**Capítulo 1**

**1.1 - Considerações Iniciais**

O desenvolvimento de um Gêmeo Digital (*Digital Twin*, DT), cópia virtual de processos reais [1], para o controle de operação de usinas hidrelétricas envolve a representação de circuitos elétricos em software. A análise de um circuito elétrico necessita da operação matricial com custo computacional da ordem de . A medida que o número de nós do circuito aumenta, o impacto no desempenho no DT é considerável, sendo necessário a busca de alternativas para a redução do tempo computacional para realizar as operações. O tempo computacional impacta no tempo do ciclo de simulação do DT, que deve ser o menor possível, proporcionando fidelidade na simulação, nos limites do hardware disponível. Um nó ou junção de um circuito é o ponto de ligação de dois ou mais elementos do circuito. Um nó une dois ou mais condutores.

Os métodos de análise de circuitos envolvem a lei de Ohm e conceitos matemáticos como a multiplicação e inversão matricial no contexto da resolução de sistemas lineares e a solução de equações com a descrição do comportamento do circuito a partir das variáveis de tensão, corrente e resistência [2]. O equacionamento de um circuito é baseado nas Leis de Kirchhoff que descrevem o comportamento das tensões nas malhas e das correntes nos nós do circuito. Uma malha é um caminho fechado de um circuito ou qualquer caminho fechado de um condutor [3]. Os conceitos de malha e nó dividem a Lei em de Kirchhoff em outras duas Leis conhecidas como Lei das malhas de Kirchhoff (*Kirchhoff Voltage Law* - KVL) e a Lei dos nós de Kirchhoff (*Kirchhoff Current Law*, KCL) [4].

**1.2 – Análise Nodal**

A análise nodal é um procedimento geral para análise de circuitos a partir do uso de tensões dos nós como variáveis do circuito. Escolhido um nó qualquer do circuito de referência, ponto de potencial zero ou terra, os demais nós do circuito têm potência fixa em relação à referência. As interconexões têm resistência zero e todos os pontos ligados a um nó tem a mesma tensão elétrica [5].

A análise nodal é realizada nas etapas descritas a seguir. Encontrar o número de nós presentes no circuito. O número de equações necessárias para efetuar a análise do circuito. Para um circuito com *n* nós vão existir nós com um potencial fixo em relação ao nó de referência escolhido. Cada um dos nós têm uma equação associada para realizar a análise do circuito [6]. Um nó do circuito é escolhido como nó de referência e é atribuído a ele uma tensão nula ou terra. Um nó com muitos ramos é candidato a ser nó de referência. É feita a escolha de um sentido arbitrário para a corrente para cada elemento do circuito e é lhe atribuída a sua polaridade. Aplica-se a KCL a todos os nós do circuito, exceto ao nó de referência. Finalmente, as tensões dos nós são obtidas a partir da solução do sistema de equações resultantes do equacionamento de cada nó [7].

**1.3 – Análise de malhas**

A análise de malhas ou método das correntes das malhas é baseada na Lei KVL. A aplicação da análise de malhas tem como pré-requisito a planaridade do circuito, caso contrário, não é possível usar a análise de malhas. O circuito planar é desenhado em um único plano sem que dois ramos se cruzem [8]. A análise de malhas consiste nas etapas descritas a seguir. Verificar a planaridade do circuito, caso o circuito não seja planar não possível aplicar a análise de malhas. Escolher de modo arbitrário, o sentido das correntes da malha [9]. O número de correntes arbitrárias deve ser igual ao número de ramos mais o número de nós do circuito. O número de equações é igual ao número de correntes, ou seja, igual a número de malhas no circuito analisado. Uma corrente da malha deve percorrer todos os elementos do circuito, passando preferencialmente apenas uma vez em cada elemento. Identificar a polaridade da tensão a cada ramo do circuito. Finalmente, obter uma equação para cada malha percorrendo o circuito no mesmo sentido da corrente [10].

Contudo, a simulação de tais modelos gera um custo computacional elevado, o que justifica a investigação de implementações em plataformas de processamento paralelo. Neste projeto, será explorado o uso de unidade de processamento gráfico (*Graphics Processor Unit*, GPU) empregando a arquitetura de dispositivo de computação unificada (*Compute Unified Device Architecture*, CUDA), desenvolvida pela NVIDIA. Sendo utilizado através da computação paralela e uso de extensão para a linguagem de programação C, e o apoio de unidade de processamento gráfico (*Graphics Processing Unit,* GPU) [11].

**1.4 – A arquitetura CUDA**

A arquitetura CUDA incluí um *pipeline* (segmentação de instruções) unificado para permitir que cada unidade lógica e aritmética (*Arithmetic Logic Unit*, ALU) do chip seja agrupada por um programa para realizar cálculos de uso geral [12]. O NVIDIA pretendia que a nova família de processadores gráficos fosse usada para computação de uso geral [13]. As ALUs foram construídas com os requisitos de aritmética de ponto flutuante de precisão única projetadas para o uso de um conjunto de instruções personalizado para computação geral e não específico para gráficos com poder de processamento paralelo devido as motivações de seu desenvolvimento [14].

As unidades de execução na GPU ganharam acesso arbitrário de leitura e gravação a memória e acesso a um *cache* gerenciado por softwares conhecido como memória compartilhada [11]. Todos esses recursos da arquitetura CUDA foram adicionados a uma GPU para ganhar desempenho em cálculo e executar bem as tarefas gráficas tradicionais [15]. Quando comparado ao *pipeline* de um processador de dados de uma unidade central de processamento (*Central Processing Unit,* CPU) tradicional, a execução de cálculos de uso geral em uma GPU é um novo conceito [16]. As aplicações CUDA envolvem a solução de problemas em diferentes áreas como a dinâmica de fluidos, simulação de modelos climáticos, ramos da criptografia como o de cripto-moedas e traço de raios para estudos geofísicos [12].

**1.5 – Visão Geral**

Espera-se aumentar o grau de confiabilidade e ganho computacional, aumentando a assertividade geral, diminuindo-se indisponibilidades na geração de cálculos, sejam elas de natureza acidental ou programada. Vale ressaltar que a simulação dos modelos que envolvem o DT é de grande complexidade, resultando em algoritmos custosos computacionalmente e dessa forma faz-se necessário a redução do tempo computacional gastos pelo sistema. Assim é proposto que os modelos sejam paralelizados e processados em placas gráficas para redução temporal de simulação, onde o grande problema é o elevado custo computacional sendo processado sobre uma única unidade de processamento, resultando em um modelo menos eficiente.

Para o cumprimento dos objetivos propostos, a metodologia utilizada visa em explorar o uso de GPU para redução do tempo computacional envolvido na simulação do DT. Para esse propósito, emprega-se a arquitetura CUDA, que facilita o desenvolvimento de rotinas computacionais para processamento paralelo em GPUs utilizando linguagens de alto nível. Tal arquitetura tem encontrado diversas aplicações no âmbito da computação científica, podendo-se citar a solução de equações diferenciais estocásticas [17], simulações de dinâmicas moleculares [18] e modelagem da poluição do ar [19] dentre outras. Como resultado, espera-se que a taxa de crescimento do tempo computacional seja substancialmente menor na implementação usando CUDA quando comparada com uma implementação sequencial, utilizando a paralelização de algumas etapas envolvidas na simulação de um DT.

A justificativa principal deste plano de trabalho é apresentar uma alternativa à computação tradicional para a análise nodal empregando a tecnologia *GPU CUDA*. Pretende-se resolver os sistemas lineares que representam os nós dos circuitos elétricos em análise no contexto de um *DT* para o controle de operação de uma Usina Hidrelétrica ora em desenvolvimento.

**1.6 – Organização do trabalho**

Além deste capítulo, o trabalho está estruturado no Capítulo 2, onde será abordado sobre materiais e métodos e apresentados conceitos básicos relacionados a arquitetura de uma GPU; No Capítulo 3 abordaremos a programação paralela em CUDA e a multiplicação de matrizes é discutida; No Capítulo 4 abordaremos a metodologia proposta para resolução de circuito elétrico e os resultados experimentais são analisados; E no Capítulo 5 será realizada uma breve conclusão.