



<b>Lead institution: University of São Paulo</b> <b>Work Address of the position: Av. Prof. Mello Moraes, 2231</b>	
<b>Supervisor name: Bruno Souza Carmo</b>	<b>Supervisor department: Mechanical Engineering</b>
<b>APPLY AT:</b> <a href="http://www.rcgi.poli.usp.br/opportunities/">http://www.rcgi.poli.usp.br/opportunities/</a> <a href="https://www.rcgi.poli.usp.br/opportunities-application/">https://www.rcgi.poli.usp.br/opportunities-application/</a> REF: 21PDR123	<b>Type: Post-doctoral</b> <b>Period: full time</b> <b>Number of months: 12</b>
<b>Project title</b>  Desenvolvimento de algoritmos para inversão sísmica usando o método de elementos finitos  Development of algorithms for seismic inversion employing finite element method	
<b>Research theme area</b>  Geofísica, computação de alto desempenho, inversão sísmica  Geophysics, high performance computing, seismic inversion	

## Abstract

Para a inversão de forma de onda completa (FWI) e métodos relacionados, a estimativa inicial do campo de velocidade é suave; todavia, a solução final pode conter irregularidades. No momento em que uma descontinuidade faz parte do processo de solução, as técnicas baseadas em elementos finitos, em particular Galerkin descontínuo, se mostram superiores aos métodos de diferenças finitas do ponto de vista computacional (isto é, precisão e tempo para solução), desde que as faces dos elementos de malha estejam alinhadas com as descontinuidades nas propriedades do material. Além disso, esses métodos escalam extremamente bem por causa de seu suporte compacto e os núcleos numéricos formam pequenas matrizes densas que se mapeiam bem em processadores vetoriais modernos.

Entretanto, essas técnicas podem ser complexas para aprender e paralelizar, especialmente quando melhorias na modelagem geofísica precisam ser feitas (por exemplo, desenvolver um novo propagador). Por esta razão, iremos nos concentrar no desenvolvimento dessas técnicas usando o software spyro (<https://github.com/krober10nd/spyro>), software desenvolvido pelo nosso time que utiliza o Firedrake como pacote de elementos finitos. O Firedrake é um sistema automatizado para a solução portátil de PDEs (e seus adjuntos) usando uma ampla gama de discretizações no espaço e no tempo para métodos de elementos finitos (FEM) - incluindo métodos de alta ordem e descontínuos. O Firedrake usa a Unified Form Language (UFL), um idioma específico do domínio para o método dos elementos finitos. Isso permite que os usuários desenvolvam algoritmos sofisticados e de alto desempenho e aplicações rapidamente em Python e UFL, que é o que foi feito no spyro. O objetivo deste esforço é produzir softwares de inversão de onda completa (FWI) escalável, de qualidade de produção, com elementos finitos.

For full waveform inversion (FWI) and related methods, the initial estimate of the velocity field is smooth; however, the final solution may contain irregularities. At the moment a discontinuity is part of the solution process, finite element methods techniques, in particular those based on discontinuous Galerkin discretizations, prove to be superior to computational finite difference methods (ie accuracy and time to solution), provided that the faces of the are in line with the discontinuities in the material properties. In addition, these methods scale extremely well because of their compact support and the numerical cores form small dense arrays that map well to modern vector processors. However, these techniques can be complex to learn and parallelize, especially when improvements in geophysical modeling need to be made (eg developing a new propagator). For this reason, we will focus on the development of these techniques using spyro (<https://github.com/krober10nd/spyro>), a software package developed by our team, which uses Firedrake as the finite element package. Firedrake is an automated, portable system for the solution of PDEs (and their adjuncts), using a wide range of space and time discretizations for finite element methods (FEM) - including high order and discontinuous methods. Firedrake uses the Unified Form Language (UFL), a domain-specific language for the finite element method. This allows users to develop sophisticated, high-performance algorithms and applications quickly in Python and UFL, which was the approach adopted to develop spyro. The objective of this effort is to produce scalable, production quality, finite element-based full waveform inversion (FWI) software.

## Description

No processamento sísmico de produção a maioria dos operadores de propagação baseados em ondas são discretizados usando técnicas de diferenças finitas. Sabe-se que tais técnicas sofrem de efeitos de escada degradando a precisão até a primeira ordem, independentemente da precisão inicial empregada. Para simulações ligeiramente sub-resolvidas, todos os métodos de discretização espacial são aproximadamente equivalentes em velocidade computacional para obtenção de resultados de baixa precisão. Para migração, um campo de velocidades suave é empregado, no entanto, apenas anisotropia simples pode ser modelada. Quando uma estrutura pré-sal geometricamente complexa está presente, a malha multiblocos variável para diferenças finitas pode ser complicada, dificultando ainda mais um problema já bastante desafiador do ponto de vista geofísico. Para a inversão de forma de onda completa (FWI) e métodos relacionados, a estimativa inicial do campo de velocidade é suave; todavia, a solução final pode conter irregularidades. Para diferenças finitas centradas-padrão (sem empregos de técnicas de upwind), tais discontinuidades levam ao conhecido fenômeno de Gibbs, que pode impedir a convergência para um mínimo verdadeiro, ou fazer com que muito mais iterações sejam necessárias para isso. Técnicas paliativas como funções de penalização na função de custo são atualmente empregadas, regularizando a solução e efetivamente amortecendo o efeito das oscilações dos resultados de propagação. No momento em que uma discontinuidade faz parte do processo de solução, as técnicas baseadas em Galerkin descontínuo se mostram superiores aos métodos de diferenças finitas do ponto de vista computacional (isto é, precisão e tempo para solução), desde que as faces dos elementos de malha estejam alinhadas com as discontinuidades nas propriedades do material. Além disso, esses métodos escalam extremamente bem por causa de seu suporte compacto e os núcleos numéricos formam pequenas matrizes densas que se mapeiam bem em processadores vetoriais modernos. Entretanto, essas técnicas podem ser complexas para aprender e paralelizar, especialmente quando melhorias na modelagem geofísica precisam ser feitas (por exemplo, desenvolver um novo propagador). Por esta razão, iremos nos concentrar no desenvolvimento dessas técnicas usando o software spyro (<https://github.com/krober10nd/spyro>), software desenvolvido pelo nosso time que utiliza o Firedrake como pacote de elementos finitos. O Firedrake é um sistema automatizado para a solução portátil de PDEs (e seus adjuntos) usando uma ampla gama de discretizações no espaço e no tempo para métodos de elementos finitos (FEM) - incluindo métodos de alta ordem e descontínuos. O Firedrake usa a Unified Form Language (UFL), um idioma específico do domínio para o método dos elementos finitos. Isso permite que os usuários desenvolvam algoritmos sofisticados e de alto desempenho e aplicações rapidamente em Python e UFL, que é o que foi feito no spyro. O objetivo deste esforço é produzir softwares de inversão de onda completa (FWI) escalável, de qualidade de produção, com elementos finitos.

In seismic production processing most wave propagation operators are discretized using finite difference techniques. It is known that such techniques suffer from ladder effects by degrading accuracy to the first order, regardless of the initial precision employed. For slightly subresolved simulations, all spatial discretization methods are approximately equivalent in computational speed to obtain low accuracy results. For migration, a smooth velocity field is employed, however, only simple anisotropy can be modeled. When a geometrically complex pre-salt structure is present, the variable multi-block mesh for finite differences can be complicated, making even more challenging a problem from the geophysical point of view. For full waveform inversion (FWI) and related methods, the initial estimate of the velocity field is smooth; however, the final solution may contain irregularities. For standard-centered finite differences (without employing upwind techniques), such discontinuities lead to the well-known Gibbs phenomenon, which may prevent convergence to a true minimum, or cause much more iterations to be necessary for this. Palliative techniques such as penalty functions in the cost function are currently employed, regulating the solution and effectively

cushioning the effect of the oscillations of the propagation results. At the moment a discontinuity is part of the solution process, discontinuous Galerkin-based techniques prove to be superior to computational finite difference methods (ie accuracy and time to solution), provided that the faces of the are in line with the discontinuities in the material properties. In addition, these methods scale extremely well because of their compact support and the numerical cores form small dense arrays that map well to modern vector processors. However, these techniques can be complex to learn and parallelize, especially when improvements in geophysical modeling need to be made (eg developing a new propagator). For this reason, we will focus on the development of these techniques using spyro (<https://github.com/krober10nd/spyro>), a software package developed by our team, which uses Firedrake as the finite element package. Firedrake is an automated, portable system for the solution of PDEs (and their adjuncts), using a wide range of space and time discretizations for finite element methods (FEM) - including high order and discontinuous methods. Firedrake uses the Unified Form Language (UFL), a domain-specific language for the finite element method. This allows users to develop sophisticated, high-performance algorithms and applications quickly in Python and UFL, which was the approach adopted to develop spyro. The objective of this effort is to produce scalable, production quality, finite element-based full waveform inversion (FWI) software.

#### **Requirements to fill the position**

Doutorado em geofísica, engenharia, ciência da computação ou matemática aplicada. Habilidades avançadas de programação em Python e conhecimento de computação de alto desempenho. Desejável experiência em técnicas de inversão sísmica.

PhD in Geophysics, Engineering, Computing Science or Applied Mathematics. Advanced techniques in Python programming and knowledge in high-performance computing. Experience in seismic inversion techniques is desirable.

#### **INFORMAÇÕES SOBRE A BOLSA:**

O candidato selecionado receberá bolsa de R\$ 6.819,30 reais mensais, concedida pela FUSP - Fundação de Apoio à Universidade de São Paulo. MAIORES INFORMAÇÕES <http://www.rcgi.poli.usp.br/opportunities> e INSCRIÇÃO em <https://www.rcgi.poli.usp.br/opportunities/application-form-rcgi/> REF 21PDR123

#### **INFORMATION ABOUT FELLOWSHIP:**

The selected candidate will receive a scholarship of R\$ 6.819,30 reais monthly granted by FUSP - Foundation of Support to the University of São Paulo. MORE INFORMATION <http://www.rcgi.poli.usp.br/opportunities> AND APPLICATION AT REF <https://www.rcgi.poli.usp.br/opportunities/application-form-rcgi/21PDR123>