

GPGPU 超並列信号処理による相似地震検出及び解析の高速化 GPGPU-Accelerated Digital Signal Processing Method for Detection and Analysis of Repeating Earthquake

川上 大喜^{1*}, 大久保 寛¹, 内田 直希², 竹内 伸直², 松澤 暢²

Taiki Kawakami^{1*}, Kan Okubo¹, Naoki Uchida², Nobunao Takeuchi², Toru Matsuzawa²

¹ 首都大学東京システムデザイン研究科, ² 東北大学大学院理学研究科地震・噴火予知研究観測センター

¹Graduate School of System Design, Tokyo Metropolitan University, ²Graduate School of Science, Tohoku University

相似地震とは、プレート境界において同一のアスペリティが原因で発生する地震のことを指す。相似地震は同じ地点で観測すると、発生日時が異なってもほぼ同一な波形が得られるという特徴を有している。この相似地震を詳細に調べることにより、個々の地震を引き起こすアスペリティの特性を明らかにすることや、プレート境界でのすべりの時空間発展をモニタすることが可能となる。また、小さいアスペリティ由来の地震は短期間で繰り返し発生する傾向にあるため、数回の観測を実施することによってそのアスペリティの地震周期を特定することが可能となる。さらに、この小規模な繰り返し地震から相似地震群を特定し、そこから得られた周期を用いることでプレート境界における地震の中期的な発生予測の実現が期待できる。近年では GPU 観測網により準静的すべり域の特徴が徐々に解明されているが、十分な情報であるとは言いがたい。

先行研究により、アスペリティ及び相似地震の存在、またアスペリティと準静的すべり域の関連については多くが明らかになってきているが、相似地震を安定かつ高速に検出・解析する手法は未だ確立されているとは言えない。

本研究では、相似地震の検出法として帯域制限型フーリエ位相相関法を、また解析法として直交 3 成分地震波データを用いた振幅 2 乗コヒーレンス関数を適用してその有効性を検討した。更に、GPU (Graphics Processing Units) を用いた高速化の検討も併せて実施した。

従来、GPU は主に画像処理演算に用いられる装置であったが、近年ではその演算能力の高さから関心が集まり、GPGPU (General-Purpose computing on GPU) と呼ばれる GPU を用いた汎用計算が一般的に行われるようになってきている。GPU は CPU と比較してコア数が多い為、これら多数のコアを効果的に並列動作させた場合、GPU は CPU よりも数段高速に演算を実行することが可能となる。我々は本研究で用いる手法と GPGPU との親和性の高さに着目し、GPU による高速化を試みた。

相似地震の検出法に用いた帯域制限型フーリエ位相相関法は、2 つの信号の位相情報にのみ着目して相関値を導出する方法である。本研究では地震波の特徴であるコーナー周波数に着目して 1Hz から 10Hz の間の周波数帯域のみを利用した。相似地震波形に対して本手法を適用すると相関結果に鋭いピークを確認できる。

相似地震の解析法として適用した直交 3 成分地震波データを用いた振幅 2 乗コヒーレンス関数は、信号間のコヒーレンスを算出することで 2 つの信号間の相似度合いを周波数ごとに解析することが出来る方法である。地震波は鉛直・東西・南北方向と、物理的に直交した 3 成分の波形が観測されるため、これらを用いてコーレンス計算を行った。

CPU で処理した場合と GPU で処理した場合の計算処理時間を比較した結果、帯域制限型フーリエ位相限定相関法では約 8.0 倍、直交 3 成分コヒーレンス関数では約 12.7 倍 GPU の方が高速であった。従って、これらの相関計算に GPGPU を適用することは有意であるといえる。しかし、現行の手法は RAM と VRAM のデータ転送とファイルの I/O については、克服すべき課題がある。

今後はデータ I/O の効率化や CPU 処理と GPU 処理を並行して実行できるようなアルゴリズムの提案・実装、また演算のマルチ GPU 化を実現しさらなる高速化を検討する。

キーワード: 相似地震, GPGPU, CUDA, フーリエ位相相関, コヒーレンス関数

Keywords: Repeating Earthquake, GPGPU, CUDA, Fourier phase correlation, Coherence