Newton's Method for PDE II

Ryan McPeck

Northern Arizona University

April 13, 2010

K ロ ⊁ K 倒 ≯ K ミ ⊁ K ミ ⊁

重

[Outline](#page-1-0)

[A Quick Review](#page-2-0) [Newton's Method](#page-3-0) [Overview of Programming and Results](#page-7-0) [Acknowledgments](#page-12-0)

[Outline](#page-1-0)

[A Quick Review](#page-2-0)

[Newton's Method](#page-3-0)

[Standard Form](#page-3-0) [Newton's Method in Higher Dimensions](#page-4-0) [A Useful Theorem](#page-5-0)

[Overview of Programming and Results](#page-7-0)

[Programming](#page-7-0) [Modifying Newton's Method Further](#page-8-0) [Results](#page-10-0)

[Acknowledgments](#page-12-0)

|④ 重 B ④ 重 B

General Information

- \triangleright A partial differential equation, or PDE, is a differential equation involving partial derivatives.
- \blacktriangleright PDEs have many applications:
	- \blacktriangleright Harvesting
	- \blacktriangleright Starburst
	- \blacktriangleright Heat
	- \triangleright Waves
- ► Specifically we studied $\Delta u + su + u^3 = 0$ on the unit disk centered at the origin in \mathbb{R}^2 with $u = 0$ on the boundaries.

 \leftarrow \cap \rightarrow \leftarrow \cap \rightarrow

 Ω

[Standard Form](#page-3-0) [Newton's Method in Higher Dimensions](#page-4-0) [A Useful Theorem](#page-5-0)

Standard Form

$$
\blacktriangleright x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}
$$

Ryan McPeck [Newton's Method for PDE II](#page-0-0)

メロメ メ都 メメ きょくきょう

重

 299

[Standard Form](#page-3-0) [Newton's Method in Higher Dimensions](#page-4-0) [A Useful Theorem](#page-5-0)

Newton's Method in Higher Dimensions

- \blacktriangleright f : $\mathbb{R}^m \to \mathbb{R}^m$ an arbitrary function
- \triangleright $a_{k+1} = a_k (J_f(a_k)))^{-1} f(a_k)$
	- \blacktriangleright J_f is a Jacobian of the function f , or a vector of first derivatives

K ロ ⊁ K 倒 ≯ K ミ ⊁ K ミ ⊁

[Standard Form](#page-3-0) [Newton's Method in Higher Dimensions](#page-4-0) [A Useful Theorem](#page-5-0)

Theorem

- Let $J: X \to \mathbb{R}$ (Recall that X is the function space that we are working in)
- $\triangleright \nabla J(u) = 0 \Longleftrightarrow u$ is a solution to the pde

$$
\triangleright \quad J(u) = \int \frac{|\nabla J(u)|^2}{2} - F(u) d\hat{x}
$$

$$
\triangleright \quad F(u) = \frac{su^2}{2} + \frac{u^4}{4} = \int_0^u f(s) ds
$$

K ロ ⊁ K 倒 ≯ K ミ ⊁ K ミ ⊁

 $2Q$

目

[Standard Form](#page-3-0) [Newton's Method in Higher Dimensions](#page-4-0) [A Useful Theorem](#page-5-0)

Proof in One Direction

Suppose $u(x)$ is a solution to $u'' + u^3 = 0$. Then $\int (u'' + u^3)v dx = 0$ for all v. \blacktriangleright $J(u) = \int \frac{|\nabla J(u)|^2}{2}$ $\frac{|u|^2}{2} - \frac{u^4}{4}$ $\frac{1}{4}d\hat{x}$ ► $J'(u)(v) = \int u'v' - u^3v = -\int (u'' + u^3)v = -\langle u'' + u^3, v \rangle$ ► By definition, $\nabla J(u) = -(u'' + u^3) = 0$.

メロメ メ御 メメミメメミメ

へのへ

[Programming](#page-7-0) [Modifying Newton's Method Further](#page-8-0) [Results](#page-10-0)

Programming

- \blacktriangleright Language: MATLAB
- \blacktriangleright Goals:
	- Find roots of $J(u) = \int \frac{|\nabla J(u)|^2}{2}$ $\frac{(-1)^n}{2} - F(u) d\hat{x}$ using Newton's Method
	- \triangleright Display and interpret results

イロト イ部 トイモ トイモト

重

[Programming](#page-7-0) [Modifying Newton's Method Further](#page-8-0) [Results](#page-10-0)

Gradient Newton Galerkin Algorithm

- \triangleright Mentor and colleagues have successfully used modified Newton's Method to find solutions for PDE
- \triangleright Our particular PDE of interest was studied by a student working on his master's thesis

$$
\blacktriangleright a^{k+1} = a^k - \left(\frac{\partial g_i}{\partial a_j}\right)^{-1} g(a^k)
$$

- I ∂gⁱ $\frac{\partial \mathbf{S}^{\prime}}{\partial \mathbf{a}_{j}}$ is the Hessian, a matrix filled with second derivatives
- \blacktriangleright g is the gradient that we are trying to find roots of

K ロ ト K 御 ト K ヨ ト

∽≏ດ

[Programming](#page-7-0) [Modifying Newton's Method Further](#page-8-0) **[Results](#page-10-0)**

Filling in the Gradient and the Hessian

$$
\blacktriangleright g_i = J'(u)(\Psi_i) = a_i \lambda_i - \int f(u)\Psi_i
$$

$$
\blacktriangleright h_{ij} = J''(u)(\Psi_i, \Psi_j) = \lambda_i \delta_{ij} - \int f'(u)\Psi_i \Psi_j
$$

- \blacktriangleright Integrals were computed numerically with Riemann sums ► Solve: $\frac{\partial g_i}{\partial x}$ $\frac{\partial S_i}{\partial a_j} \cdot \chi = g$ to avoid calculating the inverse of the Hessian
- \triangleright Tolerances and iteration limits imposed to find and save off solutions

メロメ メ御 メメ ミメ メミメ

 Ω

[Programming](#page-7-0) [Modifying Newton's Method Further](#page-8-0) **[Results](#page-10-0)**

Bifurcation Diagrams

イロト イ押 トイモト イモト

ă

 299

[Programming](#page-7-0) [Modifying Newton's Method Further](#page-8-0) **[Results](#page-10-0)**

What These Solutions Look Like

- \triangleright Solutions can be reassembled using linear combinations
- \triangleright Sadly, we have not quite figured out the easiest way to do this yet
- \triangleright Recall graphics from last presentation

 \leftarrow \cap \rightarrow \leftarrow \cap \rightarrow

 $\epsilon = 1$ i al T≣ira

Acknowledgments

- ▶ Dr. John Neuberger, mentor
- \triangleright Mr. Jeff Rushall, for all the opportunities to practice presenting
- ▶ NAU NASA Space Grant program
- \triangleright University of Arizona and conference organizers

 $4.17 \times$

 \rightarrow \overline{m} \rightarrow

 \leftarrow \equiv \rightarrow