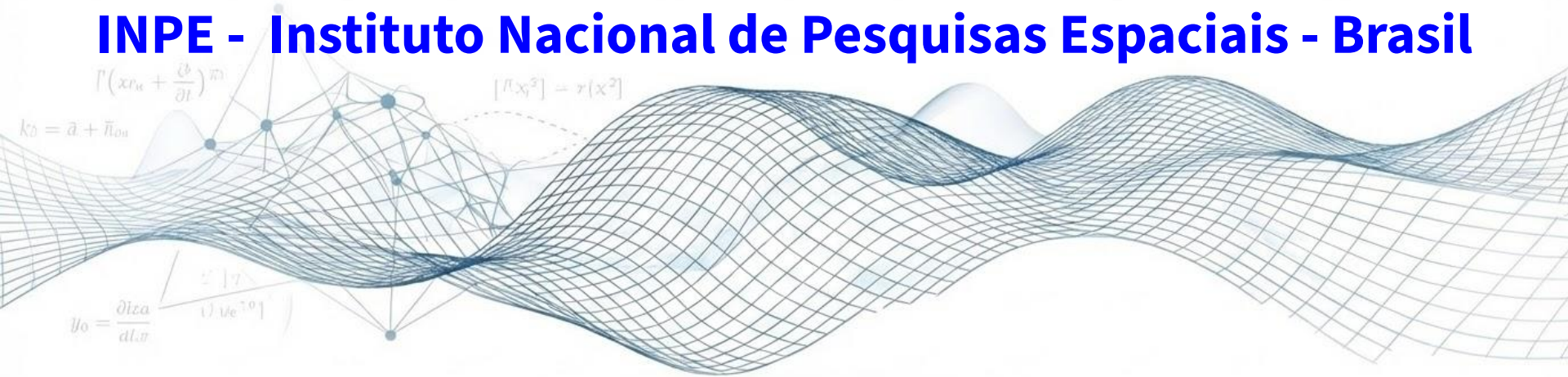


Descoberta de Parâmetro na Equação 2D de Burgers por Rede Neural Informada pela Física

Autores: Eduardo F. Miranda, Roberto P. Souto, Stephan Stephany

INPE - Instituto Nacional de Pesquisas Espaciais - Brasil



CIACA 2025 (#66)

Aprendizado com Rede Neural Informada pela Física

- **Modelo Especialista ("Caso Base")**

- Resolve o problema p/ único valor inicial de v (“educated guess”)
- Novo experimento c/ outro valor inicial exige retreinamento (~9 min)
- Treinamento se perde se o valor inicial for muito diferente do real
- Modelo gerado infere então o valor da viscosidade v

- **Modelo Generalista ("Caso Atual")**

- Treinamento com vários datasets gerados com viscosidades diferentes dentro de uma faixa discreta (ex.: 0,01 - 0,10)
- Modelo gerado infere então o valor da viscosidade v

O Problema Inverso da Equação de Burger's 2D

- **Burgers 2D é uma EDP (sistema acoplado e não-linear), onde a estimativa de parâmetro ν (problema inverso) é mais instável que a solução da própria Equação de Burger's 2D (problema direto), onde ν é a viscosidade cinemática do fluido**

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = \nu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right)$$

$$\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} = \nu \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right)$$

- **Em geral novas arquiteturas PINN têm sido propostas, geralmente para problemas diretos**
- **Questões críticas como estouro de memória, otimização, e comparações como Adam x L-BFGS-B são frequentemente omitidas**

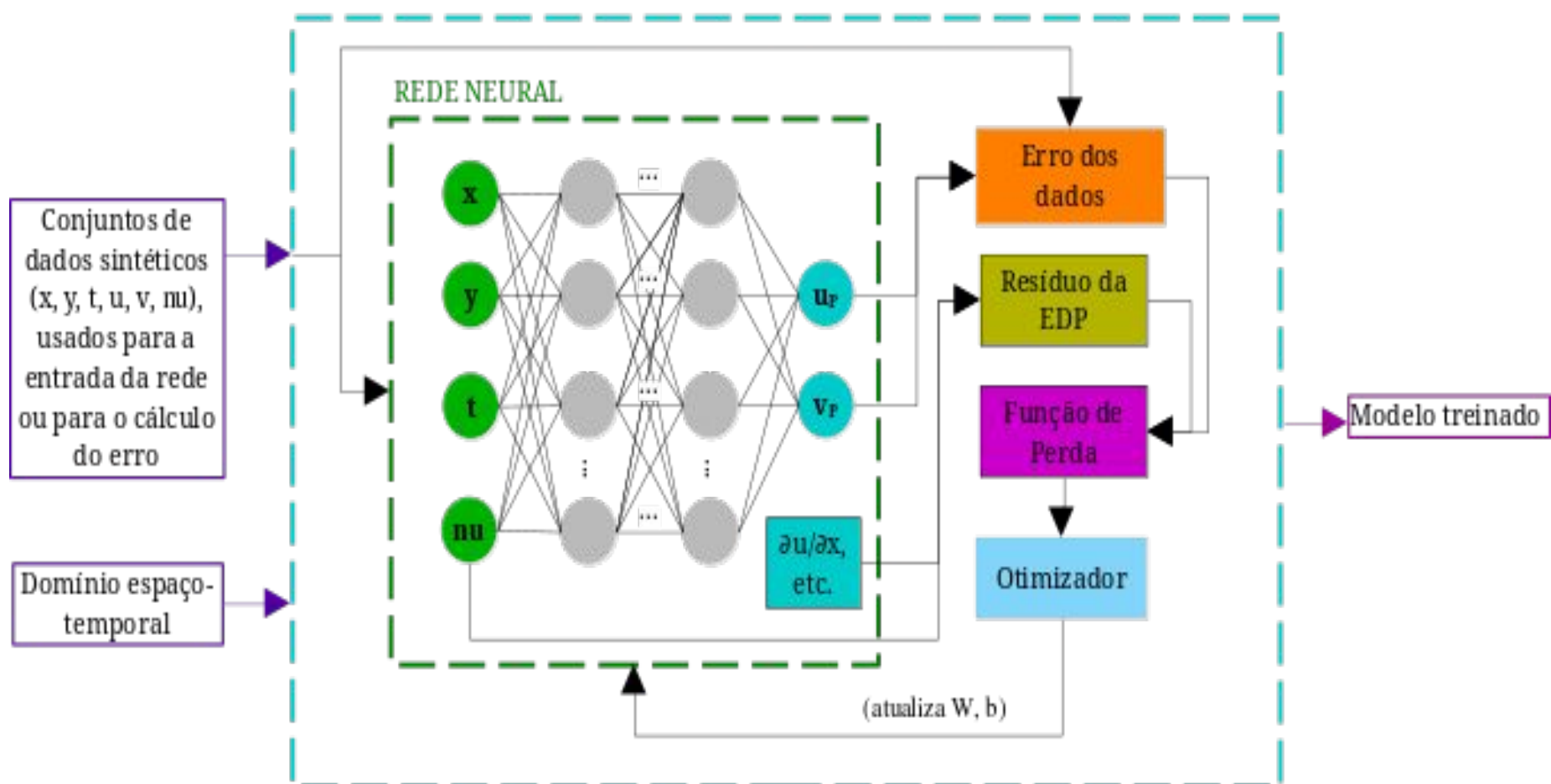
Redes Neurais Informadas pela Física (PINNs)

- **Implementação em TensorFlow, execução em PC com GPU Nvidia RTX3050**
- **PINNs supostamente resolvem EDP's a um custo computacional muito menor do que métodos numéricos tradicionais, cuja grade de resolução implica em muito processamento**
- **PINN integra a EDP governante na função de perda (PDEloss)**
 - Não depende de malhas de discretização de alta resolução
 - Reduz o custo de simulações de larga escala
- **Otimização Híbrida: Adam (busca global) + L-BFGS-B (busca local)**
 - A busca local via L-BFGS-B da solução v permite que o domínio de busca seja limitado com recurso do próprio otimizador (“B”)

Treinamento - Caso Atual - Estágio 1

(pesos/biases atualizados usando dataset velocidades sem ainda inferir a viscosidade)

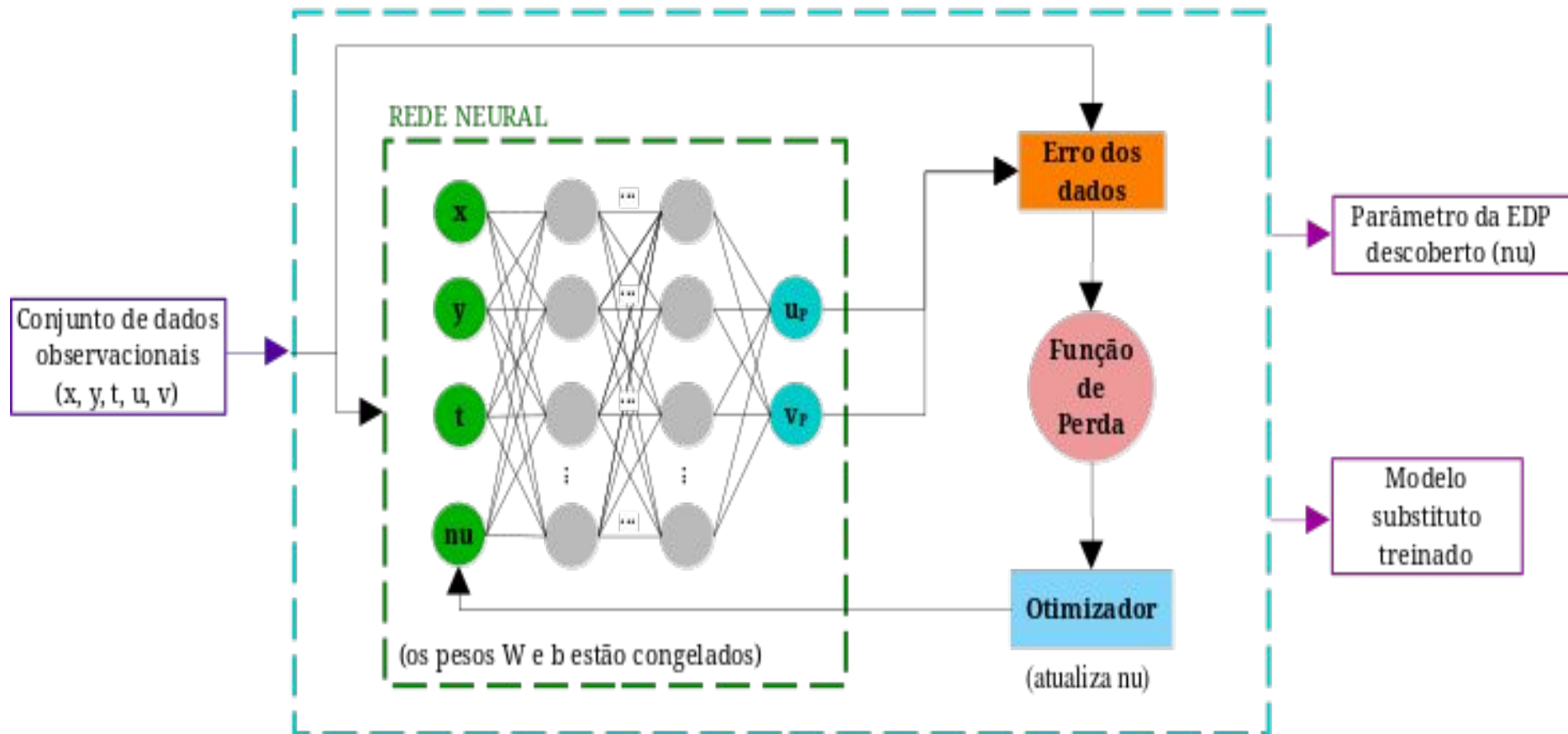
ESTÁGIO 1 - Treinamento do Modelo Substituto Paramétrico



Treinamento - Caso Atual - Estágio 2

(pesos/biases “congelados” e inferência da viscosidade)

ETAPA 2 - PINN (Problema Inverso)



Função de perda L - Caso Base e Caso Atual

$$L_{\text{total}} = \lambda_{\text{data}} \cdot L_{\text{data}} + \lambda_{\text{pde}} \cdot L_{\text{pde}}$$

Onde:

L_{total} : valor da perda que o otimizador minimiza

λ_{data} : peso adaptativo da Perda de Dados

L_{data} : erro das previsões em relação aos dados de simulação

λ_{pde} : peso adaptativo da Perda da EDP

L_{pde} : erro das previsões em relação à EDP

Perda da Física - termo EDP (L_{pde})

$$L_{pde} = MSE_{f_u} + MSE_{f_v}$$

Onde:

$$MSE_{f_u} = (1/N_{pde}) \cdot \sum [w_v \cdot f_u(u_{previsto})]^2$$

$$MSE_{f_v} = (1/N_{pde}) \cdot \sum [w_v \cdot f_v(v_{previsto})]^2$$

w_v : peso do Aprendizado Curricular

f_u & f_v : equações residuais

Mecanismos de Estabilização do Treinamento

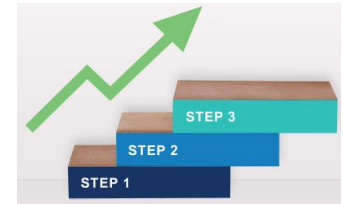
- **Ponderação Adaptativa**

- Pesos λ_{data} & λ_{pde} são ajustados continuamente provendo balanceamento



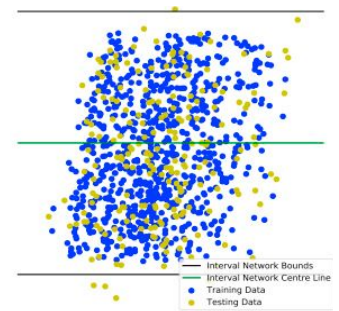
- **Aprendizado Curricular**

- espaço de busca da viscosidade v vai sendo gradualmente aumentado



- **Regularização Baseada em Intervalos**

- Limita a busca dentro de um intervalo fisicamente plausível (usado apenas no Estágio 2)



Caso Base: Treinamento em Dois Estágios

- **Estágio 1 - Pré-treinamento somente c/ termo *Dataloss***
 - Ajusta pesos e biases (W, b) da rede usando campo de velocidades sintético gerado com uma dada viscosidade desconhecida para a PINN
- **Estágio 2 - Treinamento principal *Dataloss* + *PDEloss***
 - Utiliza a rede pré-treinada (W, b fixos) e a função de perda composta (data loss + PDE loss) para estimação de v

Caso Atual: Treinamento em Três Estágios

- **Estágio 1.A - Pré-treinamento c/ termo *Dataloss***
 - Ajusta pesos e biases (W, b) usando múltiplos conjuntos de dados, cada um para uma viscosidade discreta
- **Estágio 1.B - Treinamento principal c/ *Dataloss* + *PDEloss***
 - Reajusta (W, b) p/ modelo aprender a mapear as entradas (x, y, t, v) para as saídas (u, v) usando múltiplos conjuntos de dados
- **Estágio 2 - Descoberta de v (Problema Inverso)**
 - (W, b) são congelados e o otimizador estima v minimizando apenas ao termo *Dataloss* (termo *PDEloss* não é usado)

Dataset de Treinamento

- **Método de Diferenças Finitas (FDM) é usado para gerar dados sintéticos do campo de velocidades em diferentes instantes**
- **Dimensão Espacial (Grid)**
 - `grid_points_x`: 41 pontos no eixo x
 - `grid_points_y`: 41 pontos no eixo y
 - Grade 2D com $41 * 41 = 1.681$ pontos
- **Dimensão Temporal (Snapshots)**
 - 4 snapshots: $t=0,012s$; $t=0,025s$; $t=0,037s$ e $t=0,050s$
 - $1.681 \text{ grid} \times 4 \text{ snapshots} = 6.724$ pontos de dados
- **Cada ponto = (x, y, t, v, u, v)**

Otimização de Hiperparâmetros

- **Caso Base: reflete o modelo especialista do artigo**
- **Caso Atual: rede PINN foi refinada c/ a biblioteca Hyperopt, para otimização dos hiperparâmetros e arquitetura da rede**

Parâmetro	Caso Base	Caso Atual
Arquitetura da Rede	5 camadas x 60 neurônios	4 camadas x 50 neurônios
Estratégia de Treino	2 Estágios	3 Estágios (500 / 5k / 2k épocas)
Taxa de Aprendizado	1e-3	Otimizada para 2,3e-4
Tolerância a Ruído	Não otimizado	Otimizado para 4% de ruído

Diagnóstico e Solução de Gargalo de Memória

- **Execução limitada pela memória (Memory-Bound), não pela capacidade de cálculo (Compute-Bound)**
 - Latência de transferência CPU-GPU dominava o processo
 - Uso da ferramenta NVIDIA Nsight Compute (NCU)
- **Redução do batch size de 20.000 para 4.096**
 - Aumento de ~16% na taxa de acerto do Cache L2
 - Redução de ~67% no tráfego total entre as memórias (CPU-GPU)
 - Redução estimada de ~2x no tempo de treinamento da Etapa 1, viabilizando a escalabilidade relativa aos dados

Otimização das Derivadas de 2ª ordem

- **Falhas sistemáticas de “Out of Memory” ao calcular derivadas de segunda ordem (u_{xx} , u_{yy})**
 - Retenção excessiva do grafo computacional pelo `tf.GradientTape(persistent=True)`, esgotando VRAM/GPU
 - Necessário para calcular e reter o cálculo das derivadas usando o mesmo “tape” do TensorFlow
- **Implementação de estrutura de Tapes Aninhados incluindo gerenciamento de memória e liberação explícita de recursos**
 - Uso de memória do algoritmo foi otimizado ocorrências de Out of Memory para um consumo de memória gerenciável
 - Permitiu uso de batch sizes maiores ou mais pontos de colocação

Quantidade de *Datasets* de Treinamento

- **Fusão e carregamento simultâneo de todos os datasets levou a estouro de memória e consequente carga de 3 datasets de cada vez resultou em subajuste (underfitting)**
- **Geração de 15 datasets, utilizados alternadamente ao longos das épocas do treinamento da rede, por (i) seleção aleatória (erro alto) ou por (ii) Hipercubo Latino (diminui erro)**

Caso Base: Prova de Conceito

- Rede específica treinada para resolver apenas uma viscosidade (ex.: $\nu=0,05$ ou $\nu=0,02$)
- Descoberta de ν com erros relativos consistentemente baixos (0,0% a 1,7%) em vários experimentos
- Funcional, mesmo com ruído gaussiano (erro de 0,33% com 10% de ruído)
- Convergência rápida, cerca de ~9 minutos por experimento
- A generalização para uma faixa mais ampla de valores de ν ficou como um desafio para investigação futura (no Caso Base)

Caso Atual: Modelo Substituto Paramétrico

- Estágio 1: a rede resolve o problema direto $f(x, y, t, v) \rightarrow (u, v)$
- Estágio 2: descoberta de v (problema inverso) por retreinamento exclusivo da viscosidade, sem alterar pesos e biases
- Treinamento com os 15 datasets com v variados
- Uso de conjunto separado validação, inferindo v em um dataset "*hold-out*" inédito ($v_{\text{true}}=0,0382$)
- Obtenção de estabilidade numérica e gestão crítica de memória

Amostragem dos Datasets no Treinamento da Etapa 1: Aleatória e por Hipercubo Latino (LHS)

- **Caso Atual (Amostragem Aleatória)**

- Seleção puramente estocástica
- Gera *clusters* e deixa lacunas no espaço paramétrico de viscosidade
- Resultado: alta instabilidade
 - Erro Médio: 46,45%
 - Pior Caso: 84,01%

- **Caso Atual (LHS)**

- Amostragem estratificada (divisão do domínio em intervalos equiprováveis)
- Garante que a rede “veja” exemplos de toda a faixa de viscosidades, sem repetições próximas
- Resultado: maior robustez e convergência
 - Erro Médio: 20,56%
 - Redução de ~2,3x no erro médio

Custos de Treinamento e Inferência

- Implementação em TensorFlow em PC com GPU Nvidia RTX3050
- Caso Atual
 - Etapa 1.A treinada em ~1,5 minutos
 - Etapa 1.B treinada em ~16 minutos
 - A inferência na Etapa 2 é feita em ~73 segundos e pode ser reusada
 - O maior custo de treinamento é compensado pela eliminação do retreinamento
- Caso Base
 - Etapa 1 é treinada em ~7,5 minutos
 - Etapa 2 (problema inverso) é feita em ~46 segundos
 - Necessário retreinamento para cada viscosidade

Trabalhos Futuros

- Investigar a complexidade do modelo para problemas de maior dimensão, como a Equação de Burgers 3D
- Aplicar estratégias Multi-GPU/Multi-Node para reduzir o tempo de treinamento e aumentar a complexidade (supercomputador SD)
- Explorar a otimização de GPU para aumento de desempenho (*profiling* no SD)

Obrigado! Perguntas, por favor?

- **Autores**

- Eduardo F. Miranda - Instituto Nacional de Pesquisas Espaciais (INPE)
- Roberto P. Souto - Laboratório Nacional de Computação Científica (LNCC)
- Stephan Stephany - Instituto Nacional de Pesquisas Espaciais (INPE)

- **Implementação**

- <https://github.com/efurlanm/burgers-pinn>

- **Agradecimentos**

- Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, Brasil (CAPES)
- LNCC (Laboratório Nacional de Computação Científica) pelo acesso ao supercomputador Santos Dumont por meio do projeto AMPEMI
- Ministério da Ciência, Tecnologia e Inovação e à Agência Espacial Brasileira