Emergent Numerical Algorithms

D. M. Bortz

Applied Mathematics

University of Colorado

Boulder, CO

March 21, 2013

GPU Tech Conference

San Jose, CA

FA9550-09-1-0403

Outline

1. Introduction
   - Goals
   - Plan
2. Progress
   - Core algorithm
   - Analysis
   - Results
3. Summary

Motivation

- Flops are free

Motivation

- Flops are free
- Massively multi-core architecture (SIMD)

Motivation

- Flops are free
- Massively multi-core architecture (SIMD)
- (Relatively) modest memory: 6 Gig on C2070

Motivation

- Flops are free
- Massively multi-core architecture (SIMD)
- (Relatively) modest memory: 6 Gig on C2070
- Hierarchical Memory (PBSM, texture, constant, global)

Motivation

- Flops are free
- Massively multi-core architecture (SIMD)
- (Relatively) modest memory: 6 Gig on C2070
- Hierarchical Memory (PBSM, texture, constant, global)
- Re-imagine algorithms for this environment

Motivation

- Flops are free
- Massively multi-core architecture (SIMD)
- (Relatively) modest memory: 6 Gig on C2070
- Hierarchical Memory (PBSM, texture, constant, global)
- Re-imagine algorithms for this environment
- Statistical Inference of mesh point locations.

Motivation

- Flops are free
- Massively multi-core architecture (SIMD)
- (Relatively) modest memory: 6 Gig on C2070
- Hierarchical Memory (PBSM, texture, constant, global)
- Re-imagine algorithms for this environment
- **Statistical Inference of mesh point locations.**

Non-Uniform Mesh Generation Example

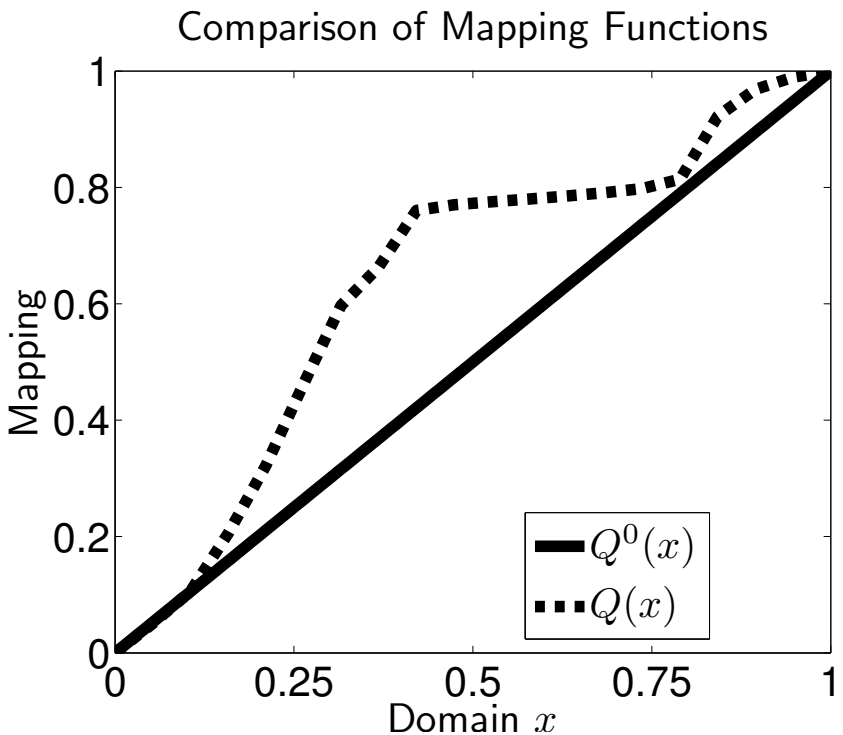
- Identifying optimal change of independet variable $Q^*(x)$ such that approximate solution to PDE has convergence properties superior to $Q(x) \equiv x$.
- Poisson equation example:

$$u''(Q^*(x)) \quad = \quad f(Q^*(x))$$

Non-Uniform Mesh Generation Example

- Identifying optimal change of independet variable $Q^*(x)$ such that approximate solution to PDE has convergence properties superior to $Q(x) \equiv x$.
- Poisson equation example:

$$u''(Q^*(x)) \quad = \quad f(Q^*(x))$$



Non-Uniform Mesh Generation Example

- Identifying optimal Q^* such that approximate solution to

$$u''(Q^*(x)) \quad = \quad f(Q^*(x))$$

  have convergence properties superior to $Q(x) \equiv x$.
- Cast $P(x) = Q^{-1}(x)$ as a probability measure

Non-Uniform Mesh Generation Example

- Identifying optimal Q^* such that approximate solution to

$$u''(Q^*(x)) \quad = \quad f(Q^*(x))$$

  have convergence properties superior to $Q(x) \equiv x$.
- Cast $P(x) = Q^{-1}(x)$ as a probability measure
- Identify P^* via sparse sampling:

Non-Uniform Mesh Generation Example

- Identifying optimal Q^* such that approximate solution to

$$u''(Q^*(x)) \quad = \quad f(Q^*(x))$$

  have convergence properties superior to $Q(x) \equiv x$.
- Cast $P(x) = Q^{-1}(x)$ as a probability measure
- Identify P^* via sparse sampling:
  - **Sample n small from a candidate P, m times.**

Non-Uniform Mesh Generation Example

- Identifying optimal Q^* such that approximate solution to

$$u''(Q^*(x)) \quad = \quad f(Q^*(x))$$

  have convergence properties superior to $Q(x) \equiv x$.
- Cast $P(x) = Q^{-1}(x)$ as a probability measure
- Identify P^* via sparse sampling:
  - **Sample n small from a candidate P, m times.**
  - Perform m solves of coarse solutions (discretization order n)

Non-Uniform Mesh Generation Example

- Identifying optimal Q^* such that approximate solution to

$$u''(Q^*(x)) \quad = \quad f(Q^*(x))$$

  have convergence properties superior to $Q(x) \equiv x$.
- Cast $P(x) = Q^{-1}(x)$ as a probability measure
- Identify P^* via sparse sampling:
  - **Sample n small from a candidate P, m times.**
  - Perform m solves of coarse solutions (discretization order n)
  - Compute pointwise statistical moments of the coarse solutions

Non-Uniform Mesh Generation Example

- Identifying optimal Q^* such that approximate solution to

$$u''(Q^*(x)) \quad = \quad f(Q^*(x))$$

  have convergence properties superior to $Q(x) \equiv x$.
- Cast $P(x) = Q^{-1}(x)$ as a probability measure
- Identify P^* via sparse sampling:
  - **Sample n small from a candidate P, m times.**
  - Perform m solves of coarse solutions (discretization order n)
  - Compute pointwise statistical moments of the coarse solutions
  - Improve candidate P?

Plan

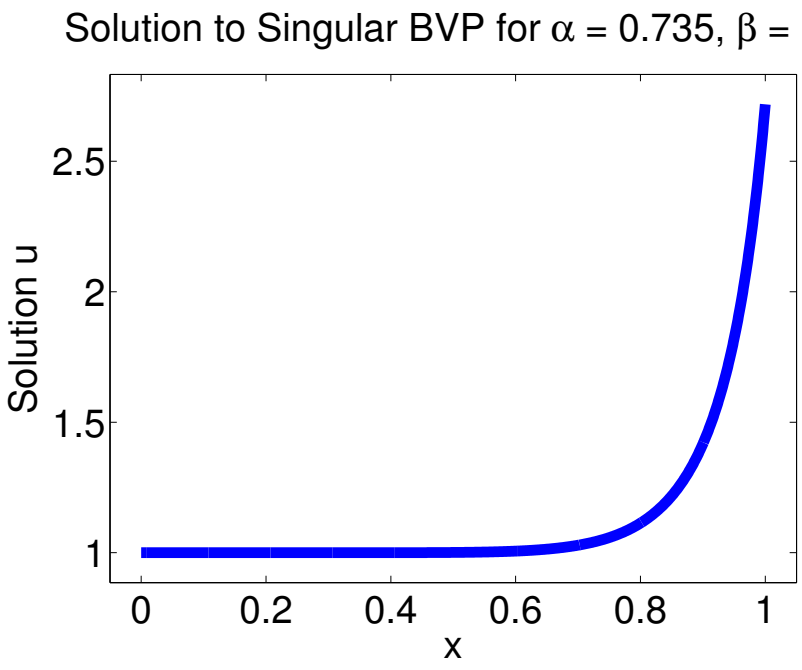
1. Core algorithm
2. Analysis
3. Test problems
4. Results

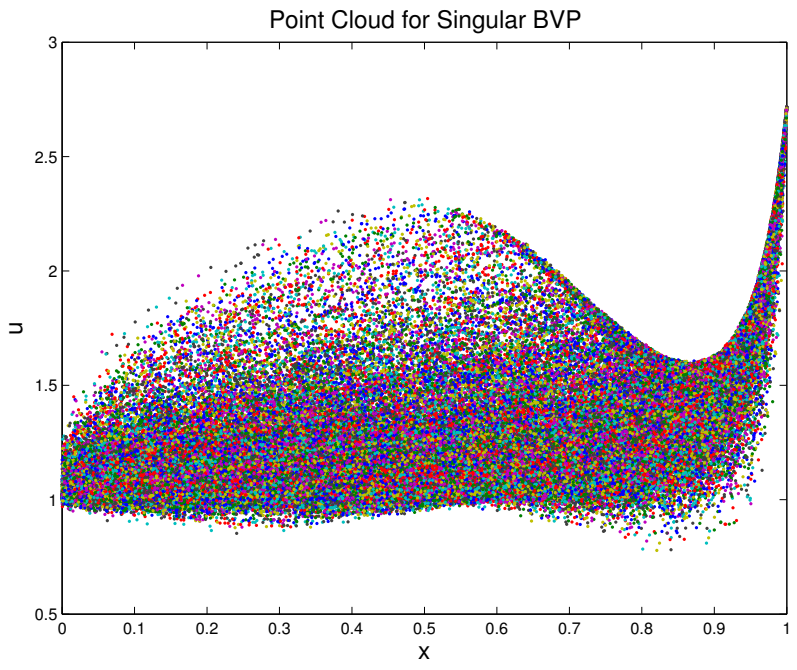
1D mesh generation: DMB & Christlieb, *SIAM J. Sci. Comp.*, in revision.
Theory: DMB & Byrne, *Inverse Problems*, in revision.
2D mesh generation: DMB & Christlieb, in prep.

- All simulations performed on a Tesla C2070 using matlab's GPU toolbox (**arrayfun**).

Core Algorithm

Sampling and Solving



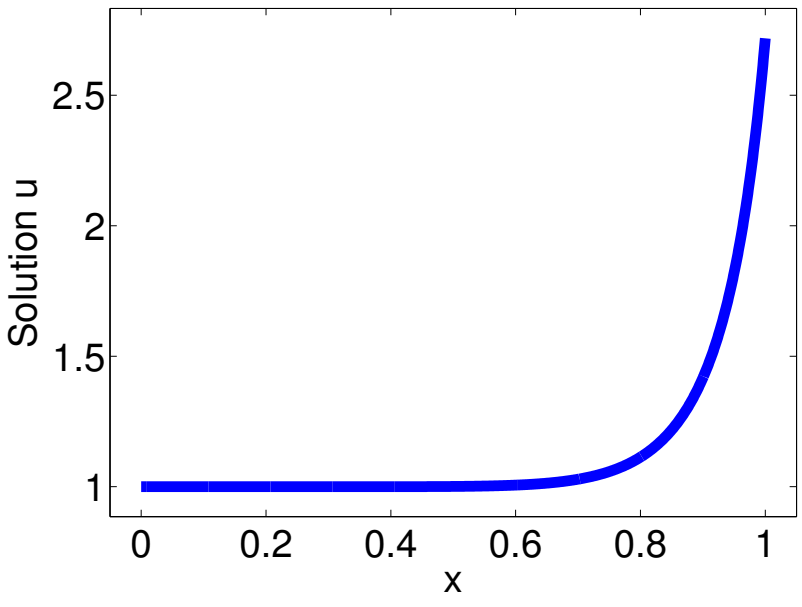


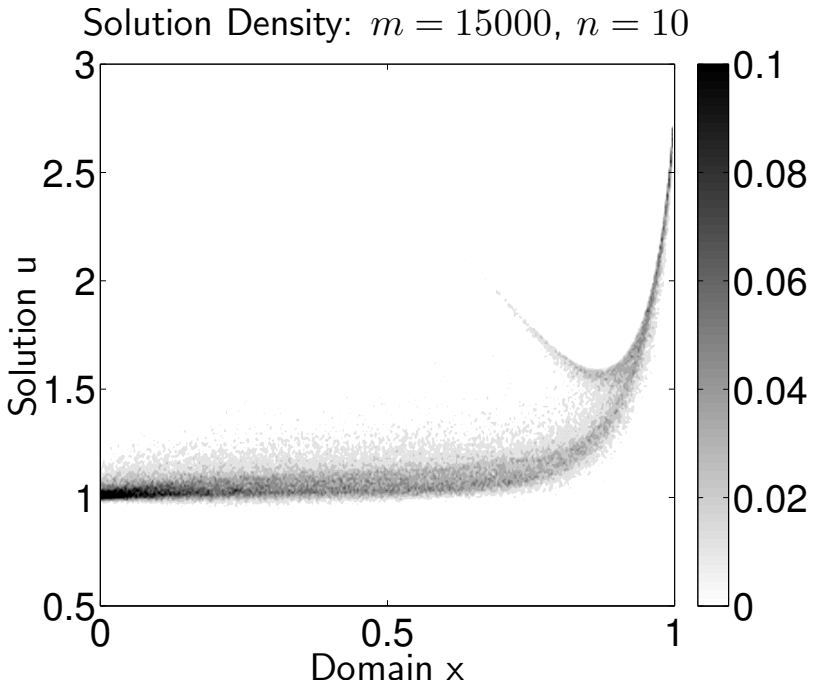
$$(x^{\alpha}u')' = \beta x^{\alpha+\beta-2}((\alpha+\beta-1)+\beta x^{\beta})u \qquad ; 0 \leq x \leq 1$$

$$u(0) = A\,,\; u(1) = B$$

$$Q^*(x) = x^{1-\alpha}$$

Sampling and Solving



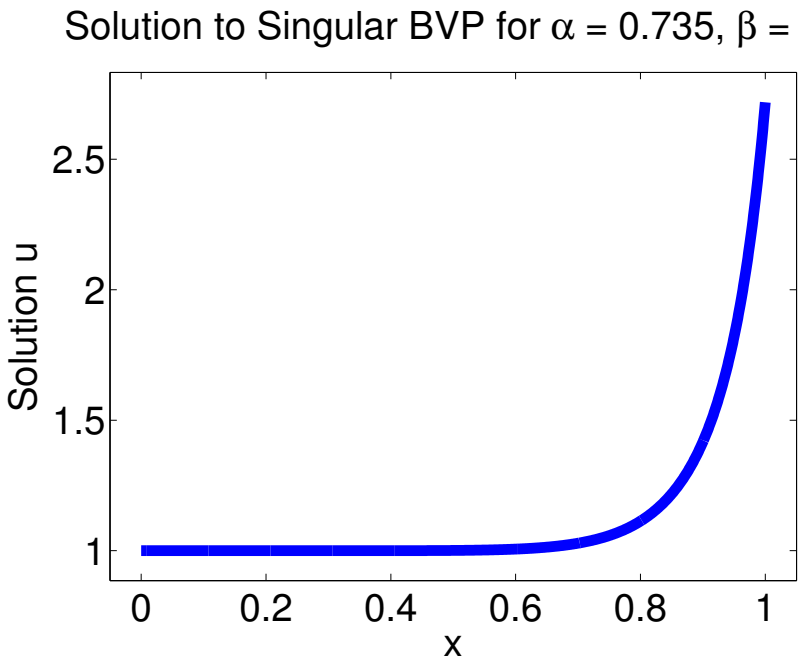


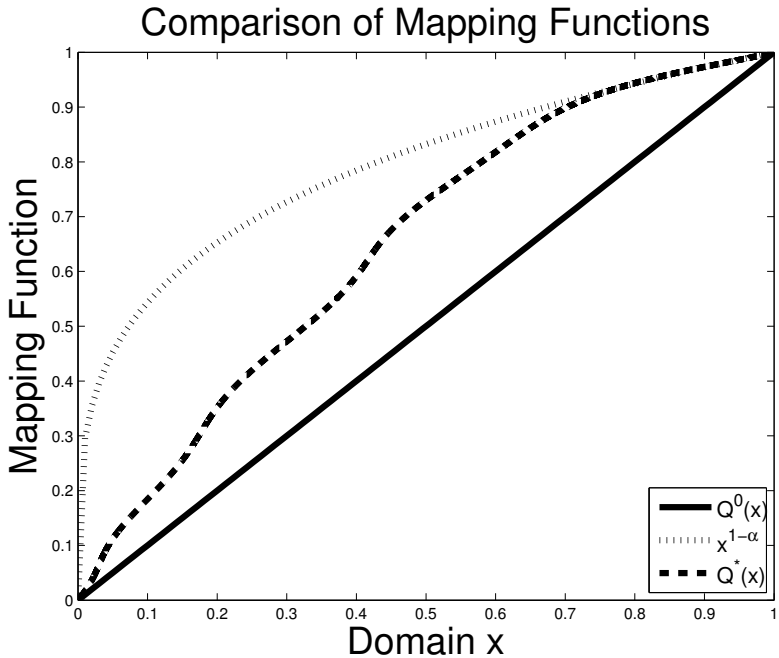
$$(x^{\alpha}u')' = \beta x^{\alpha+\beta-2}((\alpha+\beta-1)+\beta x^{\beta})u \qquad ; 0 \le x \le 1$$

$$u(0) = A\,,\; u(1) = B$$

$$Q^*(x) = x^{1-\alpha}$$

Sparse-Mesh Solutions (Aziz & Kumar, JCAM 2001)



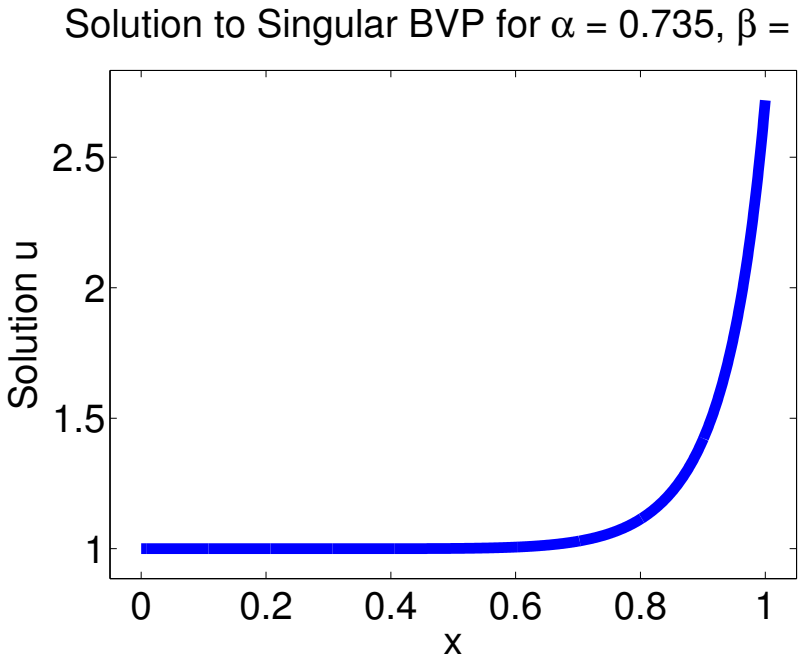


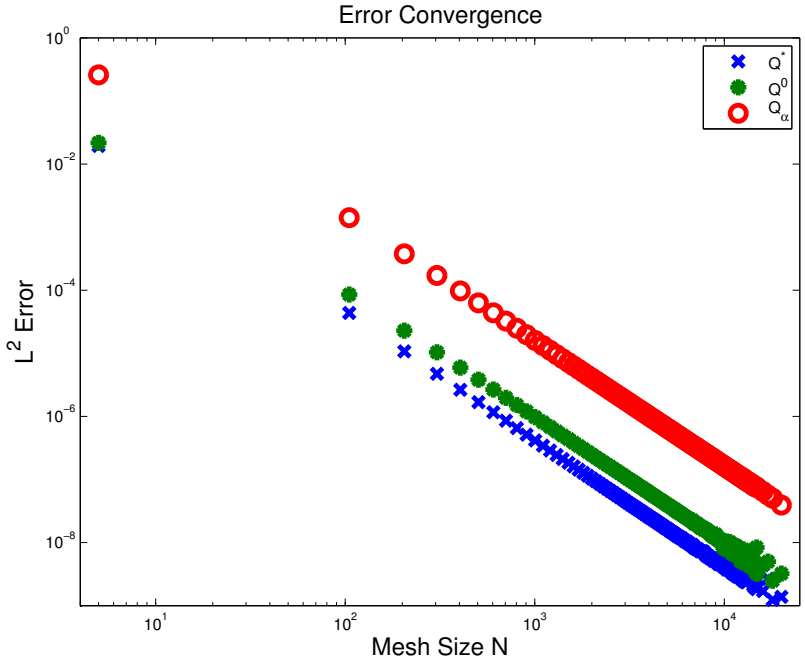
$$(x^{\alpha}u')' = \beta x^{\alpha+\beta-2}((\alpha+\beta-1)+\beta x^{\beta})u \qquad ; 0 \le x \le 1$$

$$u(0) = A\,,\; u(1) = B$$

$$Q^*(x) = x^{1-\alpha}$$

Sparse-Mesh Solutions (Aziz & Kumar, JCAM 2001)





$$(x^{\alpha}u')' = \beta x^{\alpha+\beta-2}((\alpha+\beta-1)+\beta x^{\beta})u \qquad ; 0 \le x \le 1$$

$$u(0) = A\,,\; u(1) = B$$

$$Q^*(x) = x^{1-\alpha}$$

Algorithm Development

$$X(P) = \text{random variable with distribution } P$$

Algorithm Development

$$X(P) = \text{random variable with distribution } P$$

$$\mathbb{X}_n(P) = (X_1(P), X_2(P), \ldots, X_n(P))^T$$

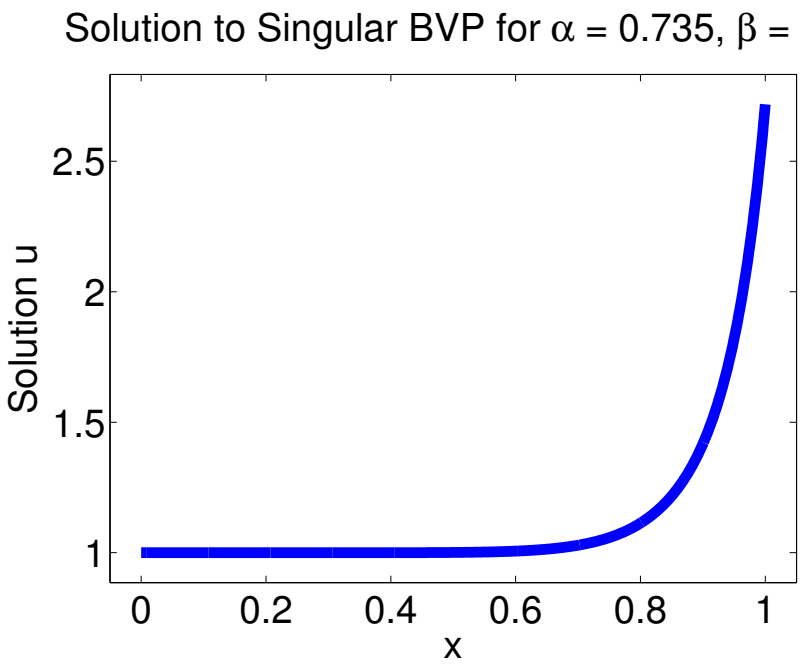
Algorithm Development

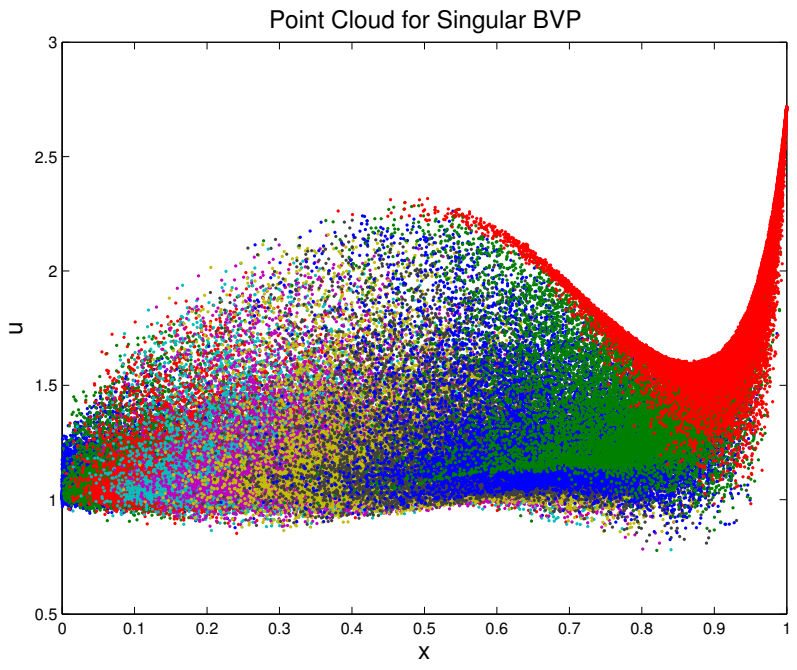
$$X(P) = \text{random variable with distribution } P$$

$$\mathbb{X}_n(P) = (X_1(P), X_2(P), \ldots, X_n(P))^T$$

$$\mathbb{X}_{(n)}(P) = \text{sorted vector}$$

Sampling and Solving





$$(x^\alpha u')' = \beta x^{\alpha+\beta-2}((\alpha+\beta-1)+\beta x^\beta)u \qquad ; 0 \le x \le 1$$

$$u(0) = A\,,\ u(1) = B$$

$$Q^*(x) = x^{1-\alpha}$$

Algorithm Development

$$X(P) = \text{random variable with distribution } P$$

$$\mathbb{X}_n(P) = (X_1(P), X_2(P), \ldots, X_n(P))^T$$

$$\mathbb{X}_{(n)}(P) = \text{sorted vector}$$

$$\begin{aligned} \Pr\left[K = k : X_{(k)} \leq \xi\right] &= \Pr\left[\left(\bigcap_{j=1}^{k-1}\{X_{(j)} < \xi\}\right) \cap \left(\bigcap_{j=k+1}^{n}\{X_{(j)} > \xi\}\right)\right] \\ \text{CDF} &= \binom{n-1}{k-1} P(\xi)^{i-1}(1 - P(\xi))^{n-(i-1)}, \end{aligned}$$

Algorithm Development

$$X(P) = \text{random variable with distribution } P$$

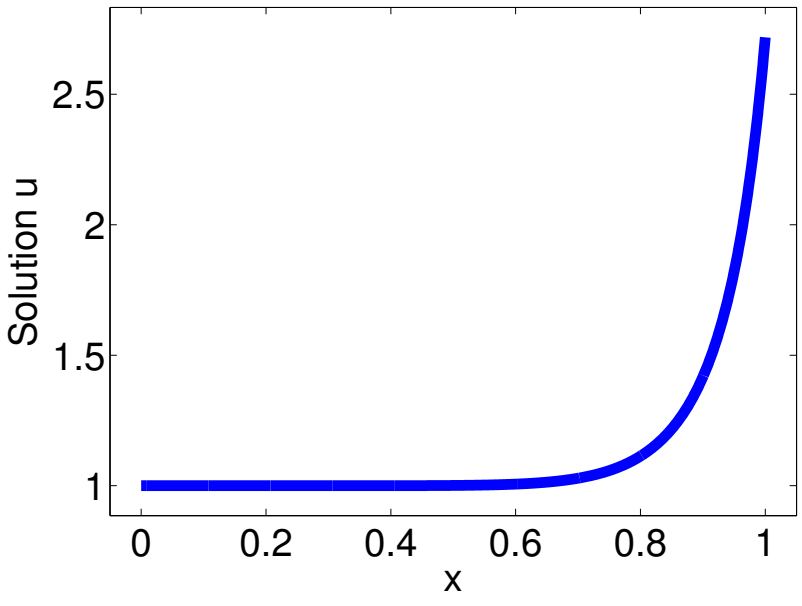
$$\mathbb{X}_n(P) = (X_1(P), X_2(P), \ldots, X_n(P))^T$$

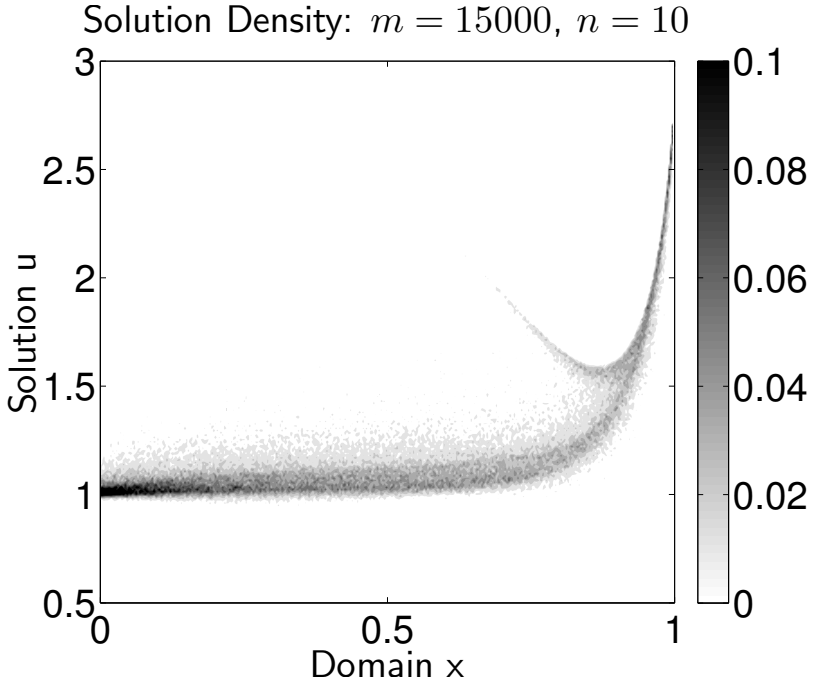
$$\mathbb{X}_{(n)}(P) = \text{sorted vector}$$

$$\begin{aligned} \Pr\left[K = k : X_{(k)} \leq \xi\right] &= \Pr\left[\left(\bigcap_{j=1}^{k-1}\{X_{(j)} < \xi\}\right) \cap \left(\bigcap_{j=k+1}^{n}\{X_{(j)} > \xi\}\right)\right] \\ \text{CDF} &= \binom{n-1}{k-1} P(\xi)^{i-1}(1 - P(\xi))^{n-(i-1)}, \end{aligned}$$

$$U(\mathbf{x}) = \text{takes a vector } \mathbf{x} \text{ and generates numerical solution}$$

Theoretical Framework?





$$(x^{\alpha}u')' = \beta x^{\alpha+\beta-2}((\alpha+\beta-1)+\beta x^{\beta})u \qquad ; 0 \le x \le 1$$

$$u(0) = A\,,\; u(1) = B$$

$$Q^*(x) = x^{1-\alpha}$$

Algorithm Development

Definition

Pointwise probability density function

$$p(\xi, \mathbb{X}_{(n)}(P)) \equiv \mathbb{E}_K \left[\left\{ U(\mathbb{X}_{(n)}(P)) \right\}_{K=k} \middle| X_{(k)} = \xi \right]$$

Algorithm Development

Definition

Pointwise probability density function

$$p(\xi, \mathbb{X}_{(n)}(P)) \equiv \mathbb{E}_K \left[\left\{ U(\mathbb{X}_{(n)}(P)) \right\}_{K=k} \middle| X_{(k)} = \xi \right]$$

Definition

Pointwise mean

$$\mu(\xi) \equiv \mathbb{E}_P \left[\mathbb{E}_K \left[\left\{ U(\mathbb{X}_{(n)}(P)) \right\}_{K=k} \middle| X_{(k)} = \xi \right] \right] .$$

Algorithm Development

Definition

Pointwise probability density function

$$p(\xi, \mathbb{X}_{(n)}(P)) \equiv \mathbb{E}_K \left[\left\{ U(\mathbb{X}_{(n)}(P)) \right\}_{K=k} \middle| X_{(k)} = \xi \right]$$

Definition

Pointwise mean

$$\mu(\xi) \equiv \mathbb{E}_P \left[\mathbb{E}_K \left[\left\{ U(\mathbb{X}_{(n)}(P)) \right\}_{K=k} \middle| X_{(k)} = \xi \right] \right] .$$

Definition

Pointwise variance

$$v(\xi) \equiv \mathbb{V}_P \left[\mathbb{E}_K \left[\left\{ U(\mathbb{X}_{(n)}(P)) \right\}_{K=k} \middle| X_{(k)} = \xi \right] \right] ,$$

Theory

Theorem

Expected error converges pointwise to zero.

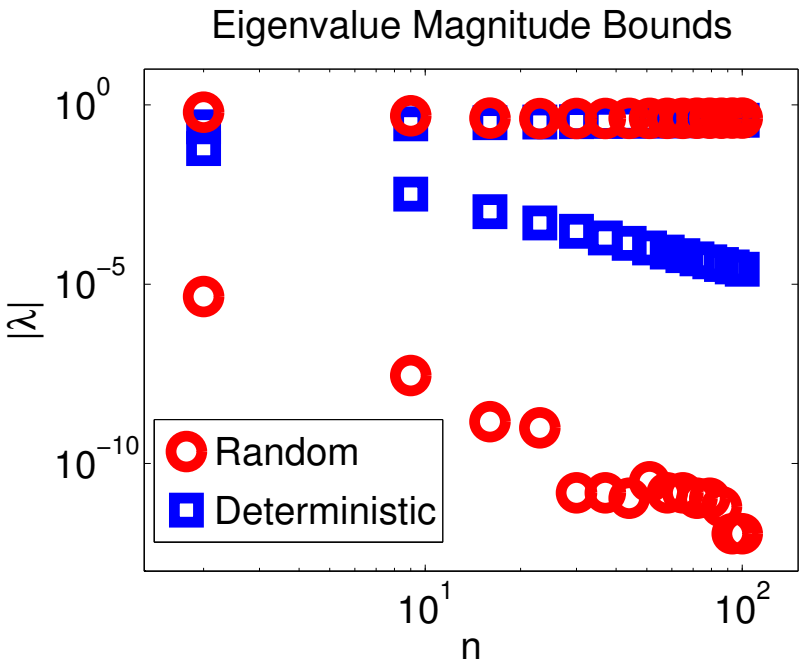
Proof.

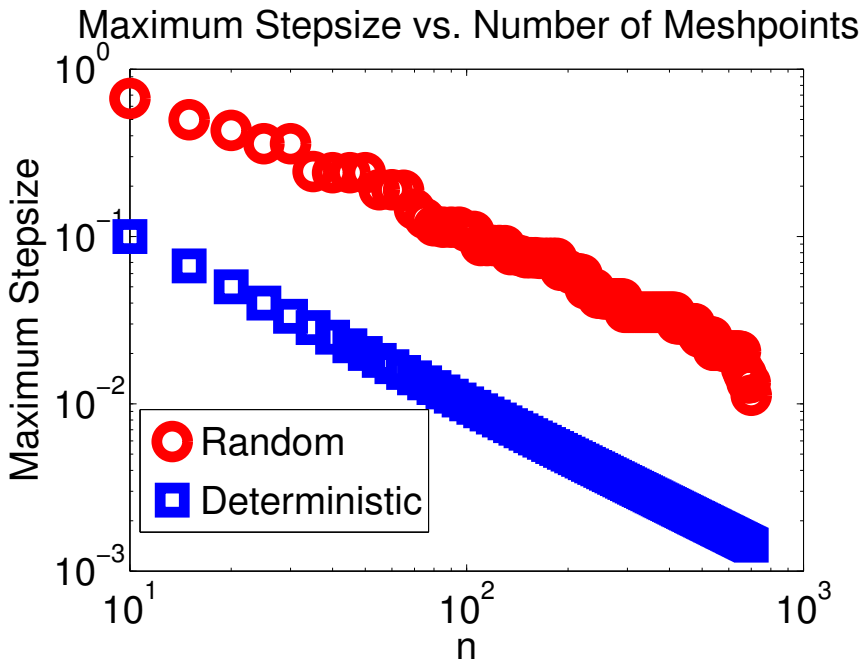
$$\left\| u(\xi) - \mu(\xi; U(\mathbb{X}_{(n)}(P))) \right\| \leq \mathbb{E}_P \left[\left\| A^{-1}_{\mathbb{X}_{(n)}(P)} \right\| \cdot \mathcal{O}(\max_{i=1,\ldots,n-1} |X_{(i)} - X_{(i+1)}|) \right]$$

□

DMB & AJC, *SIAM J. Sci. Comp.*, in revision.

"Proof"


Eigenvalue Magnitude Bounds
10^0
10^{-5}
10^{-10}
$|\lambda|$
Random
Deterministic
10^1
10^2
n


Maximum Stepsize vs. Number of Meshpoints
10^0
10^{-1}
10^{-2}
10^{-3}
Maximum Stepsize
Random
Deterministic
10^1
10^2
10^3
n

$$m = 1000$$

Expected Least Squares $\mathbb{E}[J]$

$$\mathbb{E}\left[J(U(\mathbb{X}_{(n)}(P))\right] \approx \mathbb{E}\left[J(\mu_n^*) + J_1(\mu_n^*) + J_2(\mu_n^*)\right]$$

Expected Least Squares $\mathbb{E}[J]$

$$\mathbb{E}\left[J(U(\mathbb{X}_{(n)}(P))\right] \approx \mathbb{E}\left[J(\mu_n^*) + J_1(\mu_n^*) + J_2(\mu_n^*)\right]$$

where

$$\begin{aligned} J_1(U_n^*) &= (U(P) - \mu_n^*)\nabla_U J(U_n^*) \\ &= 2\left(U(P) - \mu_n^*\right)^T (\mu_n^* - u) = 0 \end{aligned}$$

Expected Least Squares $\mathbb{E}[J]$

$$\mathbb{E}\left[J(U(\mathbb{X}_{(n)}(P))\right] \approx \mathbb{E}\left[J(\mu_n^*) + J_1(\mu_n^*) + J_2(\mu_n^*)\right]$$

where

$$\begin{aligned} J_1(U_n^*) &= (U(P) - \mu_n^*)\nabla_U J(U_n^*) \\ &= 2\left(U(P) - \mu_n^*\right)^T (\mu_n^* - u) = 0 \end{aligned}$$

and

$$\begin{aligned} J_2(U_n^*) &= (U(P) - \mu_n^*)^T \nabla_U \left[(U(P) - \mu_n^*)^T \nabla_U \left[J(U_n^*)\right]\right] \\ &= (U(P) - \mu_n^*)^T \nabla_U^2 J(U_n^*)\left(U(P) - \mu_n^*\right) \\ &= (U(P) - \mu_n^*)^T (U(P) - \mu_n^*) = \text{Variance}\,. \end{aligned}$$

Expected Least Squares $\mathbb{E}[J]$

$$\mathbb{E}\left[J(U(\mathbb{X}_{(n)}(P))\right] \approx \mathbb{E}\left[J(\mu_n^*) + J_1(\mu_n^*) + J_2(\mu_n^*)\right]$$

where

$$\begin{aligned} J_1(U_n^*) &= (U(P) - \mu_n^*)\nabla_U J(U_n^*) \\ &= 2\left(U(P) - \mu_n^*\right)^T (\mu_n^* - u) = 0 \end{aligned}$$

and

$$\begin{aligned} J_2(U_n^*) &= (U(P) - \mu_n^*)^T \nabla_U \left[(U(P) - \mu_n^*)^T \nabla_U \left[J(U_n^*)\right]\right] \\ &= (U(P) - \mu_n^*)^T \nabla_U^2 J(U_n^*)\left(U(P) - \mu_n^*\right) \\ &= (U(P) - \mu_n^*)^T (U(P) - \mu_n^*) = \text{Variance}\,. \end{aligned}$$

- Optimize variance of U as a surrogate for error?
- SQP with Hessian Lagrangian updated via BFGS?

Theory

Theorem

Hausdorf distance convergence as $m, n \to \infty$ for

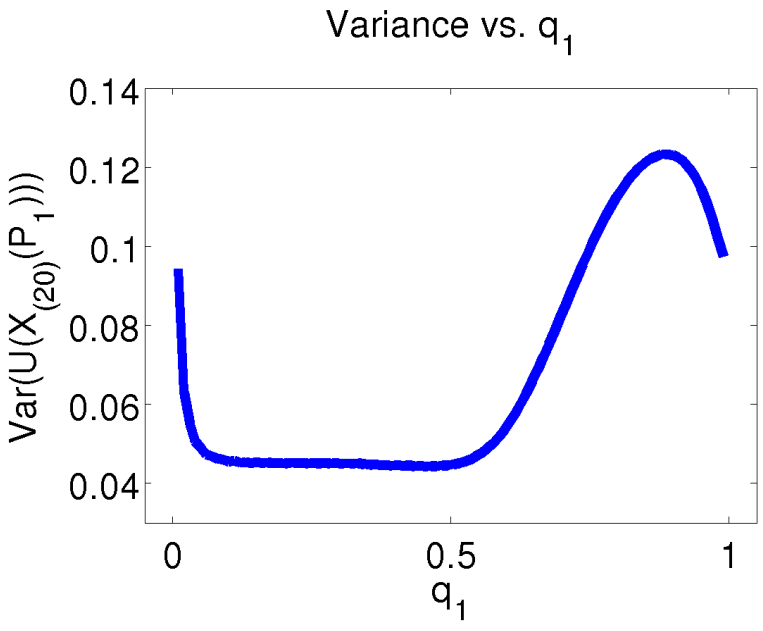
- *set of optimal distributions for approximate solution*
- *set of optimal distributions for exact solution*

Corollary

Well-posedness of inverse problem of P^* identification.

DMB & Byrne, *Inverse Problems*, in revision.

In Practice...



- Sufficient sampling addresses smoothness.
- Problems with optimization:
  - initial P guess.
  - discretization resolution N
  - number of samples M.
- Inference from moments of coarse solutions?

Distribution Construction

Eilliptic

$$q_1(x) = \int_0^x \sqrt{\left|\mu'(\xi; U(\mathbb{X}_{(n)}(P))\right|} d\xi \,,$$

Hamilton-Jacobi

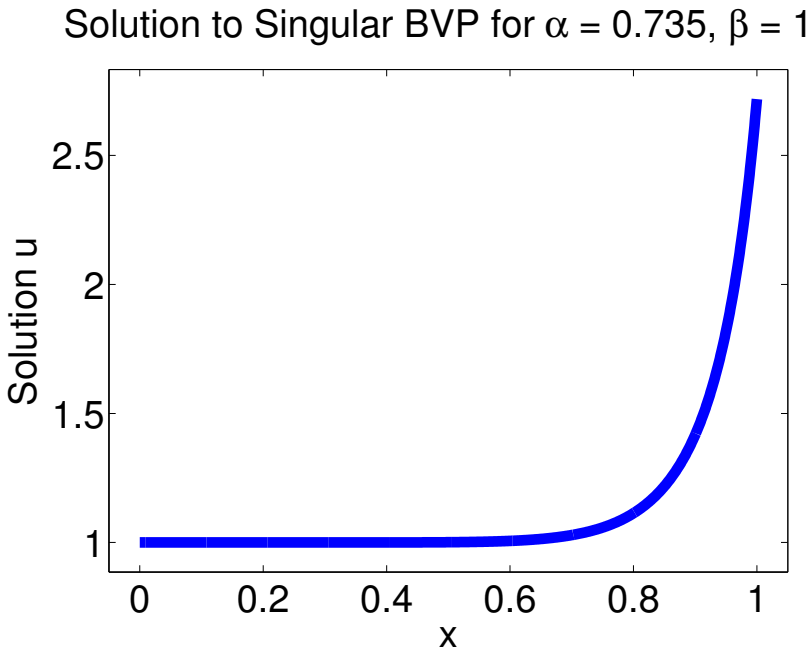
$$q_2(x) = \int_0^x \mu''(\xi; U(\mathbb{X}_{(n)}(P))^2 \, v(\xi; U(\mathbb{X}_{(n)}(P))^2 d\xi \,,$$

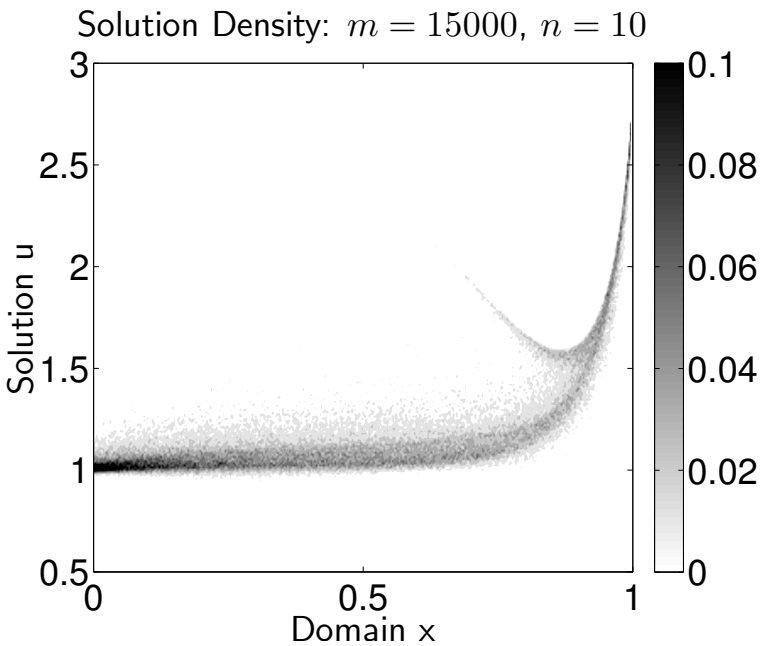
$$Q(x) = \left[\frac{q_i(\cdot)}{q_i(1)}\right]^{-1}(x)$$

Results[1]

[1] Simulations from runs on a Tesla C2070 using matlab's GPU toolbox (**arrayfun**).

Singular BVP



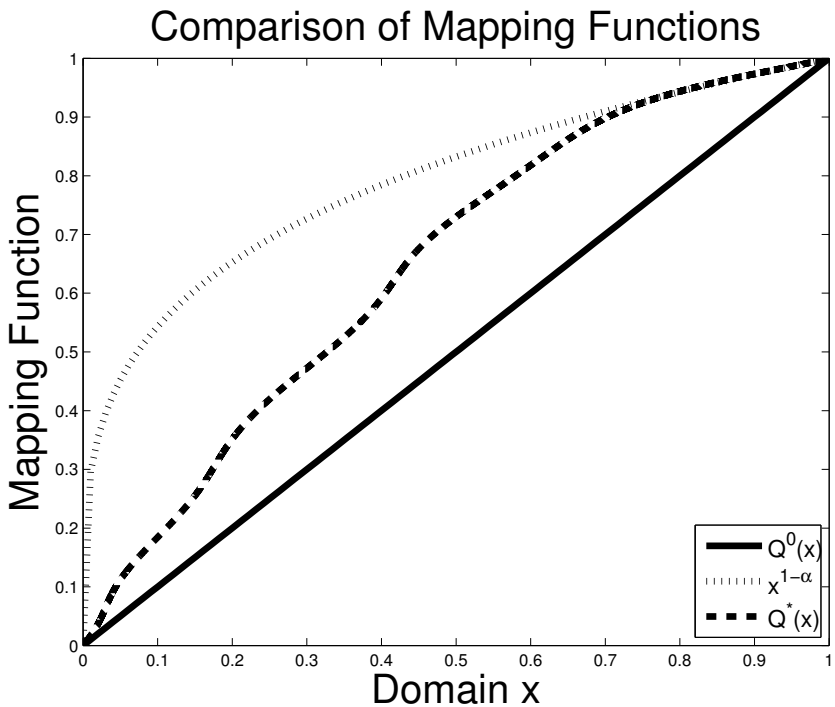


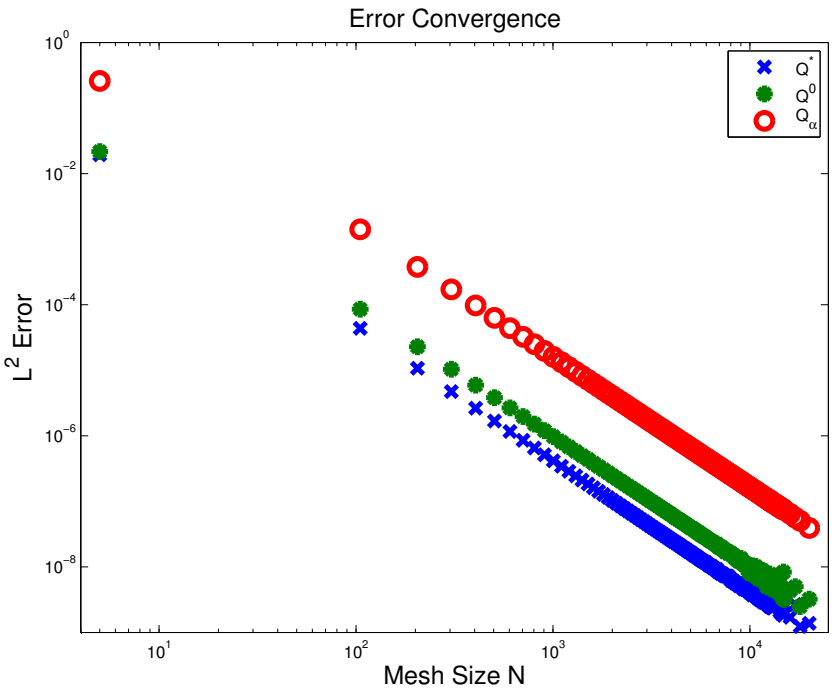
$$(x^{\alpha}u')' = \beta x^{\alpha+\beta-2}((\alpha+\beta-1)+\beta x^{\beta})u \qquad ; 0 \leq x \leq 1$$

$$u(0) = A\,,\; u(1) = B$$

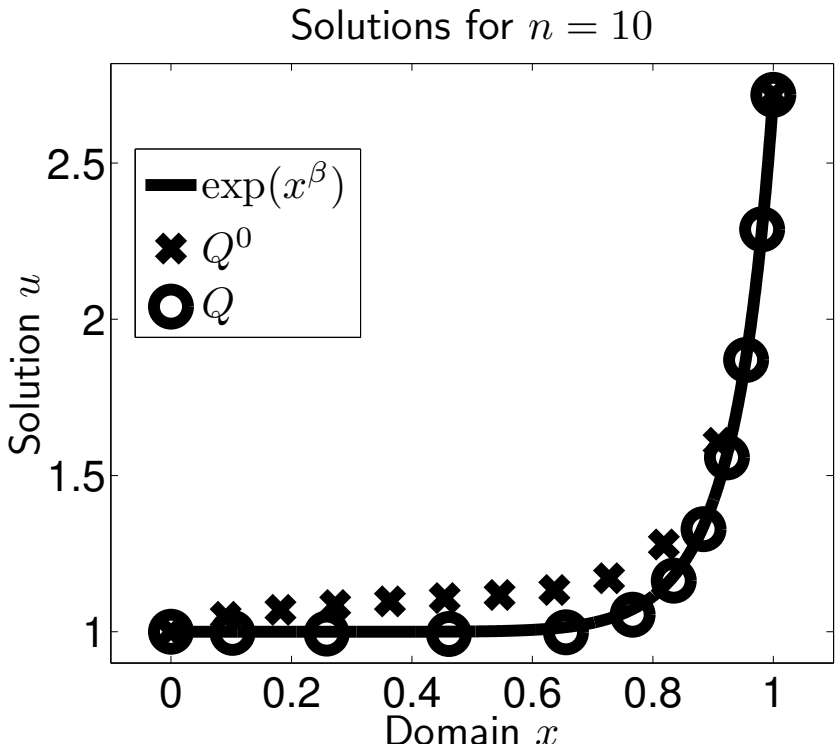
$$u(x) = \exp(x^{\beta})$$

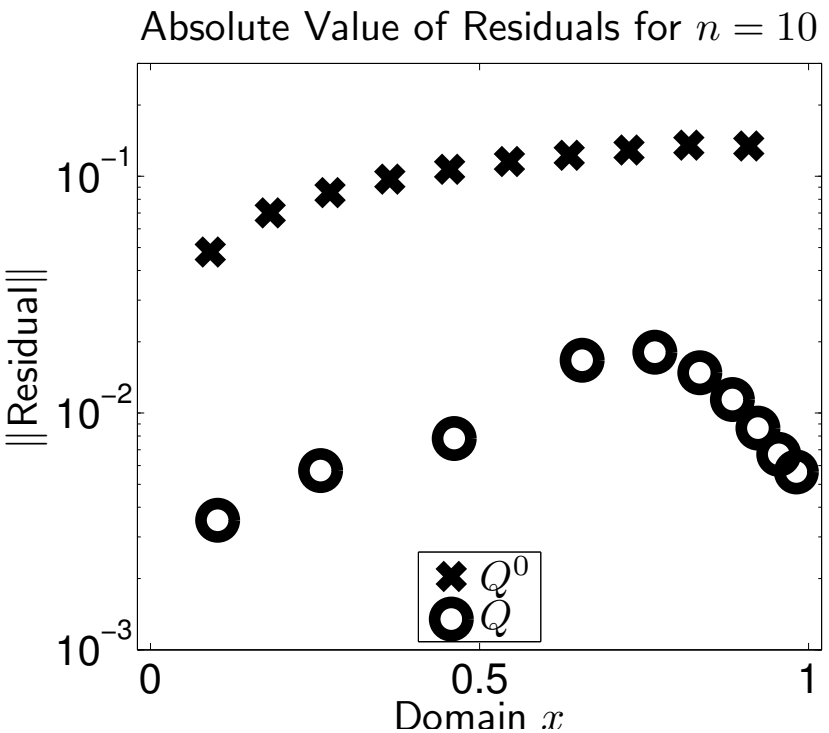
Singular BVP



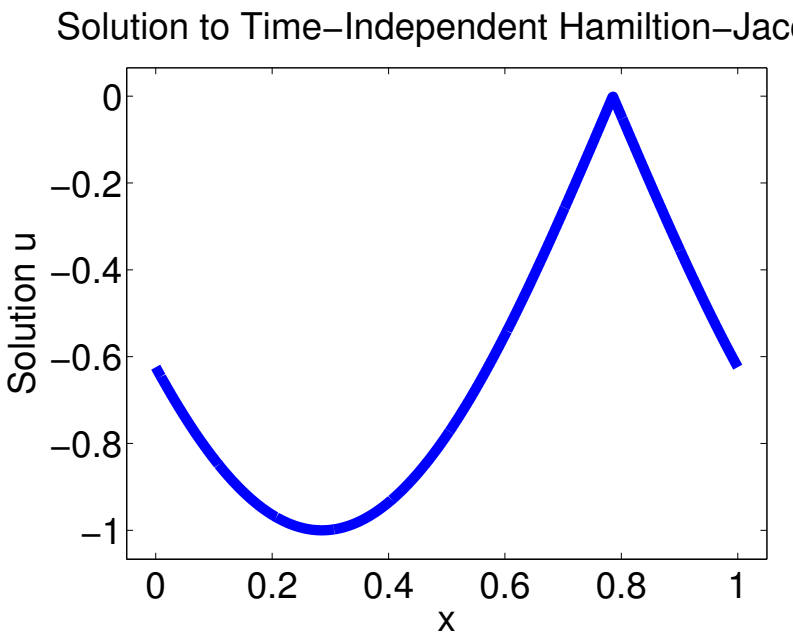


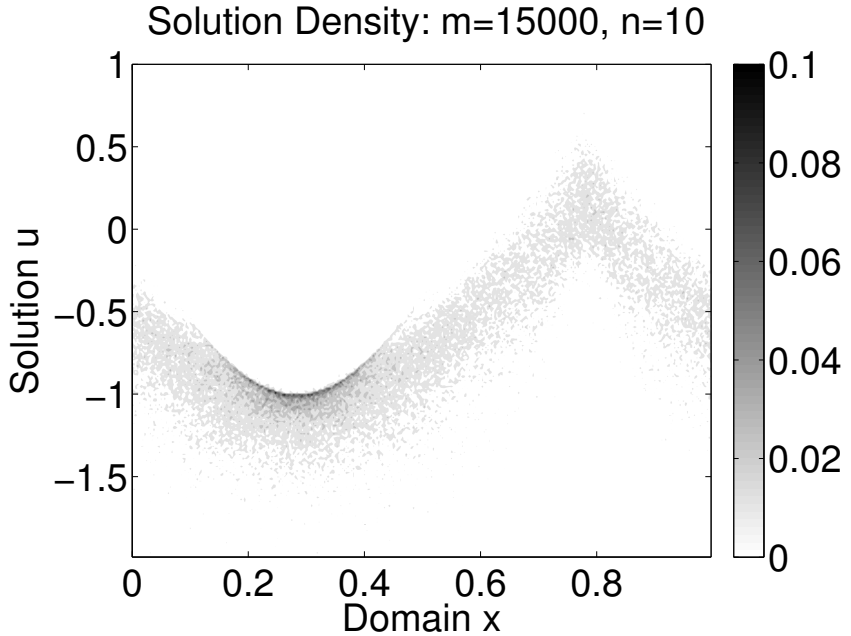
Singular BVP





Time-Invariant Hamilton-Jacobi Equation



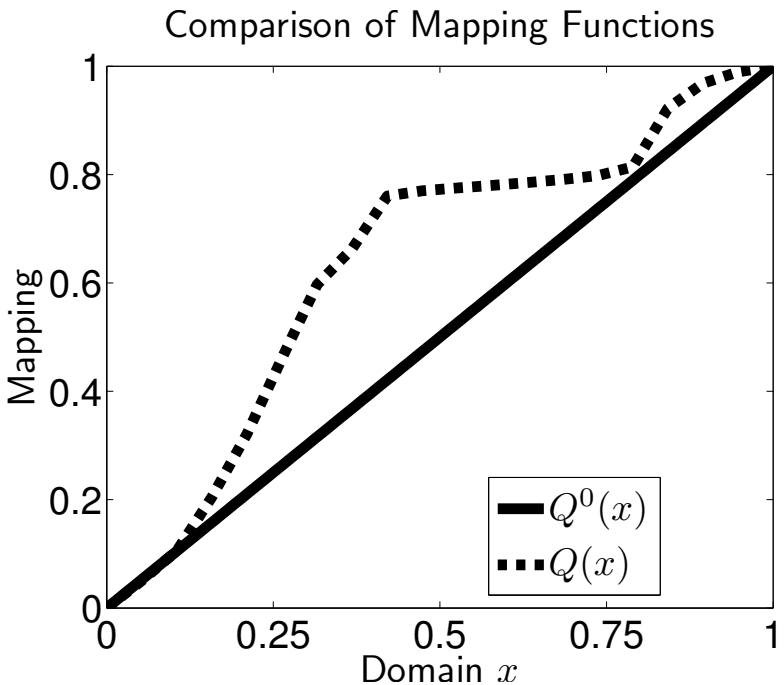


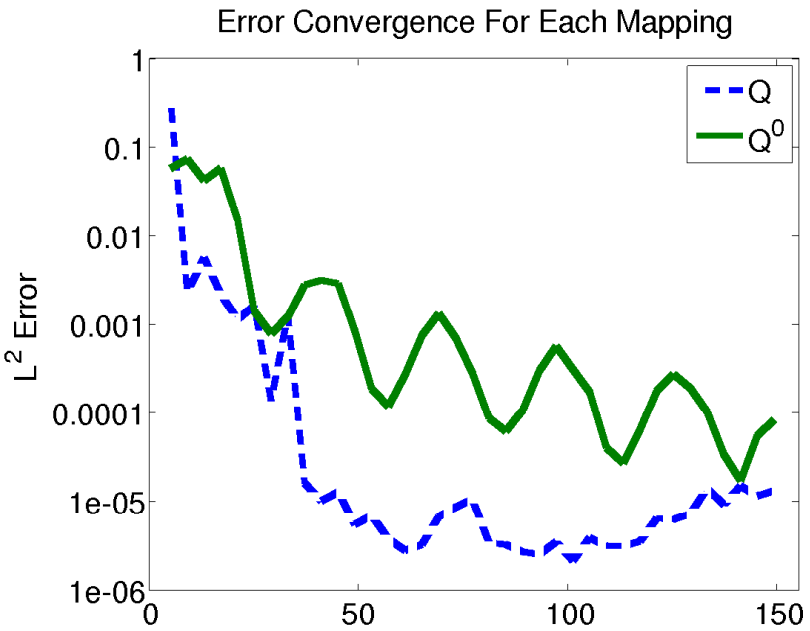
$$u + (u_x)^2/\pi^2 = -\left|\sin(\pi(x - \frac{\pi}{4}))\right| + \cos^2(\pi(x - \frac{\pi}{4})), \qquad (1)$$

$$\text{Solution: } u(x) = -\left|\sin\left(\pi\left(x - \frac{\pi}{4}\right)\right)\right|$$

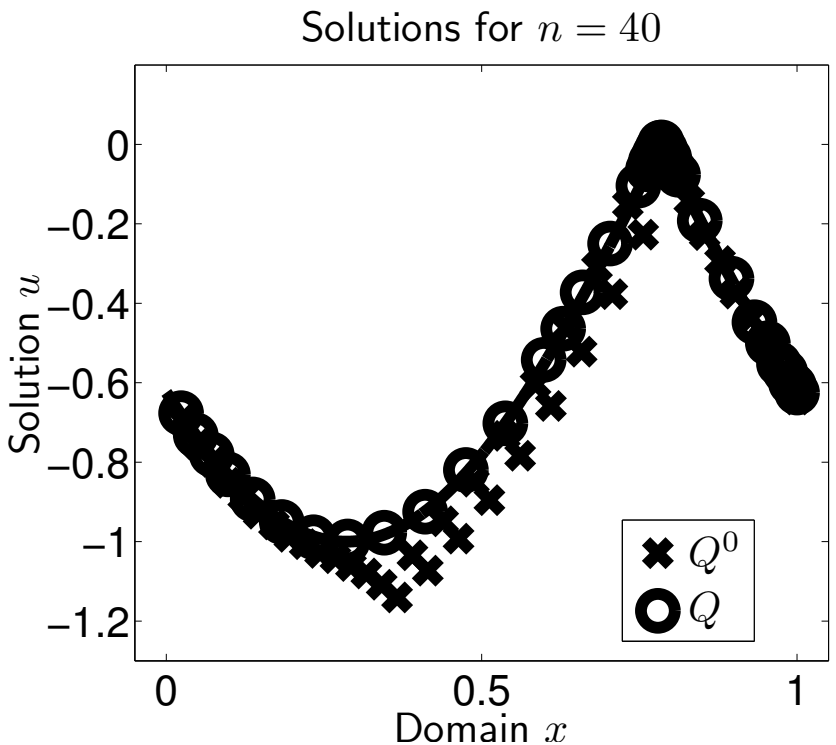
Cockburn & Yenikaya 2005

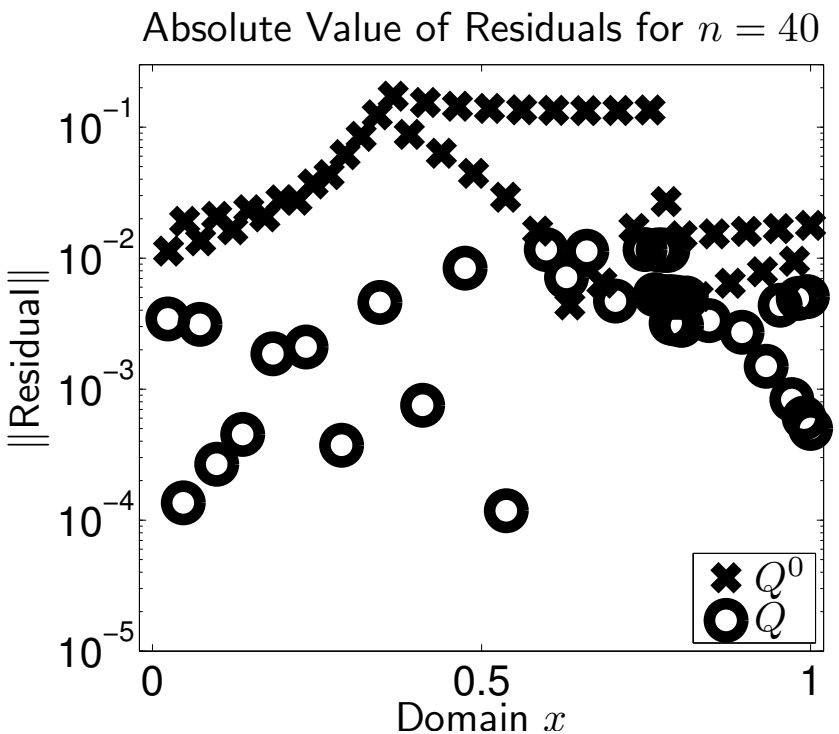
Hamilton-Jacobi



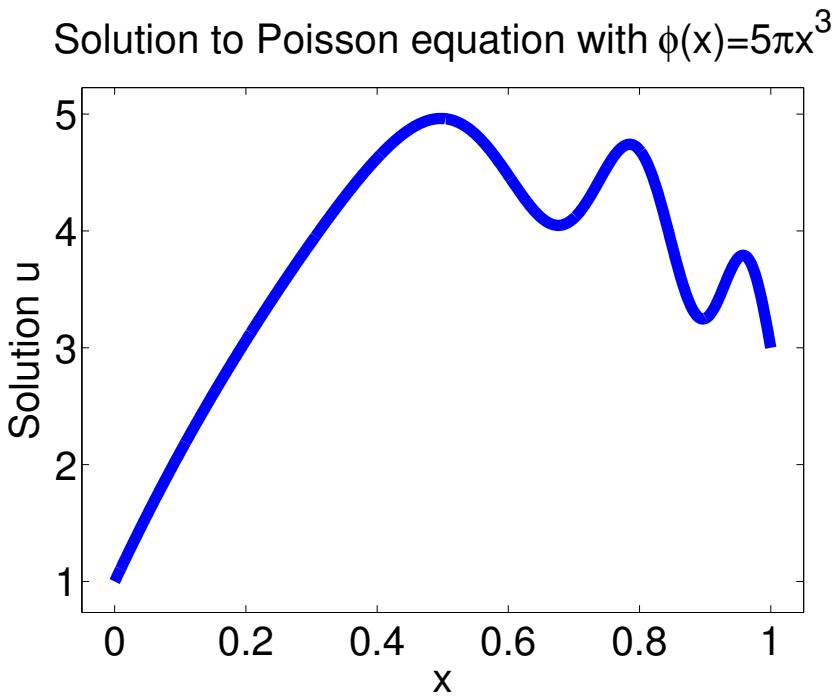


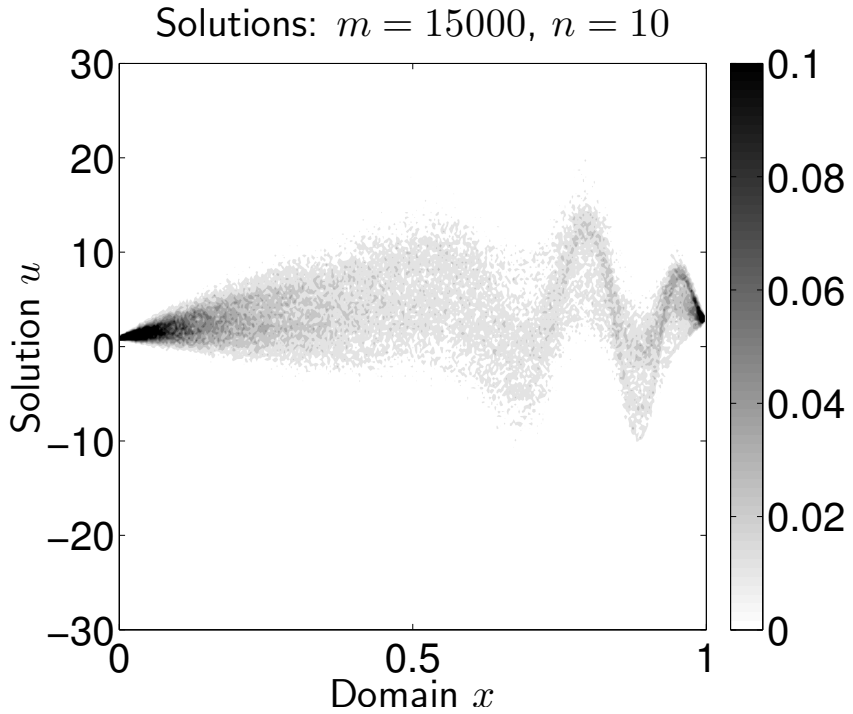
Hamilton-Jacobi





Oscillatory BVP



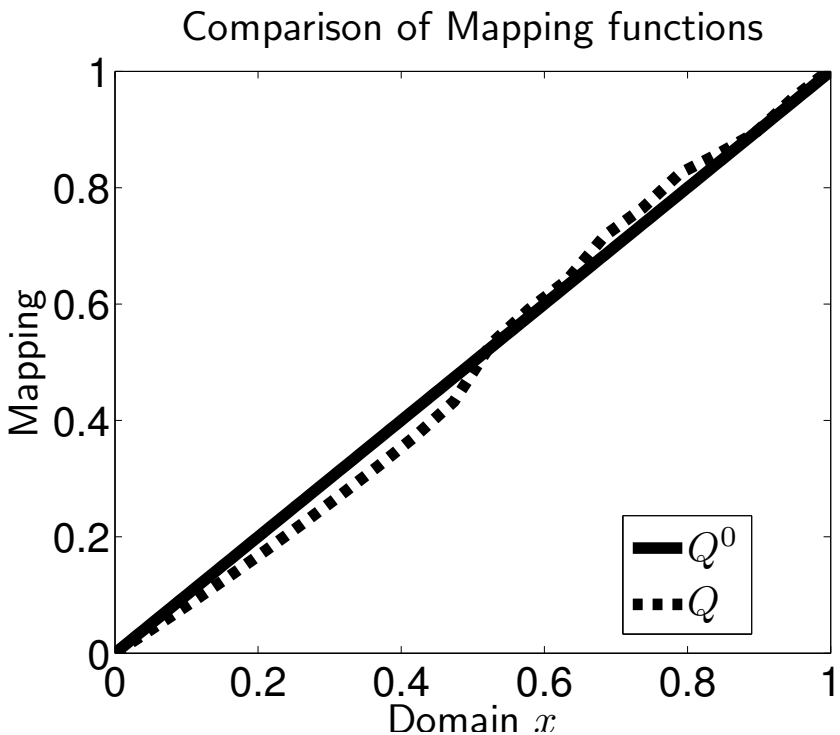


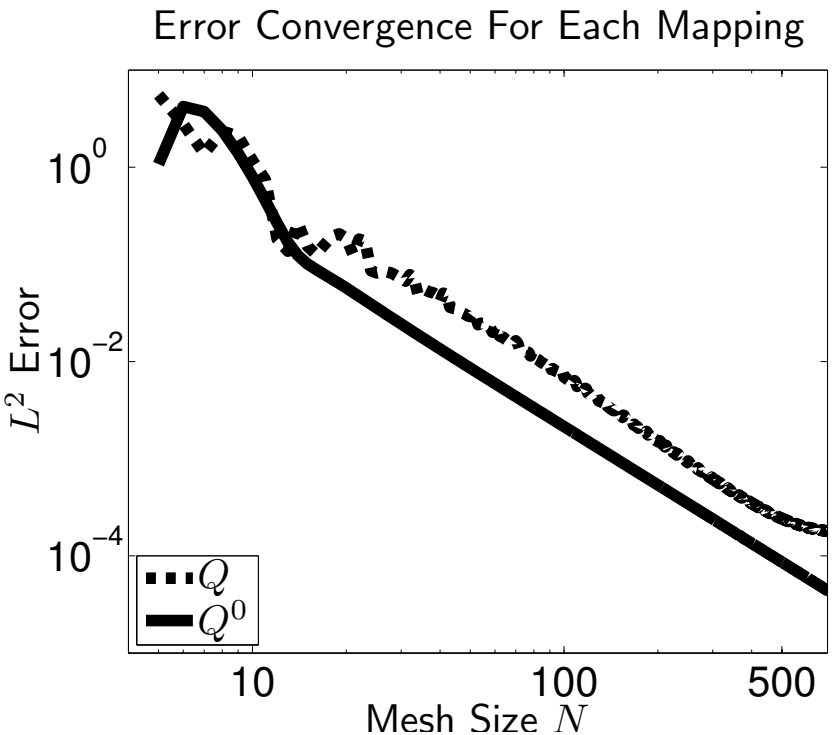
$$u''(x) = -20 + 0.5\phi''(x)\cos(\phi(x)) - 0.5(\phi'(x))^2 \sin(\phi(x))$$

for $\phi(x) = 5\pi x^3$, $u(0) = 1$, and $u(1) = 3$ with solution

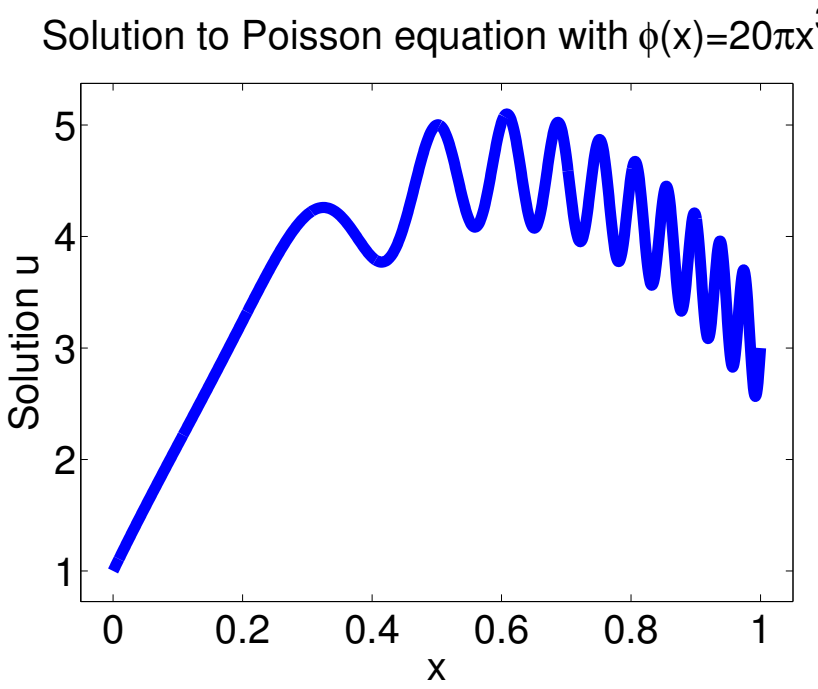
$$u(x) = 1 + 12x - 10x^2 + a\sin(\phi(x))$$

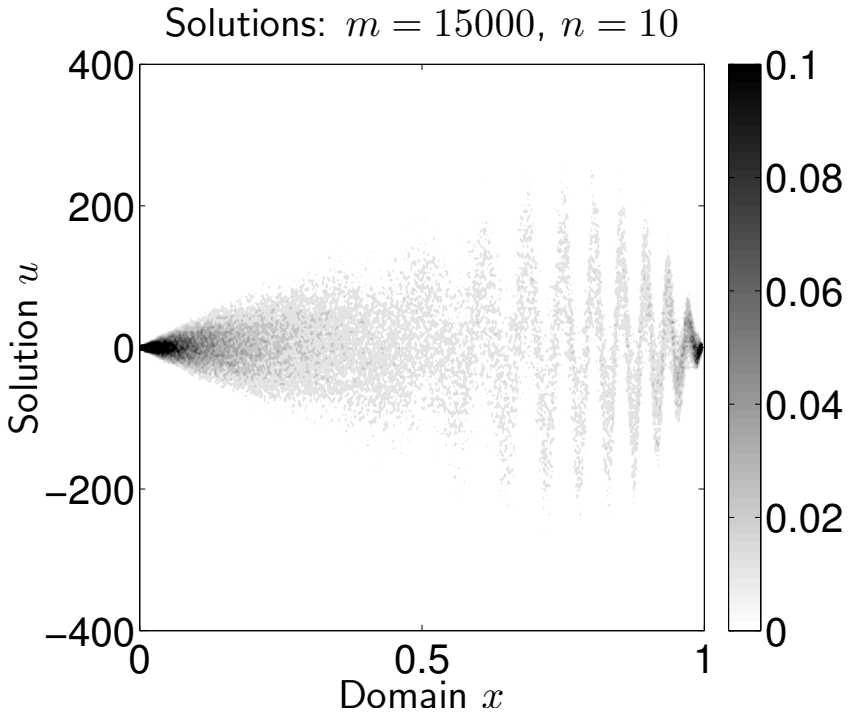
Oscillatory BVP





Highly Oscillatory BVP



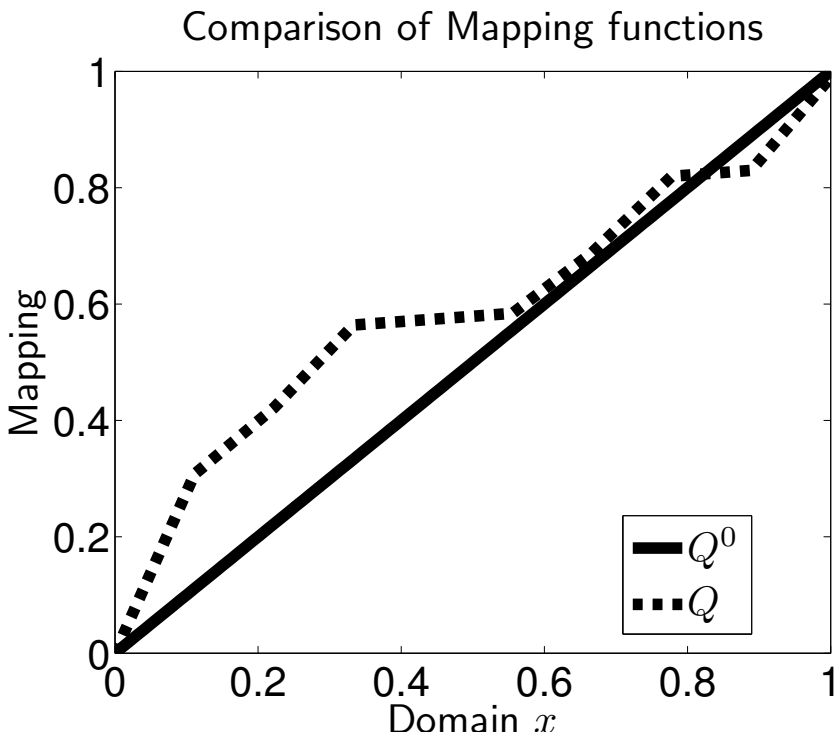


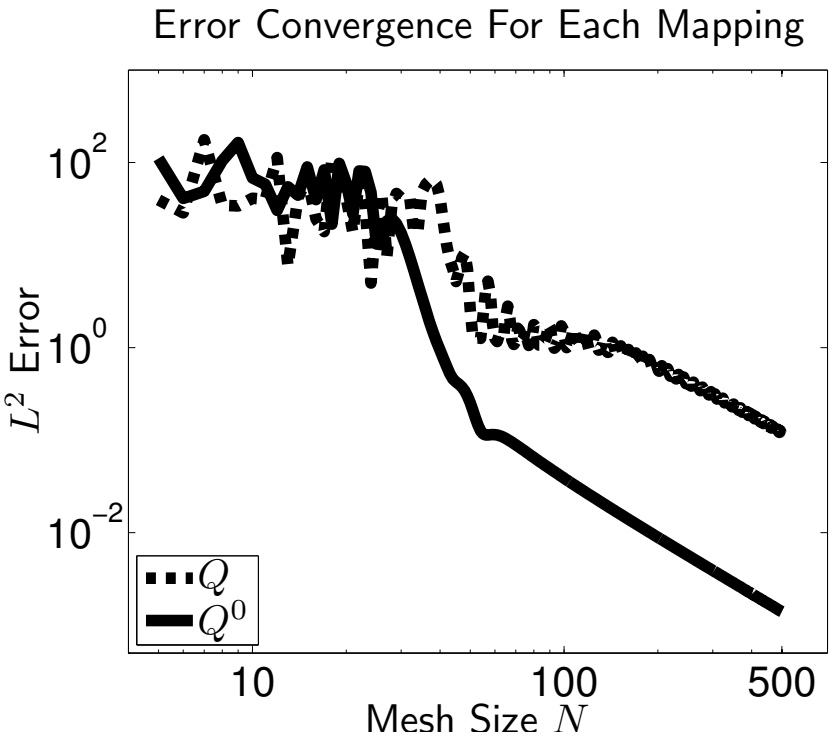
$$u''(x) = -20 + 0.5\phi''(x)\cos(\phi(x)) - 0.5(\phi'(x))^2 \sin(\phi(x))$$

for $\phi(x) = 20\pi x^3$, $u(0) = 1$, and $u(1) = 3$ with solution

$$u(x) = 1 + 12x - 10x^2 + a\sin(\phi(x))$$

Highly Oscillatory BVP





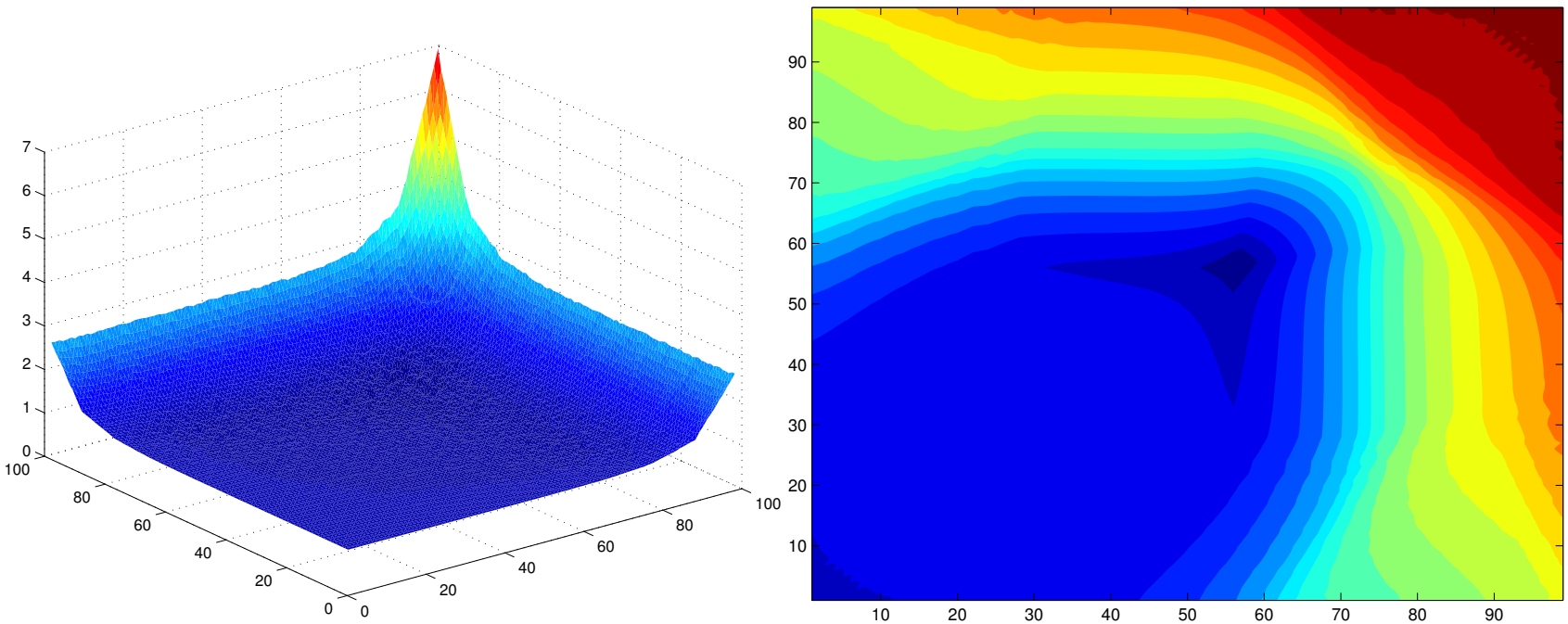
Summary

- Methodology exploiting massively-multicore architecture:
  - Minimizing variance of solutions is not robust
  - Use moments of sparse-mesh solutions to create superior nonuniform mesh
  - Promising for problems with local features
  - Disappointing for problems with oscillations
- Theory
  - First moment convergence (in expectation)
  - Consistency and Stability (in expectation)
  - Inverse Problem well-posedness
- 1D mesh generation: DMB & AJC, *SIAM J. Sci. Comp.*, in revision.
- Theory: DMB & Byrne, *Inverse Problems*, in revision.

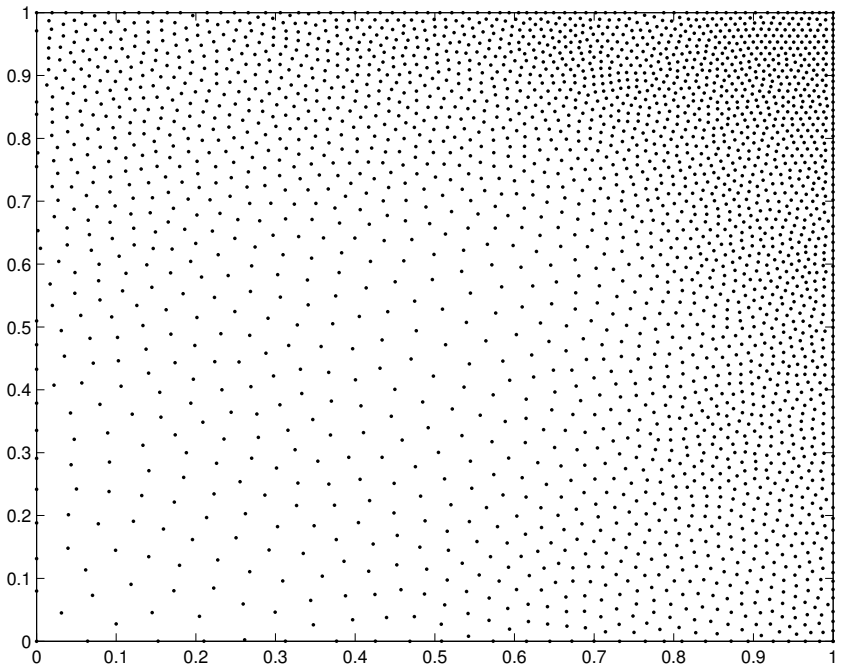
Work in Progress

- Framework for creating Q from moments.
- 2D investigation
  - Standard FEM solve
  - Comparison with moving meshes
  - DMB & AJC, in prep.
- Moving Mesh Methods
  - Monitor functions

Singular BVP

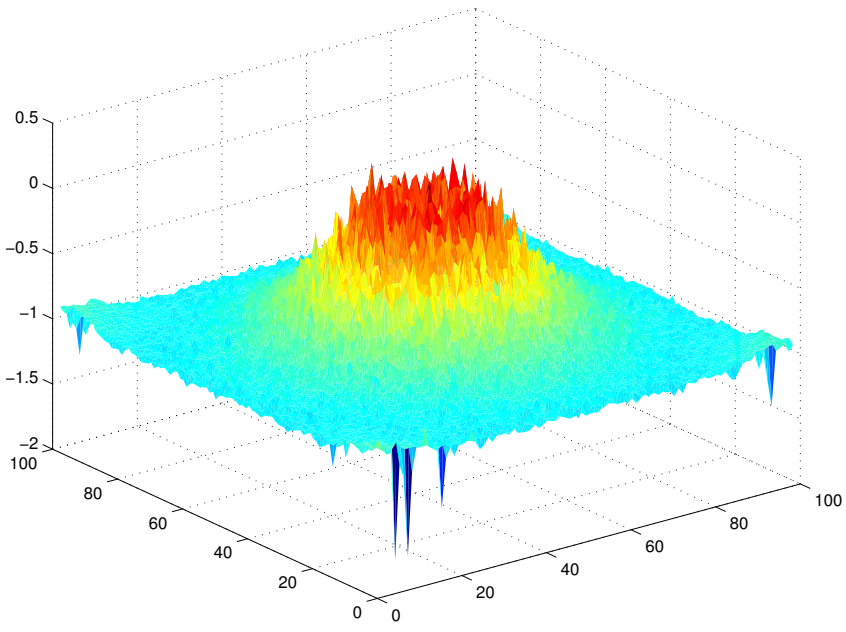


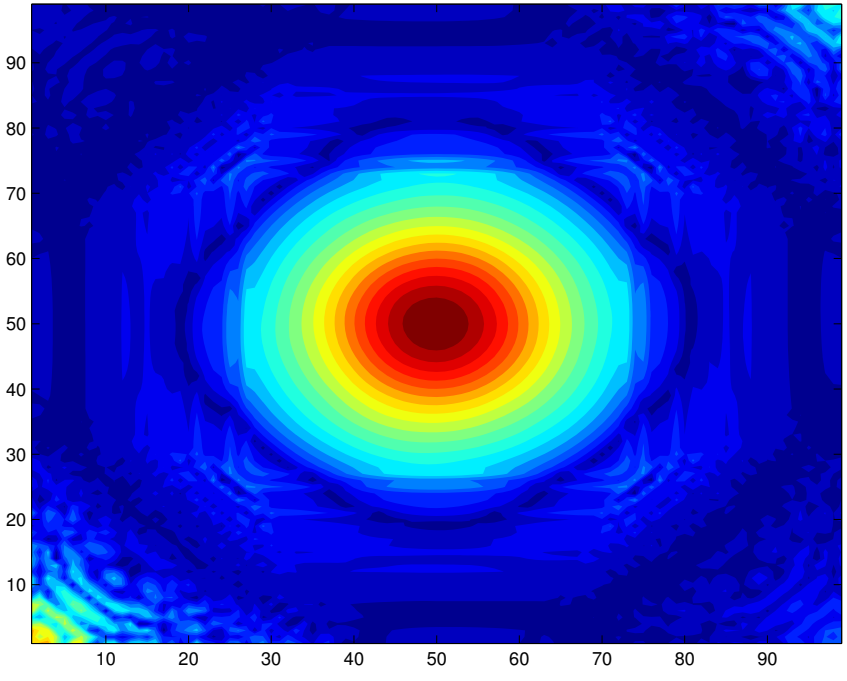
Singular BVP



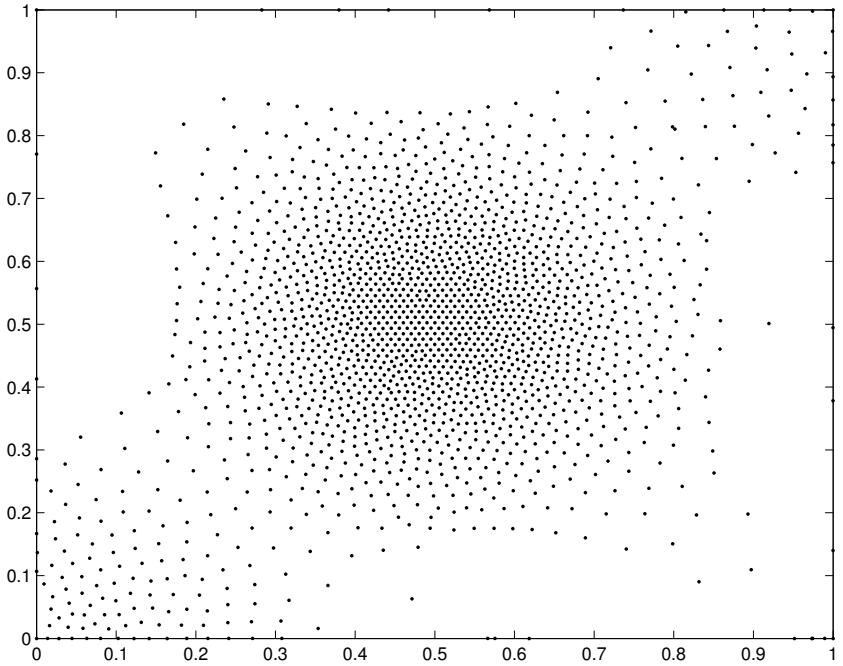
$$h_{min} = 0.01$$

Hamilton-Jacobi





Hamilton-Jacobi



Acknowledgments

Comp. Math. FA9550-09-1-0403

Academic Partnership Program