Y}_{nl,n} + \frac{h}{2}[\vec{F}_{nl}(\vec{Y}_n, t_n) + \vec{F}_{nl}(\vec{Y}_{n+1}, t_{n+1})]$$

e

$$\vec{\bar{Y}}_{hh,n+1} = \vec{Y}_{hh,n} + \frac{h}{2} [\vec{F}_{hh}(\vec{Y}_n, t_n) + \vec{F}_{hh}(\vec{Y}_{n+1}, t_{n+1})]$$

6. Processamos a matriz $\bar{A}_{n+1/2} = \frac{1}{2}[A(\vec{Y}_n, t_n) + A(\vec{Y}_{n+1}, t_{n+1})]$ e efetuamos, através do método de uniformização,

$$\vec{\bar{Y}}_{mk,n+1} = \vec{Y}_{mk,n} e^{h\bar{A}_{n+1/2}} .$$

7. Com as soluções $\vec{Y}_{n+1}$ e $\vec{\bar{Y}}_{n+1}$ calculadas, fazemos as operações de adaptação do passo h conforme descrito no Algoritmo 2.

O cômputo do método de uniformização apresentado no item 6 não pode ser reaproveitado para o passo seguinte, uma vez que as sucessivas multiplicações matriz-vetor se dão através de uma matriz normalizada gerada a partir da média aritmética entre as matrizes $A(\vec{Y}_n, t_n)$ e $A(\vec{Y}_{n+1}, t_{n+1})$.

No entanto, algumas otimizações são possíveis: o valor de $q(t_{n+1})$ obtido para a geração da Matriz $A^*(\vec{Y}_{n+1}, t_{n+1})$ pode ser computado no item 6, e o valor de q utilizado para o cômputo de $\vec{\bar{Y}}_{mk,n+1}$ pode ser adotado como o máximo valor entre $q(t_{n+1})$ e $q(t_n)$ - veja a Equação (4.10) -, dispensando o seu cálculo a partir da matriz média $\bar{A}_{n+1/2}$. Deste modo, o cômputo da taxa q pode ser realizado uma única vez por passo de tempo.

4.7 Experimentos Computacionais

4.7.1 Modelos Utilizados

Com o objetivo de validar e de analisar o desempenho dos métodos apresentados nos capítulos 3 e 4, aplicamos os mesmos para a realização de simulações numéricas a partir de quatro modelos da eletrofisiologia de miócitos cardíacos. A seguir apresentamos uma breve descrição para cada um deles:

1. Bondarenko *et al.* (2004) introduz um modelo que visa descrever os miócitos ventriculares de camundongos. Este modelo é composto de 45 variáveis de estado,

sendo 26 delas associadas a 4 diferentes cadeias de Markov de 4 a 9 estados, 8 são variáveis do tipo HH e as 11 restantes são variáveis totalmente não-lineares.

2. Winslow *et al.* (1999) apresenta um modelo para o miócito ventricular canino, composto de 33 equações diferenciais ordinárias, incluindo 9 do tipo NL, 7 variáveis de formulação Hodgkin-Huxley (tipo HH) e duas cadeias de Markov contendo de 4 até 12 variáveis de estado.
3. Iyer *et al.* (2004) propôs um modelo bastante complexo para a descrição de miócitos do ventrículo esquerdo humano, composto por 67 EDOs. Dentre elas, o modelo apresenta 57 variáveis na forma de 7 cadeias de Markov, cuja extensão varia entre 4 e 13 estados. O modelo também possui 9 variáveis do tipo NL e apenas uma do tipo HH.
4. Finalmente, o quarto modelo utilizado foi proposto por Tusscher e Panfilov (2006). Ele descreve o comportamento elétrico de miócitos do ventrículo esquerdo humano, sendo composto por 19 equações diferenciais, dentre as quais 12 são do tipo *quasi-linear* e as restantes são totalmente não-lineares. Portanto, tal modelo não apresenta cadeias de Markov.

4.7.2 Protocolos de Estímulo Elétrico

Outra informação importante acerca dos experimentos realizados refere-se ao *protocolo de estímulo* utilizado nas simulações dos modelos apresentados na seção anterior. O protocolo de estímulo é definido por certas informações sobre corrente de estímulo I_{stim} aplicada à membrana sarcoplasmática: (i) a amplitude da corrente $I_{stim,max}$; (ii) a duração do pulso; (iii) a frequência de estímulo (também denominada *paceing rate*).

A amplitude e a duração do pulso de estímulo foram fixadas em, respectivamente: $80pA/pF$ e $0.5ms$ para o modelo de Bondarenko *et al.*; $21.13\mu A/\mu F$ e $2ms$ para o modelo de Winslow *et al.*; $15\mu A/\mu F$ e $3ms$ para o modelo de Iyer *et al.*; $52pA/pF$ e $1ms$ para o modelo de Tusscher-Panfilov. Tais valores foram extraídos dos artigos originais referentes aos modelos utilizados.

A frequência de estímulo para os modelos de Iyer *et al.*, Winslow *et al.* e Tusscher-Panfilov foi fixada em 1 Hz. Para o modelo de Bondarenko *et al.* utilizamos a frequência de 14 Hz, uma vez que a duração do PA em camundongos é consideravelmente menor.

4.7.3 Ferramentas Computacionais

Para cada modelo descrito na Seção 4.7.1, obtemos os códigos em C++ para os lados direitos das EDOs através da plataforma AGOS (Campos *et al.*, 2010), que é capaz de produzir *solvers* em C++ a partir de arquivos no padrão CellML (Cuellar *et al.*, 2003). O projeto *CellML.org* reúne diversos modelos matemáticos para a fisiologia celular em um formato padronizado, que pode ser interpretado por diversas ferramentas de simulação. Deste modo, todos os modelos utilizados neste trabalho puderam ser encontrados no portal CellML.org e processados via AGOS para a obtenção de equações e parâmetros no formato de código C++. Todos os métodos numéricos apresentados neste trabalho foram programados em C++ orientado a objetos, de tal maneira que sua implementação fosse independente do modelo matemático a ser simulado.

Para a medir a quantidade de memória alocada por parte dos *solvers* implementados, utilizamos a ferramenta *Valgrind*, que trata-se de um *profiler* capaz de fornecer informações detalhadas sobre a execução de aplicações C / C++.

4.7.4 Ambiente Computacional

Todas as simulações executadas neste trabalho foram realizadas em uma máquina com processador AMD FX-8350 (8 núcleos, 4.0 GHz) e 8GB de memória RAM a 1866 MHz em *dual channel*. O sistema operacional utilizado foi o Linux Debian, versão 7.7.

4.8 Métricas

Os métodos implementados neste trabalho serão comparados em termos de utilização de memória, de tempo de execução, de passo de tempo (maiores valores indicam melhor estabilidade) e de erro numérico. A seguir descreveremos as métricas adotadas para a

realização de medidas e estipulação de parâmetros.

4.8.1 Erros Numéricos e Tolerâncias

Os valores de erros numéricos apresentados neste trabalho são calculados pela Equação (4.13), considerando os valores discretizados do potencial transmembrânico v_k :

$$erro = \frac{\sqrt{\sum (v_k^{ref} - v_k)^2}}{\sqrt{\sum (v_k^{ref})^2}}, \quad (4.13)$$

onde v_k é a discretização de $V(kh)$ relativa ao método avaliado, v_k^{ref} representa valores de referência adotados para $V(kh)$ e h é o passo de tempo. Como soluções de referência, consideramos os resultados obtidos pelo método de Runge-Kutta clássico (de quarta ordem) (Holmes, 2007) utilizando o passo de tempo de 10^{-5} ms. Com relação à tolerância do método de Uniformização λ , definida pela Equação (3.11), nas situações em que sua magnitude for omitida neste texto, o valor utilizado foi 10^{-6} .

4.8.2 Tempos de Computação

Os tempos de computação apresentados neste trabalho foram calculados a partir da utilização do programa *time* do sistema operacional Linux. Cada valor de tempo de computação apresentado representa uma média aritmética obtida de 10 execuções do *solver* correspondente a partir de parâmetros idênticos. Os valores de desvio padrão calculados para tais amostragens foram, consistentemente, inferiores a 5%.

4.8.3 Utilização de Memória

Nesta seção descreveremos as métricas utilizadas neste trabalho acerca dos experimentos relacionados à utilização de memória por parte do método de Uniformização quando aplicado na resolução de modelos da eletrofisiologia cardíaca. Para realizar medições sobre a utilização de memória em tempo de execução, utilizamos o pacote *Valgrind* e sua

ferramenta interna *Massif*, os quais permitem o rastreamento das alocações de memória realizadas durante a execução de um programa C / C++.

Pode-se dizer que um processo (instância de uma aplicação em um sistema operacional) apresenta, basicamente, dois distintos espaços de memória: a Pilha (ou *Stack*) e o *Heap*. A pilha é um trecho reservado de memória que impõe um modelo específico de alocação sequencial e contígua, sendo geralmente alocada no início da aplicação e utilizada para o empilhamento de dados durante a execução de blocos de código. São alocadas na pilha as variáveis estáticas e os parâmetros de função.

O *Heap* não restringe os alocamentos de memória a um padrão específico. Além disso, o *Heap* é considerado dinâmico, uma vez que espaços de memória podem ser criados e desalocados de forma livre ao longo de sua extensão. Por isso, o *Heap* é utilizado para a alocação dinâmica de variáveis. Neste trabalho, vetores, matrizes e estruturas de dados foram todos alocados dinamicamente, de modo que a maior parte dos dados processados pelos métodos implementados são armazenados no *Heap* durante a execução. Além disso, parâmetros passados para funções implementadas em todos os *solvers* consistem em valores simples ou em ponteiros para estruturas armazenadas no *Heap*. Com base nestes fatos, as análises pertinentes à utilização de memória por parte do método de Uniformização foram baseadas unicamente nos espaços de memória alocados no *Heap*.

4.8.4 Passos de Tempo

Embora seja desejável a utilização de passos de tempo h de magnitudes razoáveis, estipulamos um limite máximo para o seu valor com a finalidade de evitarmos situações em que a corrente de estímulo não produza um potencial de ação corretamente. Tais situações podem ocorrer se o passo de tempo utilizado for grande o suficiente para que parte significativa do tempo de aplicação da corrente de estímulo não seja computado. Deste modo, considerando-se a ordem de grandeza das larguras de pulso de estímulo adotadas para os modelos, estabelecemos que o valor máximo para h em todas as simulações realizadas foi de 0.5ms.

No capítulo subsequente iremos apresentar resultados de simulações numéricas

associados a tolerâncias de erro numérico. Em tais casos, com o objetivo de maximizar o desempenho computacional, o passo de tempo utilizado (para cada método numérico) foi gradativamente aumentado até que o erro numérico - calculado pela Equação (4.13) - atingisse a tolerância estabelecida. Deste modo, os passos de tempo h apresentados em tabelas da Seção 5.3 referem-se aos maiores valores tais que o erro numérico correspondente não ultrapasse a tolerância estabelecida.

4.8.5 *Análise de Rigidez*

Com o objetivo de avaliar a estabilidade dos métodos numéricos implementados, realizamos uma análise de rigidez sobre os modelos utilizados a partir de um estudo baseado em autovalores, conforme descrito em Marsh *et al.* (2012).

Para cada passo de simulação podemos realizar a seguinte diagonalização :

$$J_n = V D V^{-1} , \quad (4.14)$$

onde J_n é a matriz jacobiana do sistema no instante t_n (a qual consideraremos ser diagonalizável), V é a matriz cujas colunas são autovetores associados aos autovalores de J_n e D é uma matriz diagonal cujos termos diagonais são os autovalores de J_n . A partir do sistema de EDOs linearizado em torno de t_n

$$\vec{Y}' = J_n \vec{Y} ,$$

temos:

$$\begin{aligned} \vec{Y}' &= V D V^{-1} \vec{Y} \Rightarrow \\ V^{-1} \vec{Y}' &= V^{-1} V D V^{-1} \vec{Y} . \end{aligned}$$

Fazendo-se $\vec{Z} = V^{-1} \vec{Y}$, obtemos o seguinte sistema diagonalizado:

$$\vec{Z}' = D \vec{Z} ,$$

onde cada componente z^i de $\vec{Z}$ está associada ao autovalor $\lambda_i = [D]_{ii}$, uma vez que $\frac{dz^i}{dt} = \lambda_i z^i$.

Como $\vec{Y} = V \vec{Z}$, a variável y^i pode ser escrita da seguinte maneira:

$$y^i = v_1^i z^1 + v_2^i z^2 + \dots + v_{N_{var}}^i z^{N_{var}} , \quad (4.15)$$

onde v_j^i é a i -ésima componente do autovetor $\vec{V}_j$ associado ao autovalor λ_j . Para as análises realizadas no capítulo subsequente, diremos que o autovalor λ_j é associado à variável y^i se o coeficiente v_j^i de z^j na Equação (4.15) tiver magnitude superior em relação aos demais coeficientes da combinação linear. Para realizarmos a detecção de variáveis rígidas de um determinado modelo, tomamos os autovalores de parte real mais negativa dentre os obtidos durante a simulação de um PA e verificamos as variáveis associadas (Marsh *et al.*, 2012).

5 Resultados e Discussão

Neste capítulo apresentaremos resultados obtidos a partir de simulações computacionais dos modelos descritos na seção 4.7.1, utilizando os métodos numéricos apresentados nos Capítulos 3 e 4. Apresentaremos resultados acerca da convergência dos métodos RL2 e UNI2; da utilização de memória por parte do método de uniformização e do desempenho computacional obtido pelos esquemas numéricos propostos em relação a métodos tradicionalmente utilizados para a solução numérica de modelos da eletrofisiologia de células cardíacas.

5.1 Resultados de Convergência

Os resultados apresentados na Tabela 5.1 sugerem, experimentalmente, que a abordagem adotada para o método RL2 (Seção 4.1.1) também apresenta precisão numérica de segunda ordem. Tais resultados consistem de erros numéricos obtidos de simulações para o modelo proposto por Tusscher e Panfilov (2006). A forma com que os erros numéricos são calculados neste trabalho foram apresentadas na Seção 4.8. Na Tabela 5.1, p é a taxa de convergência do método, considerando-se que o erro numérico seja a h^p , onde h é o passo de tempo utilizado. As simulações foram realizadas com o tempo t variando de 0 até 10ms, utilizando-se o potencial transmembrânico inicial $V(0) = -40\text{mV}$ e as demais condições iniciais provenientes de Tusscher e Panfilov (2006).

h (ms)	Erro	p
$h = 3.20 \times 10^{-2}$	6.64×10^{-3}	-
$h = 1.60 \times 10^{-2}$	1.96×10^{-3}	1.76
$h = 8.00 \times 10^{-3}$	5.63×10^{-4}	1.80
$h = 4.00 \times 10^{-3}$	1.47×10^{-4}	1.94
$h = 2.00 \times 10^{-3}$	3.82×10^{-5}	1.94
$h = 1.00 \times 10^{-3}$	9.67×10^{-6}	1.98

Tabela 5.1: Taxas de convergência para o método RL2 obtidas com o modelo proposto por Tusscher e Panfilov (2006).

Apresentamos na Tabela 5.2 indícios experimentais de que o método UNI2 (Seção

4.1.3) apresenta convergência numérica de 2ª ordem, utilizando-se o modelo de Bondarenko *et al.* (2004), com o método UNI2 para as equações do tipo MK e o método RK2 para as demais equações. Nesta tabela, h é o passo de tempo utilizado e p representa a convergência numérica (assumindo que o erro numérico seja proporcional a h^p), calculada a partir da comparação de erro e h entre linhas adjacentes. O modelo foi simulado de $t = 0$ até $t = 10\text{ms}$. Os erros numéricos foram calculados conforme descrito na Seção 4.8.

h (ms)	Erro	p
$h = 8.00 \times 10^{-4}$	2.20×10^{-3}	-
$h = 4.00 \times 10^{-4}$	6.35×10^{-4}	1.79
$h = 2.00 \times 10^{-4}$	1.76×10^{-4}	1.85
$h = 1.00 \times 10^{-4}$	4.48×10^{-5}	1.97
$h = 5.00 \times 10^{-5}$	1.13×10^{-5}	1.99

Tabela 5.2: Taxas de convergência para o método UNI2 obtidas com o modelo proposto por Bondarenko *et al.* (2004).

5.2 Utilização de Memória

A partir da utilização da ferramenta *Valgrind* citada anteriormente, medimos a utilização de memória dos métodos Euler, Euler + UNI, RL, RL + UNI, SAST1 e SAST1 + UNI durante a simulação de um potencial de ação, para cada modelo baseado em cadeias de Markov dentre os apresentados na Seção 4.7.1. Em outras palavras, simulamos 80ms de atividade elétrica para o modelo de Bondarenko *et al.*, e de $t = 0$ até $t = 1\text{s}$ para os modelos de Iyer *et al.* e Winslow *et al.* O passo de tempo h utilizado para todos os pares modelo / método foi de 10^{-4}ms .

Mostramos na Tabela 5.3 as quantidades em *bytes*, relativas aos picos de utilização de memória durante as simulações, para cada par *solver* / modelo citado. Podemos notar que a adição do método de uniformização implica, sistematicamente, em um aumento na quantidade de memória alocada por parte da aplicação. Isso se deve à necessidade de alocação das matrizes de transição das Cadeias de Markov e de vetores auxiliares necessários para o cálculo do somatório vetorial dado pela Equação (3.8).

Na Tabela 5.4 mostramos o percentual de aumento na utilização de memória em

	Bondarenko <i>et al.</i>	Winslow <i>et al.</i>	Iyer <i>et al.</i>
Método	Máx. Heap	Máx. Heap	Máx. Heap
Euler	3624 bytes	3336 bytes	6520 bytes
Euler + UNI	5832 bytes	4448 bytes	11151 bytes
RL	3768 bytes	3480 bytes	6568 bytes
RL + UNI	5952 bytes	4592 bytes	11203 bytes
SAST1	3776 bytes	3488 bytes	6576 bytes
SAST1 + UNI	5960 bytes	4600 bytes	11211 bytes

Tabela 5.3: Picos de memória alocada no *Heap* durante simulações de modelos da eletrofisiologia cardíaca.

consequência da adição da técnica de uniformização aos métodos de Euler, de Rush-Larsen e de SAST1. Podemos notar que o percentual é maior para os modelos em que a proporção de equações de Markov é maior, o que é esperado devido aos seguintes fatos: (i) um maior número de cadeias de Markov implica em um maior número de matrizes de transição; (ii) uma cadeia de Markov maior implica em uma matriz de transição associada maior. Chamamos tal percentual de aumento na utilização de memória de ***Overhead de memória do método de uniformização***, o qual é dado pela expressão

$$Overhead_{UNI} = \frac{\max\{Q_{UNI}\}}{\max\{Q\}},$$

onde Q_{UNI} é a quantidade de memória alocada no *Heap* para o método híbrido em questão e Q é a quantidade de memória alocada para o método não-híbrido associado. A informação sobre o *overhead* de memória associado ao método de uniformização é importante especialmente na avaliação do consumo de memória em simulações de tecidos cardíacos. Simulações tridimensionais do órgão completo podem conter um número de nós da ordem de 10^6 . Em situações como essa a utilização do método de Euler implicaria, por exemplo, para o modelo de Iyer *et al.*, um custo de memória de aproximadamente 6.1 GB para a resolução do modelo celular em cada um dos nós. Utilizando-se o método SAST1 + UNI tal custo seria de 10.4 GB.

	Bondarenko <i>et al.</i>	Winslow <i>et al.</i>	Iyer <i>et al.</i>
<i>Híbrido / Tradicional</i>	$\text{Overhead}_{\text{UNI}}$	$\text{Overhead}_{\text{UNI}}$	$\text{Overhead}_{\text{UNI}}$
Euler + UNI / Euler	1.61	1.33	1.71
RL + UNI / RL	1.58	1.32	1.71
SAST1 + UNI / SAST1	1.58	1.32	1.70

Tabela 5.4: *Overheads* de utilização de memória por parte do método de uniformização.

5.3 Tempos de Computação

Nesta seção apresentaremos os resultados pertinentes aos tempos de computação obtidos para os métodos apresentados nos Capítulos 3 e 4, aplicados aos modelos especificados na Seção 4.7.1. Iremos dividir a apresentação dos resultados em três diferentes categorias: Métodos com passo de tempo único e constante; métodos com discretização particionada e métodos com passo de tempo adaptativo. Posteriormente, iremos realizar uma discussão acerca dos resultados apresentados para as três categorias.

5.3.1 Métodos com Discretização Temporal Uniforme

	Tolerância de 0.2%		Tolerância de 1%		Tolerância de 5%	
Método	h(ms)	TC(s)	h(ms)	TC(s)	h(ms)	TC(s)
Euler	1.97×10^{-4}	11.91	1.97×10^{-4}	11.91	1.97×10^{-4}	11.91
RL	1.97×10^{-4}	12.52	1.97×10^{-4}	12.52	1.97×10^{-4}	12.52
Euler + UNI	6.34×10^{-4}	7.65	1.02×10^{-3}	4.79	1.02×10^{-3}	4.79
RL + UNI	6.34×10^{-4}	7.96	1.02×10^{-3}	4.98	1.02×10^{-3}	4.98
SAST1	4.22×10^{-4}	20.52	1.96×10^{-3}	4.39	6.99×10^{-3}	1.25
SAST1 + UNI	2.53×10^{-3}	3.12	2.01×10^{-2}	0.43	7.24×10^{-2}	0.14
RK2	1.97×10^{-4}	22.61	1.97×10^{-4}	22.61	1.97×10^{-4}	22.61
RL2	1.97×10^{-4}	25.41	1.97×10^{-4}	25.41	1.97×10^{-4}	25.41
SAST2	1.10×10^{-2}	1.04	1.36×10^{-2}	0.86	1.36×10^{-2}	0.86

Tabela 5.5: Resultados de Simulação ($t = 0s$ até $t = 1s$) - Bondarenko *et al.*

Nesta seção apresentaremos os resultados em tempo de computação obtidos para os métodos não-adaptativos e sem discretização particionada (Gomes *et al.*, 2015). As Tabelas 5.5, 5.6, 5.7 e 5.8 comparam os tempos de computação obtidos pelos *solvers* numéricos de passo fixo implementados para diferentes tolerâncias de erro numérico (tol). A tolerância tol representa um valor máximo para os erros numéricos calculados a partir da Equação (4.13). Em tais Tabelas, h denota o maior valor de passo de tempo que gerou

Método	Tolerância de 0.2%		Tolerância de 1%		Tolerância de 5%	
	h(ms)	TC(s)	h(ms)	TC(s)	h(ms)	TC(s)
Euler	1.14×10^{-4}	24.71	1.14×10^{-4}	24.71	1.14×10^{-4}	24.71
RL	1.14×10^{-4}	26.47	1.14×10^{-4}	26.47	1.14×10^{-4}	26.47
Euler + UNI	1.10×10^{-3}	3.82	2.51×10^{-3}	1.76	2.98×10^{-3}	1.47
RL + UNI	1.10×10^{-3}	3.94	2.52×10^{-3}	1.82	2.99×10^{-3}	1.53
SAST1	1.18×10^{-5}	612.64	5.92×10^{-5}	138.37	2.26×10^{-4}	37.45
SAST1 + UNI	1.32×10^{-3}	5.61	1.40×10^{-2}	0.60	2.17×10^{-2}	0.41
RK2	1.14×10^{-4}	48.75	1.14×10^{-4}	48.75	1.14×10^{-4}	48.75
RL2	1.14×10^{-4}	53.98	1.14×10^{-4}	53.98	1.14×10^{-4}	53.98
SAST2	1.74×10^{-4}	110.43	4.33×10^{-4}	44.34	2.52×10^{-3}	7.62

Tabela 5.6: Resultados de Simulação ($t = 0s$ até $t = 1s$) - Winslow *et al.*

Método	Tolerância de 0.2%		Tolerância de 1%		Tolerância de 5%	
	h(ms)	TC(s)	h(ms)	TC(s)	h(ms)	TC(s)
Euler	1.87×10^{-4}	15.35	1.87×10^{-4}	15.35	1.87×10^{-4}	15.35
RL	1.87×10^{-4}	15.41	1.87×10^{-4}	15.41	1.87×10^{-4}	15.41
Euler + UNI	1.41×10^{-3}	7.89	3.16×10^{-3}	3.90	3.27×10^{-3}	3.82
RL + UNI	1.43×10^{-3}	7.86	3.16×10^{-3}	3.94	3.27×10^{-3}	3.87
SAST1	6.50×10^{-5}	138.23	2.99×10^{-4}	29.91	1.42×10^{-3}	6.26
SAST1 + UNI	3.53×10^{-2}	0.80	4.45×10^{-2}	0.69	4.45×10^{-2}	0.69
RK2	1.87×10^{-4}	28.01	1.87×10^{-4}	28.01	1.87×10^{-4}	28.01
RL2	1.87×10^{-4}	31.02	1.87×10^{-4}	31.02	1.87×10^{-4}	31.02
SAST2	5.14×10^{-4}	35.64	1.51×10^{-3}	12.09	4.01×10^{-3}	4.58

Tabela 5.7: Resultados de Simulação ($t = 0s$ até $t = 1s$) - Iyer *et al.*

Método	Tolerância de 0.2%		Tolerância de 1%		Tolerância de 5%	
	h(ms)	TC(s)	h(ms)	TC(s)	h(ms)	TC(s)
Euler	1.60×10^{-4}	130.04	1.60×10^{-4}	130.04	1.60×10^{-4}	130.04
RL	5.32×10^{-2}	0.45	2.41×10^{-1}	0.11	5.00×10^{-1}	0.05
SAST1	5.35×10^{-2}	0.82	2.38×10^{-1}	0.19	5.00×10^{-1}	0.09
RK2	1.60×10^{-4}	254	1.60×10^{-4}	254	1.60×10^{-4}	254
RL2	9.02×10^{-2}	0.55	2.78×10^{-1}	0.18	5.00×10^{-1}	0.10
SAST2	1.05×10^{-1}	1.02	3.22×10^{-1}	0.34	5.00×10^{-1}	0.22

Tabela 5.8: Resultados de Simulação ($t = 0s$ até $t = 10s$) - Tusscher-Panfilov.

Modelo	Tolerância de 0.2%		Tolerância de 1%		Tolerância de 5%	
	Euler	RL	Euler	RL	Euler	RL
Bondarenko <i>et al.</i>	3.82	4.01	27.70	29.12	85.07	89.43
Winslow <i>et al.</i>	4.40	4.72	41.18	44.12	60.27	64.56
Iyer <i>et al.</i>	19.19	19.26	22.25	22.33	22.25	22.33

Tabela 5.9: Ganhos de desempenho do método SAST1 + UNI em relação aos métodos de Euler e de Rush-Larsen.

solução com erros numéricos inferiores às tolerâncias predefinidas de 0.2%, 1.0% e 5.0%, e *TC* trata-se de uma abreviação para *Tempo de Computação*.

Podemos notar que, para os modelos baseados em cadeias de Markov, os métodos

RL, RL2, Euler e RK2 apresentaram problemas similares de instabilidade numérica, uma vez que os máximos passos de tempo associados não puderam ser aumentados mesmo com o aumento da tolerância de 0.2% até 5.0%. Ou seja, para tais modelos, os passos h utilizados pelos métodos RK2, Euler, RL e RL2 são independentes da tolerância. Já para o modelo de Tusscher-Panfilov os métodos RL e RL2 puderam utilizar valores de h significativamente superiores em relação aos métodos de Euler e de RK2, o que justifica a grande utilização do método de Rush-Larsen por parte da comunidade de modelagem cardíaca. Fica clara a vantagem em estabilidade numérica oferecida pelo método RL em relação ao método de Euler para tal modelo, fato esse que já é bem conhecido (MacLachlan *et al.*, 2005).

Os métodos propostos por Sundnes *et al.* (SAST1 and SAST2) reduziram as restrições de estabilidades e permitiram o aumento de h com o aumento da tolerância numérica. O mesmo acontece com os métodos híbridos baseados no método de uniformização (Euler + UNI, RL + UNI and SAST1 + UNI). Todos esses métodos permitiram a utilização de passos de tempo uma ordem de grandeza superiores àqueles utilizados pelos métodos de Euler e de RL. Para todos os casos, o único método que consistentemente permitiu que h fosse duas ordens de grandeza superior em relação aos métodos Euler e RL foi o método híbrido SAST1 + UNI.

Em termos de tempo de execução, os métodos SAST1 e SAST2 foram sempre mais lentos que os métodos Euler e RL para a estrita tolerância de 0.2% (com exceção para o método SAST2 aplicado ao modelo de Bondarenko *et al.*). Tais métodos foram consistentemente mais rápidos que os métodos RL e Euler somente para a tolerância de 5%. No entanto, para o caso particular do modelo de Tusscher-Panfilov, esses métodos foram mais lentos que o RL para todas as tolerâncias consideradas.

Por outro lado, todos os métodos híbridos baseados no método de uniformização obtiveram desempenho superior aos métodos Euler e RL para todos os modelos baseados em CMs e para todas as tolerâncias consideradas. Além disso, o método híbrido mais rápido foi o SAST1 + UNI, o qual, conforme descrito anteriormente, foi o que permitiu a utilização dos maiores valores h . Os resultados sugerem que o método explícito de

uniformização é uma técnica viável e que melhora consideravelmente os problemas de estabilidade na solução de modelos baseados na formulação de Markov. Os ganhos de desempenho atingidos com o método SAST1 + UNI estão resumidos na Tabela 5.9. SAST1 + UNI foi até 19 vezes mais rápido que os métodos RL e Euler para o caso de tolerância estrita ($tol = 0.2\%$) e até 89 vezes mais rápido para o caso de tolerância relaxada ($tol = 5\%$).

5.3.2 Métodos com Discretização Temporal Particionada

Método	tol = 0.2%			tol = 1%			tol = 5%		
	h_1 (ms)	D	TC(s)	h_1 (ms)	D	TC(s)	h_1 (ms)	D	TC(s)
Euler + UNI / DTP	6.34×10^{-4}	100	2.17	1.02×10^{-3}	100	1.32	1.02×10^{-3}	100	1.32
RL / DTP	1.97×10^{-4}	100	9.41	1.97×10^{-4}	100	9.41	1.97×10^{-4}	100	9.41
RL + UNI / DTP	6.34×10^{-4}	100	2.53	1.02×10^{-3}	100	1.60	1.02×10^{-3}	100	1.60
SAST1 / DTP	4.22×10^{-4}	100	19.18	1.96×10^{-3}	10	4.18	6.99×10^{-3}	4	1.20
SAST1 + UNI / DTP	2.53×10^{-3}	20	1.51	2.10×10^{-2}	12	0.23	7.24×10^{-2}	5	0.08

Tabela 5.10: Resultados de simulação com discretização particionada ($t = 0s$ até $t = 1s$) - Bondarenko *et al.*

Método	tol = 0.2%			tol = 1%			tol = 5%		
	h_1 (ms)	D	TC(s)	h_1 (ms)	D	TC(s)	h_1 (ms)	D	TC(s)
Euler + UNI / DTP	1.10×10^{-3}	20	1.63	2.00×10^{-3}	30	0.87	2.40×10^{-3}	30	0.70
RL / DTP	1.14×10^{-4}	100	16.65	1.14×10^{-4}	100	16.65	1.14×10^{-4}	100	16.65
RL + UNI / DTP	1.10×10^{-3}	20	1.19	2.00×10^{-3}	30	0.65	2.40×10^{-3}	30	0.53
SAST1 / DTP	1.18×10^{-5}	100	600.63	5.92×10^{-5}	100	133.05	2.26×10^{-4}	10	35.01
SAST1 + UNI / DTP	1.32×10^{-3}	11	1.69	1.10×10^{-2}	5	0.47	2.00×10^{-2}	4	0.28

Tabela 5.11: Resultados de simulação com discretização particionada ($t = 0s$ até $t = 1s$) - Winslow *et al.*

Método	tol = 0.2%			tol = 1%			tol = 5%		
	h_1 (ms)	D	TC(s)	h_1 (ms)	D	TC(s)	h_1 (ms)	D	TC(s)
Euler + UNI / DTP	1.41×10^{-3}	40	1.51	3.16×10^{-3}	80	0.71	3.25×10^{-3}	100	0.67
R / DTPL	1.86×10^{-4}	100	15.28	1.86×10^{-4}	100	15.28	1.86×10^{-4}	100	15.28
RL + UNI / DTP	1.43×10^{-3}	40	1.47	3.16×10^{-3}	80	0.70	3.25×10^{-3}	100	0.68
SAST1 / DTP	6.50×10^{-5}	100	137.98	2.99×10^{-4}	40	29.87	1.42×10^{-3}	9	6.21
SAST1 + UNI / DTP	2.05×10^{-2}	2	0.78	1.10×10^{-2}	15	0.68	1.00×10^{-2}	50	0.57

Tabela 5.12: Resultados de simulação com discretização particionada ($t = 0s$ até $t = 1s$) - Iyer *et al.*

Método	tol = 0.2%			tol = 1%			tol = 5%		
	h_1 (ms)	D	TC(s)	h_1 (ms)	D	TC(s)	h_1 (ms)	D	TC(s)
RL / DTP	5.32×10^{-2}	2	0.34	2.41×10^{-1}	2	0.08	5.00×10^{-1}	1	0.05
SAST1 / DTP	5.35×10^{-2}	2	0.66	2.38×10^{-1}	2	0.15	5.00×10^{-2}	1	0.09

Tabela 5.13: Resultados de simulação com discretização particionada ($t = 0s$ até $t = 10s$) - Tusscher-Panfilov.

Tol. Método	Bondarenko <i>et al.</i>			Winslow <i>et al.</i>			Iyer <i>et al.</i>			Tusscher-Panfilov		
	0.2%	1%	5%	0.2%	1%	5%	0.2%	1%	5%	0.2%	1%	5%
Euler + UNI / DTP	3.54	3.62	3.62	2.34	2.02	2.09	5.21	5.50	5.68	-	-	-
RL / DTP	1.33	1.33	1.33	1.63	1.63	1.63	1.01	1.01	1.01	1.32	1.37	1.00
RL + UNI / DTP	3.14	3.11	3.11	3.31	2.82	2.88	5.36	5.64	5.66	-	-	-
SAST1 / DTP	1.07	1.05	1.04	1.02	1.04	1.07	1.00	1.00	1.01	1.24	1.27	1.00
SAST1 + UNI / DTP	2.06	1.86	1.79	3.31	1.27	1.44	1.02	1.01	1.21	-	-	-

Tabela 5.14: Ganhos de desempenho utilizando discretização particionada.

As Tabelas 5.10, 5.11, 5.12 e 5.13 comparam os resolvidores numéricos de 1ª ordem implementados com discretização temporal particionada, utilizando dois diferentes valores de passo de tempo: h_1 e $h_2 = D \times h_1$. Buscamos utilizar o passo de tempo inferior h_1 o mais próximo possível dos valores correspondentes de h apresentados nas Tabelas 5.5, 5.6, 5.7 e 5.8. O passo de tempo maior, h_2 , foi incrementado para um valor máximo, respeitando as tolerâncias numéricas predefinidas e o limite superior de $100 \times h_1$, ou seja, limitamos a diferença entre h_1 e h_2 em duas ordens de grandeza.

Em termos do passo de tempo maior (h_2) notamos que, para todos os modelos baseados em formulações de Markov, h_2 pode ser aumentado de uma a duas ordens de grandeza em relação a h_1 . No entanto, para alguns casos - especialmente para modelo de Tusscher-Panfilov -, tal aumento limitou-se a uma ordem de grandeza ou menos (para dois casos extremos do modelo de Tusscher-Panfilov, não foi possível aumentar h_2 em relação a h_1). De todo modo, fomos capazes de aumentar h_2 em relação a h_1 para praticamente todos os casos. Isso sugere que o particionamento da discretização temporal foi muito eficiente para a maioria dos métodos em que foi aplicado.

A utilização de dois diferentes passos de tempo também refletiu nos tempos de execução. Ganhos de desempenho foram quase sempre alcançados para cada modelo, método e tolerância, e estão resumidos na Tabela 5.14. Os métodos que foram mais fortemente beneficiados pelo esquema de particionamento foram aqueles que utilizam o método de Euler para algumas equações nos modelos: RL, Euler+ UNI e RL + UNI (veja a Tabela 4.1). Os métodos particionados foram até 6 vezes mais rápidos para o modelo de Iyer *et al.*

Recentemente, em Marsh *et al.* (2012), uma técnica diferenciada de particionamento foi proposta e testada. O particionamento foi baseado na *rigidez* de cada equação de um

modelo, o que exigiu o cálculo prévio dos autovalores ao longo da integração do mesmo. O método de Euler foi utilizado para todo o sistema de EDOs sobre a porção não rígida do domínio temporal e o método SAST1 foi utilizado para as variáveis *stiff*. Os ganhos de desempenho reportados em Marsh *et al.* (2012) ficaram entre 12% a 9 vezes mais rápidos que os métodos SAST1 e RL. Deste modo, ambos os tipos de particionamento alcançaram resultados similares.

5.3.3 Métodos de Passo Adaptativo

Nesta seção iremos apresentar os resultados referentes aos tempos de computação obtidos pelos métodos de passo de tempo adaptativo apresentados nos Capítulos 3 e 4. Primeiramente, apresentamos os resultados obtidos pela utilização do método de Rush-Larsen adaptativo (RL / ADP) para o modelo proposto por Tusscher e Panfilov (2006). A Tabela 5.15 compara os tempos de computação referentes aos métodos Euler, Euler / ADP, RL / ADP, RL e SAST1 para diferentes tolerâncias numéricas. Comparamos o método RL / ADP com SAST1 e RL pelo fato dos dois últimos terem sido os métodos de 1ª ordem com melhor desempenho para o modelo em questão, de acordo com a Tabela 5.8.

Método	Tolerância de 0.2%		Tolerância de 1%		Tolerância de 5%	
	h(ms) / τ	TC(s)	h(ms) / τ	TC(s)	h(ms) / τ	TC(s)
Euler	1.60×10^{-4}	130.04	1.60×10^{-4}	130.04	1.60×10^{-4}	130.04
Euler / ADP	$\tau = 0.08$	92.24	$\tau = 0.08$	92.24	$\tau = 0.08$	92.24
RL	5.32×10^{-2}	0.45	2.41×10^{-1}	0.11	5.00×10^{-1}	0.05
RL / ADP	$\tau = 0.02$	0.08	$\tau = 0.08$	0.07	$\tau = 0.08$	0.07
SAST1	5.35×10^{-2}	0.82	2.38×10^{-1}	0.19	5.00×10^{-1}	0.09

Tabela 5.15: Resultados de simulações para métodos adaptativos - Tusscher e Panfilov (2006).

Na Tabela 5.15, τ representa a tolerância local adotada para o método adaptativo. Podemos notar que o método RL / ADP superou em desempenho o método RL para todas as tolerâncias, exceto a de 5 %. Isso se deve ao fato de um limite superior ter sido adotado para o passo de tempo h , conforme descrito na Seção 4.8. Para a tolerância de 5 %, o método RL permaneceu toda a execução com h em seu valor máximo permitido (0.5ms).

Deste modo, mesmo que o método RL / ADP também permanecesse toda a simulação com $h = 0.5\text{ms}$, seu tempo de computação ainda seria superior àquele observado para o método RL, devido a custos adicionais inerentes às avaliações de erros locais.

Com relação aos métodos híbridos de passo adaptativo, realizamos simulações utilizando os modelos baseados em cadeias de Markov para avaliar o desempenho dos resolvidores Euler + UNI / ADP, RL + UNI / ADP e SAST1 + UNI / ADP. Para tais métodos foi necessário um ajuste sobre a tolerância do método de uniformização, de modo a assegurarmos estabilidade numérica para simulações longas. Seu valor foi alterado para $\lambda = 10^{-8}$ nos casos dos modelos de Bondarenko *et al.* e Iyer *et al.*, e mantido a $\lambda = 10^{-6}$ para o modelo de Winslow *et al.* Para todos os métodos híbridos adaptativos, foi utilizado o passo de tempo inicial $h = 10^{-3}\text{ms}$. Ressaltamos que a magnitude do passo de tempo inicial pouco influenciou o desempenho global das simulações.

Método	Tolerância de 0.2%		Tolerância de 1%		Tolerância de 5%	
	$h(\text{ms}) / \tau$	TC(s)	$h(\text{ms}) / \tau$	TC(s)	$h(\text{ms}) / \tau$	TC(s)
Euler	1.97×10^{-4}	11.91	1.97×10^{-4}	11.91	1.97×10^{-4}	11.91
Euler / ADP	$\tau = 0.001$	1.92	$\tau = 0.005$	0.94	$\tau = 0.04$	0.73
RL	1.97×10^{-4}	12.52	1.97×10^{-4}	12.52	1.97×10^{-4}	12.52
RL / ADP	$\tau = 0.001$	2.14	$\tau = 0.005$	1.05	$\tau = 0.04$	0.79
Euler + UNI	$h = 6.34 \times 10^{-4}$	7.65	$h = 1.02 \times 10^{-3}$	4.79	$h = 1.02 \times 10^{-3}$	4.79
Euler + UNI / ADP	$\tau = 0.01$	3.09	$\tau = 0.02$	2.52	$\tau = 0.05$	2.21
RL + UNI	$h = 6.34 \times 10^{-4}$	7.96	$h = 1.02 \times 10^{-3}$	4.98	$h = 1.02 \times 10^{-3}$	4.98
RL + UNI / ADP	$\tau = 0.01$	3.27	$\tau = 0.02$	2.72	$\tau = 0.05$	2.31
SAST1 + UNI	$h = 2.53 \times 10^{-3}$	3.12	$h = 2.01 \times 10^{-2}$	0.43	$h = 7.24 \times 10^{-2}$	0.14
SAST1 + UNI / ADP	$\tau = 0.005$	1.14	$\tau = 0.08$	0.33	$\tau = 0.1$	0.19

Tabela 5.16: Resultados de Simulação para Métodos Adaptativos ($t = 0\text{s}$ até $t = 1\text{s}$) - Bondarenko *et al.*

Na Tabela 5.16 mostramos os resultados obtidos para o modelo de Bondarenko *et al.* Podemos notar que os métodos híbridos com passo adaptativo obtiveram, na maior parte dos casos, ganhos de aproximadamente 2 vezes em desempenho com relação aos seus respectivos métodos híbridos de passo fixo. O único caso em que a versão adaptativa obteve desempenho inferior à versão de passo fixo foi para o método SAST1+UNI com tolerância de 5 %. Neste caso, o passo de tempo fixo do método SAST1 + UNI é significativamente grande, de modo que a versão adaptativa não pôde obter tempo de computação inferior, mesmo utilizando passos de tempo maiores, devido a significativos custos computacionais adicionais. Na Figura 5.1 mostramos um gráfico para o passo de tempo h em função do tempo, obtido pelo método SAST1 + UNI / ADP para o modelo de

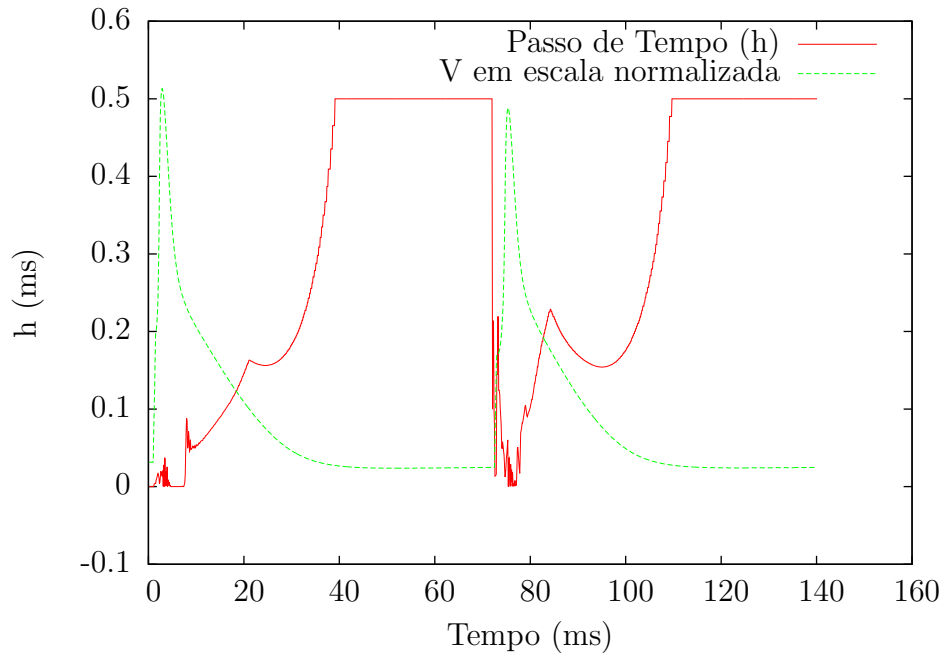


Figura 5.1: Gráfico h versus tempo para o método SAST1 + UNI / ADP ($\tau = 0.02$), aplicado ao modelo de Bondarenko *et al.*

Bondarenko *et al.*, utilizando uma tolerância local de 0.02. No mesmo gráfico mostramos o potencial transmembrânico V em uma escala normalizada para ilustrar como a magnitude de h é ajustada em função da variação sofrida por V durante o potencial de ação.

Método	Tolerância de 0.2%		Tolerância de 1%		Tolerância de 5%	
	$h(\text{ms}) / \tau$	TC(s)	$h(\text{ms}) / \tau$	TC(s)	$h(\text{ms}) / \tau$	TC(s)
Euler	1.14×10^{-4}	24.71	1.14×10^{-4}	24.71	1.14×10^{-4}	24.71
Euler / ADP	$\tau = 0.005$	1.12	$\tau = 0.08$	1.06	$\tau = 0.08$	1.06
RL	1.14×10^{-4}	26.47	1.14×10^{-4}	26.47	1.14×10^{-4}	26.47
RL / ADP	$\tau = 0.005$	1.18	$\tau = 0.08$	1.16	$\tau = 0.08$	1.16
Euler + UNI	$h = 1.10 \times 10^{-3}$	3.82	$h = 2.51 \times 10^{-3}$	1.76	$h = 2.98 \times 10^{-3}$	1.47
Euler + UNI / ADP	$\tau = 0.04$	0.44	$\tau = 0.04$	0.44	$\tau = 0.05$	0.42
RL + UNI	$h = 1.10 \times 10^{-3}$	3.94	$h = 2.52 \times 10^{-3}$	1.82	$h = 2.99 \times 10^{-3}$	1.53
RL + UNI / ADP	$\tau = 0.04$	0.46	$\tau = 0.04$	0.46	$\tau = 0.05$	0.45
SAST1 + UNI	$h = 1.32 \times 10^{-3}$	5.61	$h = 1.40 \times 10^{-2}$	0.60	$h = 2.17 \times 10^{-2}$	0.41
SAST1 + UNI / ADP	$\tau = 0.001$	0.33	$\tau = 0.02$	0.25	$\tau = 0.04$	0.23

Tabela 5.17: Resultados de Simulação para Métodos Adaptativos ($t = 0s$ até $t = 1s$) - Winslow *et al.*

Apresentamos na Tabela 5.17 os resultados referentes aos métodos de tempo adaptativo para o modelo de Winslow *et al.* Neste caso, os métodos adaptativos obtiveram ganhos significativos para todas as tolerâncias consideradas em relação aos métodos de passo fixo. Um ponto a ser destacado com relação ao modelo de Winslow *et al.* é o número relativamente baixo de cadeias de Markov (apenas duas), o que pode ter influenciado na redução do custo adicional gerado pelo método adaptativo quando aplicado a este modelo.

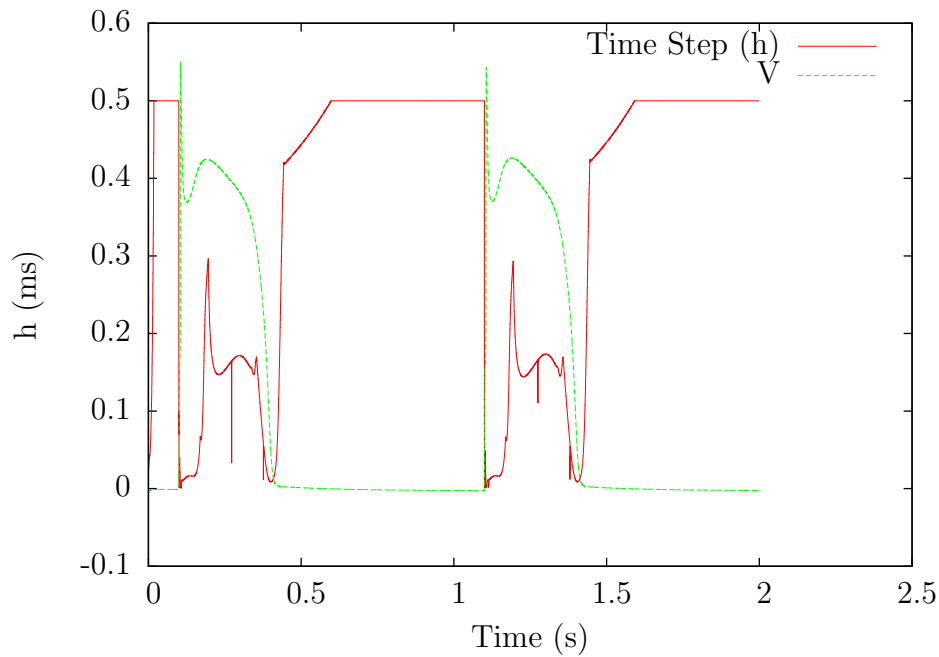


Figura 5.2: Gráfico h versus tempo para o método SAST1 + UNI / ADP ($\tau = 0.001$), aplicado ao modelo de Winslow *et al.*

Conforme mostrado na Figura 5.2, podemos notar que, para o método SAST1 + UNI / ADP, o passo de tempo h permaneceu no valor limite durante a maior parte da simulação.

Na Tabela 5.18 são mostrados os resultados de simulação para o modelo de Iyer *et al.* utilizando-se os métodos de tempo adaptativo e os métodos de passo fixo associados. Para este modelo, todos os métodos de passo adaptativo obtiveram ganhos em relação aos respectivos métodos de passo fixo, para todas as tolerâncias consideradas. A Figura 5.3 mostra um gráfico para o passo de tempo h utilizado pelo método SAST1 + UNI / ADP, juntamente com o potencial transmembrânico em escala normalizada.

Método	Tolerância de 0.2%		Tolerância de 1%		Tolerância de 5%	
	$h(\text{ms}) / \tau$	TC(s)	$h(\text{ms}) / \tau$	TC(s)	$h(\text{ms}) / \tau$	TC(s)
Euler	1.87×10^{-4}	15.35	1.87×10^{-4}	15.35	1.87×10^{-4}	15.35
Euler / ADP	$\tau = 0.08$	7.76	$\tau = 0.08$	7.76	$\tau = 0.08$	7.76
RL	1.87×10^{-4}	15.41	1.87×10^{-4}	15.41	1.87×10^{-4}	15.41
RL / ADP	$\tau = 0.08$	7.83	$\tau = 0.08$	7.83	$\tau = 0.08$	7.83
Euler + UNI	$h = 1.41 \times 10^{-3}$	7.89	$h = 3.16 \times 10^{-3}$	3.90	$h = 3.27 \times 10^{-3}$	3.82
Euler + UNI / ADP	$\tau = 0.06$	2.03	$\tau = 0.08$	1.80	$\tau = 0.08$	1.80
RL + UNI	$h = 1.43 \times 10^{-3}$	7.86	$h = 3.16 \times 10^{-3}$	3.94	$h = 3.27 \times 10^{-3}$	3.87
RL + UNI / ADP	$\tau = 0.06$	2.06	$\tau = 0.08$	1.82	$\tau = 0.08$	1.82
SAST1 + UNI	$h = 3.53 \times 10^{-2}$	0.80	$h = 4.45 \times 10^{-2}$	0.69	$h = 4.45 \times 10^{-2}$	0.69
SAST1 + UNI / ADP	$\tau = 0.006$	0.69	$\tau = 0.04$	0.55	$\tau = 0.08$	0.53

Tabela 5.18: Resultados de Simulação para Métodos Adaptativos ($t = 0s$ até $t = 1s$) - Iyer *et al.*

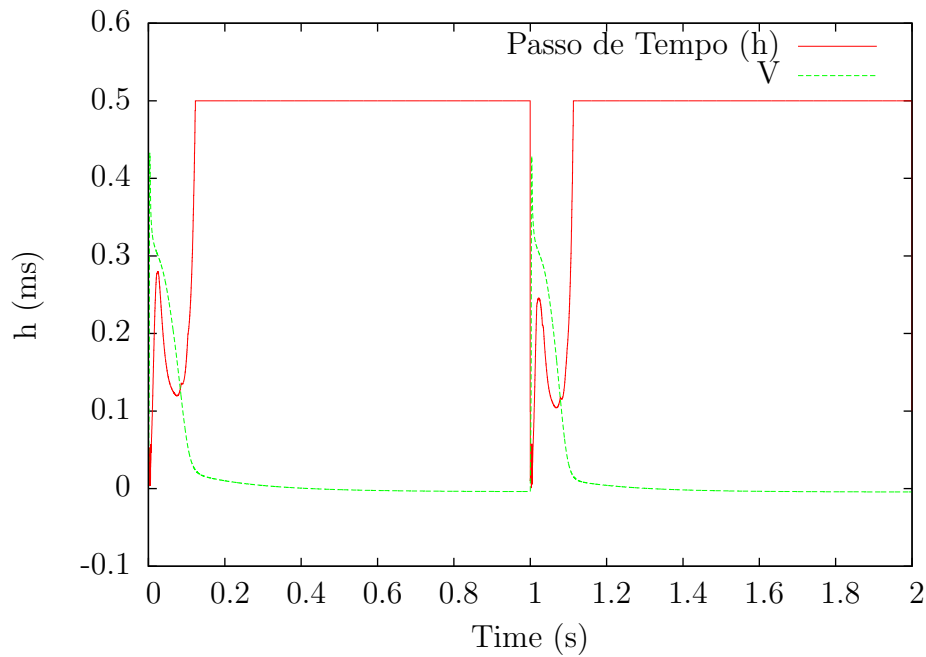


Figura 5.3: Gráfico h versus tempo para o método SAST1 + UNI / ADP ($\tau = 0.006$), aplicado ao modelo de Iyer *et al.*

5.4 Discussão

5.4.1 Métodos Híbridos

Embora os resultados apresentados neste trabalho foram obtidos a partir de quatro modelos para miócitos, eles fornecem material e informação suficientes para que possamos fazer, com algumas precauções, generalizações a respeito de como outros modelos de miócitos cardíacos irão beneficiar-se dos métodos numéricos apresentados.

As primeiras duas mensagens importantes são: (i) o método híbrido SAST1 + UNI alcançou a melhor performance de tempo de execução para tratar modelos modernos de miócitos cardíacos baseados em formulações de cadeias de Markov (iremos discutir posteriormente sobre como essas duas técnicas complementam-se para possibilitar tais ganhos de desempenho); (ii) embora tal método tenha possibilitado aumentos significativos nos passos de tempo h , o mesmo não pode ser considerado incondicionalmente estável. Este fato pode ser facilmente verificado a partir dos resultados de SAST1 + UNI para o modelo de Iyer *et al.* (veja a Tabela 5.7): o passo de tempo não pôde ser aumentado após a relaxação da tolerância numérica de 1% para 5% devido

a questões de estabilidade.

Vamos analisar o desempenho desses métodos. Lembramos ao leitor que o tempo de execução é o produto do custo de uma simples iteração pelo número total de iterações, o que é inversamente proporcional ao passo de tempo utilizado h . Os métodos SAST1, SAST2 e todos os métodos híbridos implementados são mais complexos e possuem um maior custo por iteração que os métodos Euler e RL. Em nossos experimentos, SAST1 ficou entre 1.4 e 4.1 vezes mais custoso que o método RL. O *overhead* inerente ao método de uniformização pode ser medido pelo custo do método Euler + UNI, que ficou entre 1.8 e 5.7 mais custoso que Euler e RL, ou seja, comparável ao custo do método SAST1. O método SAST2 foi aproximadamente duas vezes mais custoso que o SAST1 e apresentou um custo por passo de tempo entre 3.8 e 6.8 mais alto que aquele apresentado pelo método RL.

Podemos fornecer fórmulas analíticas para o custo de tais métodos. Por exemplo, vamos considerar o método de Euler e assumir que o custo para calcular cada lado direito f^j seja igual a C . O custo por iteração do método de Euler (ou RL) será: $Custo_{Euler} = N_{var}C$. Para o método SAST1 precisamos da computação do lado direito como no método de Euler e a computação da derivada parcial de f^j para cada variável. Entretanto, para uma equação quasi-linear, tal custo é próximo de zero, enquanto para uma equação totalmente não-linear será C . O limite inferior para a razão entre os custos de SAST1 e RL é dado por: $R_1 = 1 + N_{nl}/N_{var}$. Para o modelo de Tusscher-Panfilov, temos $N_{var} = 19$ e $N_{NL} = 7$, o que fornece $R_1 = 1.37$. De fato, por nossos resultados, observamos que SAST1 foi cerca de 1.8 mais lento que o método RL para o modelo de Tusscher-Panfilov. Fórmulas similares podem ser derivadas para todos os métodos numéricos apresentados neste trabalho. Esta é uma importante informação *a-priori* que pode ser obtida para qualquer combinação particular de método e de modelo. Entretanto, o custo em si não se correlaciona com o tempo final de execução. Por exemplo, para o modelo de Iyer *et al.* o custo por passo de tempo de Euler + UNI foi aproximadamente duas vezes maior em comparação ao SAST1. No entanto, como apresentado na Tabela 5.7, os tempos de execução de Euler + UNI foram sempre menores que aqueles associados

ao método SAST1. A razão para isso é que, neste caso, os passos de tempo h utilizados por Euler + UNI foram sempre maiores que aqueles utilizados pelo SAST1.

Todos os métodos explícitos implementados e testados neste trabalho são condicionalmente estáveis, ou seja, para cada par método-modelo existe um limite para o qual h não pode ser aumentado. Como iremos discutir posteriormente, isso está relacionado com os autovalores da matriz Jacobiana do sistema de EDOs. Entretanto, aqui vão mais duas mensagens importantes deste trabalho: (i) para a solução de cadeias de Markov o método de uniformização é mais estável que o método SAST1; (ii) a combinação específica de SAST1 e uniformização permitiu a utilização de passos de tempo ainda maiores. A primeira afirmação pode ser verificada comparando os resultados de Euler + UNI e SAST1 para os três modelos baseados em CMs (Tabelas 5.5, 5.6 e 5.7). Dentre 9 testes (3 modelos vezes 3 tolerâncias), Euler + UNI mostrou-se mais rápido que o método SAST1 em 7 casos.

Em adição a tais resultados quantitativos, podemos também fornecer uma explicação analítica e quantitativa para dar suporte à primeira afirmação. Se considerarmos uma cadeia de Markov com taxas de transição fixas, o sistema linear de EDOs associado é dado por $\frac{d\vec{Y}}{dt} = A\vec{Y}$, onde a matriz exponencial e^{tA} é definida pela série de Taylor. Portanto, nesse cenário, o método de uniformização é incondicionalmente estável. Por outro lado, se usarmos a decomposição de Jacobi e escrevermos $A = D + R$, com D sendo uma matriz diagonal, o método SAST1 (dado pela Equação (3.12)) pode ser reescrito na seguinte forma vetorial:

$$\vec{Y}_{n+1} = e^{hD}\vec{Y}_n + RD^{-1}(e^{hD} - I)\vec{Y}_n.$$

Podemos notar que, para h suficientemente pequeno, $(e^{hD} - I) \approx (hD)$. Portanto, a iteração do método SAST1 é próxima a

$$\vec{Y}_{n+1} = e^{hD}\vec{Y}_n + hR\vec{Y}_n,$$

de modo que SAST1 avança combinando um termo exponencial e^{hD} com um clássico

termo de Euler explícito hR . O último dos dois termos é conhecido por trazer características de instabilidade. Esta análise traz um melhor entendimento sobre o porquê da maior estabilidade oferecida pelo método de uniformização em relação ao método SAST1 para modelos baseados em cadeias de Markov.

No entanto, como mencionado anteriormente, o método Euler + UNI nem sempre mostrou desempenho superior ao método SAST1. Por outro lado, o método SAST1 + UNI atingiu melhor desempenho consistentemente. De fato, o método de uniformização é o mais indicado para as equações CM, enquanto o método SAST1 mostra-se melhor que o método de Euler para equações *stiff* não-lineares, como descrito em Sundnes *et al.* (2009); Marsh *et al.* (2012). De modo a investigarmos tais resultados, realizamos uma análise de rigidez baseada nos autovalores de cada um dos três modelos baseados em CMs testados, como descrito em Marsh *et al.* (2012).

Para o modelo de Bondarenko *et al.* identificamos que as variáveis mais rígidas durante as simulações foram P_{O_1} (uma variável do tipo MK) associada a um autovalor de $-1.0 \times 10^4 \text{ ms}^{-1}$ e $[Ca^{2+}]_{SS}$ (variável do tipo NL) associada a um autovalor de $-2.1 \times 10^3 \text{ ms}^{-1}$. Portanto, sem o método SAST1 o segundo autovalor - associado a uma variável totalmente não-linear - limitou a estabilidade do método Euler + UNI. Para este modelo particular, apenas com a combinação SAST1 + UNI fomos capazes de superar em desempenho o método SAST1 para todas as tolerâncias adotadas. Na Figura 5.4 mostramos a rápida variação sofrida pela variável P_{O_1} durante a fase de despolarização de um potencial de ação produzido pelo modelo de Bondarenko *et al.*

Para os outros dois modelos baseados em CMs, o método SAST1 + UNI também aumentou o desempenho e a estabilidade em relação a Euler + UNI ou a SAST1. Entretanto, para esses dois modelos Euler+UNI foi suficiente para superar o método SAST1 em desempenho. Isso também pode ser explicado por análise de rigidez e autovalores. Para o modelo de Winslow *et al.* as variáveis mais rígidas são P_{C_1} (do tipo MK) and $[Ca^{2+}]_{SS}$ (variável do tipo NL), com as correspondentes partes reais de autovalores $-1.4 \times 10^4 \text{ ms}^{-1}$ e $-8.5 \times 10^2 \text{ ms}^{-1}$, respectivamente. A magnitude do autovalor associado à variável NL é menor comparada à variável *stiff* NL do modelo de Bondarenko

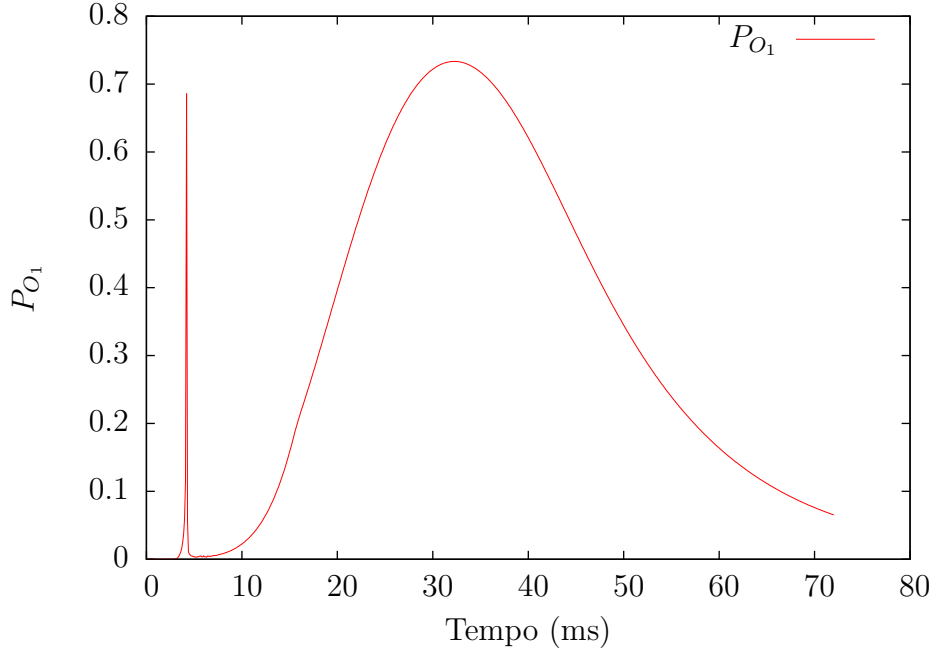


Figura 5.4: Variável rígida P_{O_1} versus tempo durante um PA do modelo de Bondarenko *et al.*

et al., o que explica a diferença observada em relação a tempos de computação. O mesmo foi observado para o modelo de Iyer *et al.*: a variável MK mais rígida é P_{C_1} (parte real de autovalor igual a $-1.1 \times 10^4 ms^{-1}$), enquanto a variável NL mais rígida é $[Ca^{2+}]_{ss}$ (parte real de autovalor igual a $-8.2 \times 10^2 ms^{-1}$). A Figura 5.5 mostra como a variável $[Ca^{2+}]_{ss}$ do modelo de Winslow *et al.* se comporta durante um PA.

A partir da discussão acima, podemos estender nossos resultados para outros modelos não considerados neste trabalho. Por exemplo, a Tabela IV em Marsh *et al.* (2012) lista dois outros modelos que possuem formulações em cadeias de Markov juntamente com suas variáveis *stiff*, i.e., aquelas associadas aos autovalores de grande magnitude: o modelo de Jafri *et al.* (1998), com três variáveis de Markov e uma totalmente não-linear, e o modelo de Pandit *et al.* (2003), com duas variáveis de Markov. O método SAST1 + UNI iria apresentar bom desempenho para ambos os modelos, porém Euler + UNI poderia mostrar desempenho superior para o modelo de Pandit *et al.*, devido à ausência de uma variável *stiff* do tipo NL.

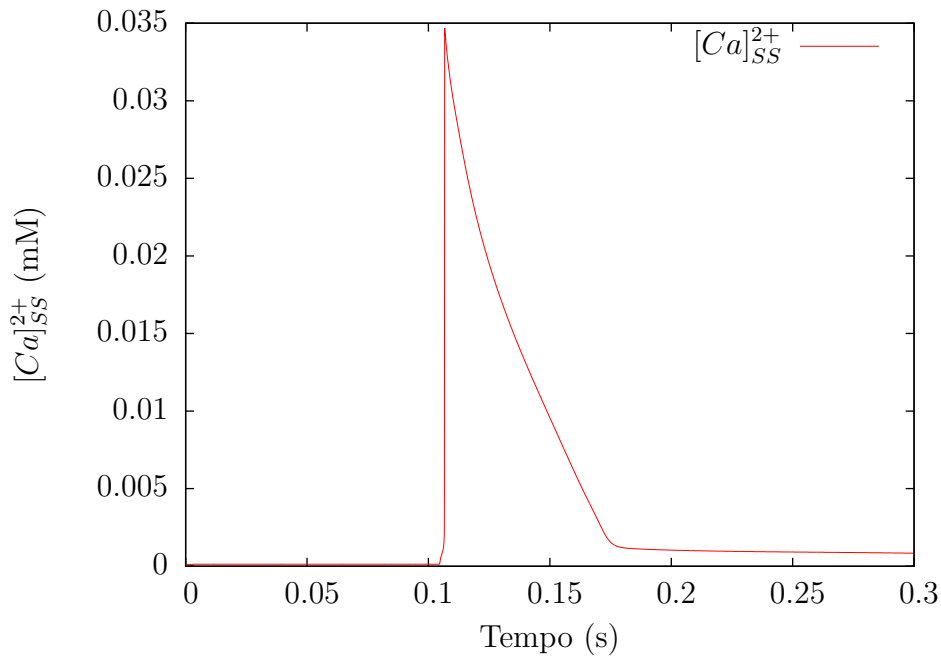


Figura 5.5: Variável rígida $[Ca^{2+}]_{ss}$ versus tempo durante um PA do modelo de Winslow *et al.*

5.4.2 Métodos Adaptativos

Os métodos adaptativos implementados obtiveram ganhos de desempenho em praticamente todas as situações em relação às respectivas versões de passo de tempo fixo. O método RL / ADP foi mais rápido que o método de Rush-Larsen para todas as tolerâncias e modelos, com exceção para o modelo de Tusscher-Panfilov no caso em que $tol = 5\%$. Em tal situação, o método de Rush-Larsen foi executado com o passo de tempo máximo estipulado (0.5ms), o que gera um impedimento teórico para que a versão adaptativa apresentasse desempenho superior. No entanto, para a tolerância estrita de 0.2%, o método RL / ADP mostrou-se cerca de 6 vezes mais rápido que o método RL para o modelo de Tusscher-Panfilov e de 2 a 25 vezes mais rápido que o método de Rush-Larsen para os modelos baseados em Cadeias de Markov. Para tais modelos, o método Euler / ADP mostrou-se ligeiramente mais eficiente que o método RL / ADP, uma vez que as restrições de estabilidade em tais modelos aplica-se de forma similar para os dois métodos (o que não ocorre para o modelo de Tusscher-Panfilov, em que o método de Rush-Larsen apresenta estabilidade consideravelmente superior).

Para o caso dos métodos híbridos adaptativos, situação semelhante foi evidenciada: os métodos obtiveram ganhos significativos para a tolerância de 0.2% em relação às versões de passo fixo e de discretização particionada. A Tabela 5.19 resume os ganhos em tempo de computação obtidos pelas diferentes versões do método SAST1 + UNI em relação ao método de Euler para os três modelos baseados em cadeias de Markov.

Modelo	Método	tol = 0.2%	tol = 1%	tol = 5%
Bondarenko <i>et al.</i>	SAST1+UNI	3.82	27.70	85.07
	SAST1+UNI / DTP	7.87	51.52	152.27
	SAST1+UNI / ADP	10.45	36.10	62.68
Winslow <i>et al.</i>	SAST1+UNI	4.40	41.18	60.27
	SAST1+UNI / DTP	14.57	52.30	86.79
	SAST1+UNI / ADP	74.8	98.83	107.44
Iyer <i>et al.</i>	SAST1+UNI	19.19	22.25	22.25
	SAST1+UNI / DTP	19.57	22.47	26.92
	SAST1+UNI / ADP	22.25	27.91	28.97

Tabela 5.19: Ganhos de desempenho em relação ao método de Euler por parte das variações do método SAST1 + UNI.

O método SAST1 + UNI / ADP foi a versão mais rápida em todos os casos, com exceção para o modelo de Bondarenko *et al.* com as tolerâncias de 1% e 5%. Para esses casos em específico, a utilização do método SAST1 + UNI / DTP seria a mais indicada. No entanto, o método adaptativo mostrou-se o mais rápido para todos os casos de tolerância numérica estrita ($tol = 0.2\%$). Com isso, podemos concluir que os esquemas híbridos adaptativos propostos neste trabalho são os mais adequados para a obtenção de bons desempenhos em aplicações que exigem elevada precisão numérica.

É importante observar que o método SAST1 + UNI / ADP apresenta custo por passo de tempo severamente superior em relação ao método Euler / ADP. No entanto, os tempos de computação obtidos pelo primeiro método são inferiores em relação ao segundo por apresentar melhores condições de estabilidade numérica.

5.4.3 Utilização de Memória

Embora a utilização do método de uniformização acarrete em um custo adicional de memória, podemos notar, a partir da Tabela 5.4, que tal *overhead* apresentou um valor máximo de 71% para um modelo tal que 85% de suas equações são do tipo MK (o modelo

de *Iyer et al.*). Deste modo, é seguro afirmar que grande parte das aplicações do método de uniformização em modelos de eletrofisiologia cardíaca implicará em *overheads* de memória inferiores a 100%. Conforme discutido nas duas seções anteriores, com os ganhos em tempo de computação oferecidos pelo método de uniformização e o seu custo de memória relativamente baixo, a utilização dos métodos híbridos aqui apresentados para a simulação de tecidos cardíacos seria perfeitamente plausível.

6 Conclusões e Trabalhos Futuros

Neste trabalho propusemos diferentes esquemas numéricos explícitos para a solução de modelos da eletrofisiologia cardíaca baseados em formulações de cadeias de Markov, comparando-os com métodos tradicionalmente utilizados pela comunidade científica. Os modelos mais recentes que descrevem detalhadamente o comportamento de canais iônicos utilizam cadeias de Markov para representar a complexa dinâmica observada em suas subestruturas proteicas. As equações diferenciais provenientes de tal modelagem são usualmente de característica *rígida*, exigindo a utilização de passos de tempo muito pequenos por parte dos esquemas explícitos tradicionais. Com a utilização do método de uniformização para a geração de esquemas híbridos, em conjunto com os métodos de Rush-Larsen, Euler Explícito e SAST1 (Sundnes *et al.*, 2009), foi possível aumentarmos em até duas ordens de grandeza os passos de tempo utilizados em relação ao métodos de Rush-Larsen e de Euler para os três modelos baseados em cadeias de Markov simulados. As versões de passo fixo de tais esquemas híbridos puderam alcançar consideráveis ganhos de desempenho em relação aos métodos Euler e RL, chegando ao patamar de 89 vezes para o método SAST1 + UNI aplicado ao modelo de Bondarenko *et al.* (2004).

O método de uniformização apresentou custos adicionais em termos de consumo de memória em decorrência da utilização de matrizes de transição, que precisam ser armazenadas e utilizadas durante a execução do método. No entanto, com a implementação de estruturas esparsas, tal custo adicional limitou-se a cerca de 70% em relação aos métodos Euler, RL e SAST1 para o modelo de Iyer *et al.* (2004), o qual possui 7 cadeias de Markov e cerca de 85% de equações diferenciais do tipo MK. Deste modo, podemos concluir que os métodos híbridos propostos oferecem elevados ganhos de estabilidade, sem comprometer sua utilização em *solvers* numéricos de tecidos celulares em razão de grandes impactos sobre o consumo de memória.

Sobre as versões de discretização particionada dos métodos híbridos implementados, os ganhos em tempo de computação com relação aos métodos Euler e RL chegaram a

valores de até 159 vezes (modelo de Bondarenko *et al.* com o método SAST1 + UNI, para a tolerância de 5%). Os métodos híbridos com passo adaptativo obtiveram desempenhos superiores aos métodos com discretização particionada para tolerâncias estritas. O método SAST1 + UNI / ADP foi até 75 vezes mais rápido que o método de Euler para a tolerância de 0.2% e atingiu tempos de computação consistentemente menores que a versão de passo de tempo particionado correspondente, para a mesma tolerância. Com base em tais fatos, os métodos híbridos de passo adaptativo propostos são altamente indicados para se obter bons desempenhos em situações que exigem precisão numérica elevada.

Como trabalho futuro pretende-se desenvolver, a partir do método híbrido de melhor desempenho obtido para as simulações aqui apresentadas (SAST1 + UNI), esquemas numéricos que combinam discretização particionada com adaptatividade temporal, visando englobar as vantagens observadas em ambas as abordagens. Além disso, almeja-se a aplicação de tais métodos para a simulação de tecidos cardíacos completos, a partir de modelos complexos baseados em cadeias de Markov. Para a obtenção de ganhos em desempenho ainda mais expressivos, pretende-se associar tais técnicas com esquemas de adaptatividade espacial e com recursos de computação paralela em múltiplas CPUs e em unidades de processamento gráfico (GPUs).

Referências Bibliográficas

- Aires, M. M., 2007. Fisiologia, terceira Edição. Guanabara Koogan.
- Alberts, B., Johnson, A., Lewis, J., 2003. Essential Cell Biology, segunda Edição. Garland Publishing.
- Aldrich, R., Corey, D., Stevens, C., 1983. A reinterpretation of mammalian sodium channel gating based on single channel recording. *Nature* 306, 436–441.
- Armstrong, C., Bezanilla, F., 1977. Inactivation of the sodium channel. ii. gating current experiments. *The Journal of General Physiology* 70, 567–590.
- Bondarenko, V. E., Gyula, P. S., Glenna, C. L. B., Kim, S.-J., Rasmusson, R. L., 2004. Computer model of action potential of mouse ventricular myocytes. *American Journal of Physiology* 287, H1378–H1403.
- Burden, R. L., Faires, J. D., 2008. Análise Numérica. Cengage Learning.
- Campos, F. O., 2008. Modelagem computacional da eletrofisiologia cardíaca: O desenvolvimento de um novo modelo para células de camundongos e avaliação de novos esquemas numéricos. Dissertação de mestrado, Universidade Federal de Juiz de Fora - Mestrado em modelagem computacional.
- Campos, R. S., Amorim, R. M., Costa, C. M., Oliveira, B. L., Barbosa, C. B., Sundnes, J., Santos, R. W., 2010. Approaching cardiac modeling challenges to computer science with CellML-based web tools. *Future Generation Computer Systems* 26, 462–470.
- Campos, R. S., Campos, F. O., Gomes, J. M., Barbosa, C. B., Lobosco, M., Santos,

- R. W., 2013. Comparing high performance techniques for the automatic generation of efficient solvers of cardiac cell models. *Computing* 95 (1), 639–660.
- Carvalho *et al.*, Grandes desafios da pesquisa em computação no Brasil. In: Seminário sobre os Grandes Desafios da Computação no Brasil, 2006. Sociedade Brasileira de Computação.
- Clancy, C., Rudy, Y., 2002. Na⁺ channel mutation that causes both brugada and long-qt syndrome phenotypes: a simulation study of mechanism. *Circulation* 105, 1208–1213.
- Constanzo, L. S., 2007. Fisiologia, terceira Edição. Elsevier.
- Costa, C. M., 2011. Modelagem da Microestrutura de Tecidos Cardíacos. Dissertação de mestrado, Universidade Federal de Juiz de Fora - Mestrado em modelagem computacional.
- Cuellar, A., Nielsen, P., Halstead, M., Bullivant, D., Nickerson, D., Hedley, W., Nelson, M., Lloyd, C., 2003. CellML 1.1 Specification. Bioengineering Institute, University of Auckland.
- Gomes, J. M., Alvarenga, A., Campos, R. S., Rocha, B. M., Silva, A. P. C., Santos, R. W., 2015. Uniformization Method for Solving Cardiac Electrophysiology Models Based on the Markov-Chain Formulation. *IEEE Transactions on Biomedical Engineering* 62 (2), 600–608.
- Guyton, A. C., 2005. Textbook of Medical Physiology, 11th Edição. Saunders.
- Hille, B., 2001. Ionic Channels of Excitable Membranes, terceira Edição. Sinauer Associates.
- Hoare, T., Milner, R., 2004. Grand Challenges in Computing. British Computing Society.
- Hodgkin, A. L., Huxley, A. F., 1952. A Quantitative Description of Membrane Current and its Application to Conduction in Nerve. *Journal of Physiology* 117, 500–544.

- Holmes, M. H., 2007. Introduction to Numerical Methods in Differential Equations. Springer.
- Iyer, V., Mazhari, R., Winslow, R. L., 2004. A Computational Model of the Human Left-Ventricular Epicardial Myocyte. *Biophysical Journal* 87, 1507–1525.
- Jafri, M. S., Rice, J. J., Winslow, R. L., 1998. Cardiac Ca^{2+} dynamics: The roles of ryanodine receptor adaptation and sarcoplasmic reticulum load. *Biophysics Journal* 74 (3), 1149–1168.
- Jenkins, G. W., Kemnitz, C. P., Tortora, G. J., 2002. Anathomy and Physiology: From science to life, segunda Edição. Lippincott Williams and Wilkins.
- Jensen, A., 1953. Markoff chains as aid in the study of Markoff processes. *Skandinavisk Aktuarietidskrift* 36, 87–91.
- Keener, J., Sneyd, J., 1998. Mathematical Physiology. Springer.
- Keener, J., Sneyd, J., 2002. Patch Clamping: An Introductory Guide to Patch Clamp Electrophysiology. Springer.
- MacLachlan, M. C., Sundnes, J., Spiteri, R. J., 2005. A comparison of non-standard solvers for ODEs describing cellular reactions in the heart. *IEEE Transactions on Biomedical Engineering* (4).
- Marsh, M. E., Ziaratgahi, S. T., Spiteri, R. J., 2012. The secrets to the success of the Rush-Larsen method and its generalizations. *IEEE Transactions on Biomedical Engineering Journal* 59 (9), 2506–2515.
- Martin, F. F., 1968. Computer modeling and simulation, primeira Edição. John Wiley and Sons.
- Melamed, B., Yadin, M., 1984. The randomization procedure in the computation of cumulative time distributions over discrete state Markov processes. *Operations Research* 32, 929–943.

- Pandit, S. V., Giles, W. R., Demir, S. S., 2003. A mathematical model of the electrophysiological alterations in rat ventricular myocytes in type-I diabetes. *Biophysics Journal* 84 (2), 832–841.
- Perego, M., Veneziani, A., 2009. An Efficient Generalization of the Rush-Larsen Method for Solving Electro-Physiology Membrane Equations. *Electronic Transaction on Numerical Analysis* 35, 234–256.
- Reibman, A., Trivedi, K., 1988. Numerical Transient Analysis of Markov Models. *Comput. Opns Res.* 15 (1), 19–36.
- Rush, S., Larsen, H., 1978. A practical algorithm for solving dynamic membrane equations. *IEEE Transactions on Biomedical Engineering* 25 (4), 389–392.
- S. Wang, S. L., Morales, M., Strauss, H., Rasmusson, R., 1997. A quantitative analysis of the activation and inactivation kinetics of hERG expressed in xenopus oocytes. *The Journal of Physiology* 502, 45–60.
- Sachse, F., 2004. *Computational Cardiology: Modeling of Anatomy, Electrophysiology and Mechanics*. Springer.
- Schram, G., Pourrier, M., Melnyk, P., S., N., 2002. Differential distribution of cardiac ion channel expression as a basis for regional specialization in electrical function. *Circ. Res.* 90, 939–950.
- Sidje, R. B., Stewart, W. J., 1996. A survey of methods for computing large sparse matrix exponentials arising in Markov chains. *Computational Statistics and Data Analysis* 29, 345–368.
- Sundnes, J., Artebrant, R., Skavhaug, O., Tveito, A., 2009. A second-order algorithm for solving dynamic cell Membrane Equations. *IEEE Transactions on Biomedical Engineering* 56, 2546–2548.
- Tusscher, K. T., Panfilov, A., 2006. Cell model for efficient simulation of wave propagation

in human ventricular tissue under normal and pathological conditions. *Physics in Medicine and Biology* 51, 6141–6156.

Winslow, R., Rice, J., Jafri, S., Marban, E., O'Rourke, B., 1999. Mechanisms of altered excitation-contraction coupling in canine tachycardia induced heart failure. *Circulation Research* 84, 571–586.

APÊNDICE A - Lista de Métodos

A Tabela A.1 lista os métodos numéricos existentes na literatura que foram utilizados ou referenciados neste trabalho. Na Tabela A.2 são listados os esquemas numéricos propostos neste trabalho, com suas características apresentadas de forma resumida.

Abreviação	Método	Referência
Euler	Euler explícito	Seção 3.1
Euler / ADP	Euler adaptativo	Seção 3.3
RL	Rush-Larsen	Seção 3.4
RK2 (ou Heun)	Runge-Kutta de 2ª ordem	Seção 3.2
SAST1	Sundnes <i>et al.</i> de 1ª ordem	Seção 3.6
SAST2	Sundnes <i>et al.</i> de 2ª ordem	Seção 3.6
UNI	Uniformização	Seção 3.5

Tabela A.1: Lista de métodos numéricos existentes na literatura.

Abreviação	Método	Referência
Euler + UNI	Híbrido Euler + uniformização	Seção 4.2.1
Euler + UNI / ADP	Híbrido Euler + uniformização com passo adaptativo	Seção 4.4.2
Euler + UNI / DTP	Híbrido Euler + uniformização com discretização temporal particionada	Seção 4.3
RL / ADP	Rush-Larsen adaptativo	Seção 4.4.1
RL / DTP	Rush-Larsen com discretização temporal particionada	Seção 4.3
RL + UNI	Híbrido Rush-Larsen + uniformização	Seção 4.2.2
RL + UNI / ADP	Híbrido Rush-Larsen + uniformização com passo adaptativo	Seção 4.4.2
RL + UNI / DTP	Híbrido Rush-Larsen + uniformização com discretização temporal particionada	Seção 4.3
RL2	Rush-Larsen de 2ª ordem	Seção 4.1.1
SAST1 / DTP	SAST1 com discretização temporal particionada	Seção 4.3
SAST1 + UNI	Híbrido SAST1 + uniformização	Seção 4.2.3
SAST1 + UNI / ADP	Híbrido SAST1 + uniformização com passo adaptativo	Seção 4.4.2
SAST1 + UNI / DTP	Híbrido SAST1 + uniformização com discretização temporal particionada	Seção 4.3
SAST2 simplificado	SAST2 com abordagem semelhante ao método de Heun	Seção 4.1.2
UNI2	Uniformização de 2ª ordem	Seção 4.1.3

Tabela A.2: Lista de esquemas numéricos propostos.