Variational quantum algorithms for numerical PDE solving

Pete Rigas, Dartmouth Applied Math seminar

McMahon Lab

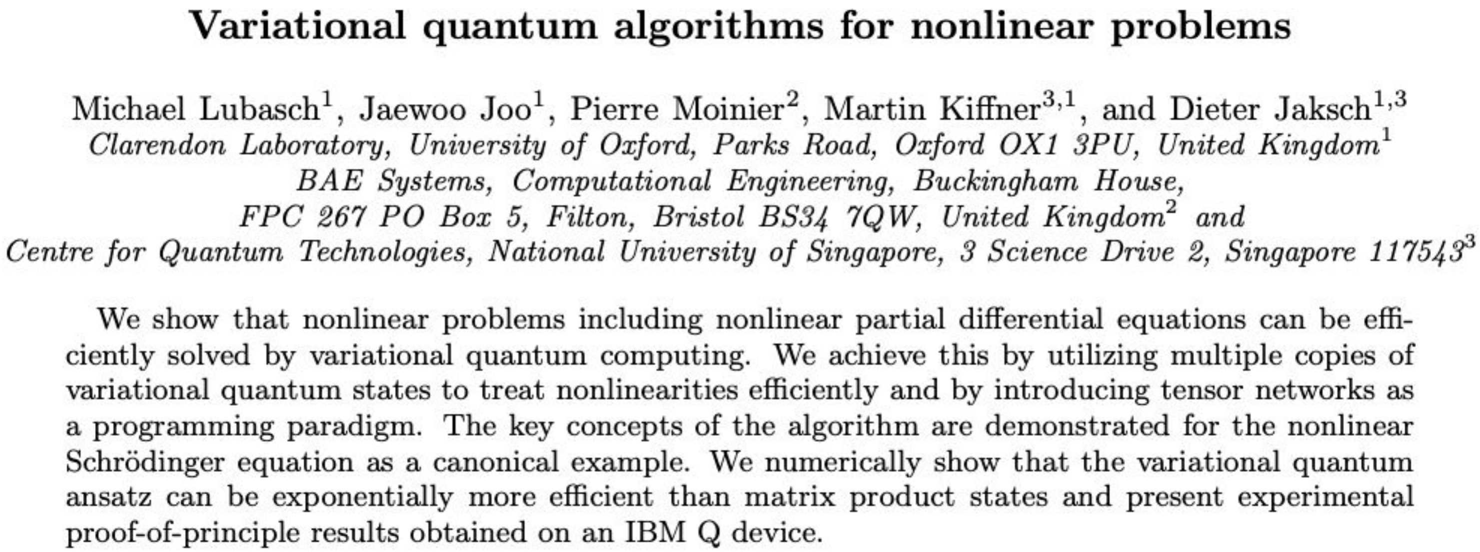
Presentation overview

- Introduce connections of our work with numerical methods for solving PDEs raised from a new quantum algorithm
- Noiseless quantum simulation results generated with Cirq
- Devote more attention towards preliminary simulation results for some PDEs since April

Straddle between two fields

Pure Mathematics ⟷ Applied Mathematics

Lubasch et al


Variational quantum algorithms for nonlinear problems

Michael Lubasch[1], Jaewoo Joo[1], Pierre Moinier[2], Martin Kiffner[3,1], and Dieter Jaksch[1,3]
Clarendon Laboratory, University of Oxford, Parks Road, Oxford OX1 3PU, United Kingdom[1]
BAE Systems, Computational Engineering, Buckingham House,
FPC 267 PO Box 5, Filton, Bristol BS34 7QW, United Kingdom[2] *and*
Centre for Quantum Technologies, National University of Singapore, 3 Science Drive 2, Singapore 117543[3]

We show that nonlinear problems including nonlinear partial differential equations can be efficiently solved by variational quantum computing. We achieve this by utilizing multiple copies of variational quantum states to treat nonlinearities efficiently and by introducing tensor networks as a programming paradigm. The key concepts of the algorithm are demonstrated for the nonlinear Schrödinger equation as a canonical example. We numerically show that the variational quantum ansatz can be exponentially more efficient than matrix product states and present experimental proof-of-principle results obtained on an IBM Q device.

<u>Main points of investigation</u>

- Can quantum algorithms solve well-posed IVPs?
- Can the solution approximations generated from such algorithms “compare” to classically computed solutions that we already know?

Lubasch et al

Let's break down components of the algorithm for one PDE, the Schrodinger equation

1

Formulate cost function by expanding the norm

In the variational setting, instead of directly computing $|f(t+\tau)\rangle$ via this formula, it is more efficient to define the cost function

$$\mathcal{C}(|f(t+\tau)\rangle) = \| |f(t+\tau)\rangle - (\mathbb{1} + \tau O(t))|f(t)\rangle \|^2 \qquad \text{(S3)}$$

and minimize this cost function via the variational parameters of $|f(t+\tau)\rangle$.

Lubasch et al

For the quantum algorithm we define $|f(t+\tau)\rangle = \lambda_0|\psi(\boldsymbol{\lambda})\rangle = \lambda_0\hat{U}(\boldsymbol{\lambda})|\mathbf{0}\rangle$ and $|f(t)\rangle = \tilde{\lambda}_0|\tilde{\psi}\rangle = \tilde{\lambda}_0\hat{\tilde{U}}|\mathbf{0}\rangle$ for every value of t. Then for each time step τ, the following cost function needs to be minimized:

$$\begin{aligned}\mathcal{C}(\lambda_0,\boldsymbol{\lambda}) &= \| \lambda_0|\psi(\boldsymbol{\lambda})\rangle - (\mathbb{1}+\tau\hat{O})\tilde{\lambda}_0|\tilde{\psi}\rangle\|^2 \qquad \text{(S4)}\\ &= |\lambda_0|^2 - 2\Re\{\lambda_0\tilde{\lambda}_0^*\langle\tilde{\psi}|(\mathbb{1}+\tau\hat{O})|\psi(\boldsymbol{\lambda})\rangle\} + \text{const.}\\ &= |\lambda_0|^2 - 2\Re\{\lambda_0\tilde{\lambda}_0^*\langle\mathbf{0}|\hat{\tilde{U}}^\dagger(\mathbb{1}+\tau\hat{O})\hat{U}(\boldsymbol{\lambda})|\mathbf{0}\rangle\} + \text{const.},\end{aligned}$$

Equate the time-evolved solution state with a Unitary transformation

2

Resultant superposition state that we would like to optimize for all time steps

3

Computationally minded questions relating to Item 3 from the previous slide

$$
\begin{aligned}
\mathcal{C}(\lambda_0, \boldsymbol{\lambda}) &= ||\lambda_0|\psi(\boldsymbol{\lambda})\rangle - (\mathbb{1} + \tau\hat{O})\tilde{\lambda}_0|\tilde{\psi}\rangle||^2 \qquad \text{(S4)} \\
&= |\lambda_0|^2 - 2\Re\{\lambda_0\tilde{\lambda}_0^*\langle\tilde{\psi}|(\mathbb{1} + \tau\hat{O})|\psi(\boldsymbol{\lambda})\rangle\} + \text{const.} \\
&= |\lambda_0|^2 - 2\Re\{\lambda_0\tilde{\lambda}_0^*\langle\mathbf{0}|\hat{\tilde{U}}^\dagger(\mathbb{1} + \tau\hat{O})\hat{U}(\boldsymbol{\lambda})|\mathbf{0}\rangle\} + \text{const.},
\end{aligned}
$$

- How many time steps of evolution does it take to reach the ground state of the cost function?
- How does the computational complexity of the optimization procedure depend upon the initial conditions of the PDE that is imposed through the initial solution state?

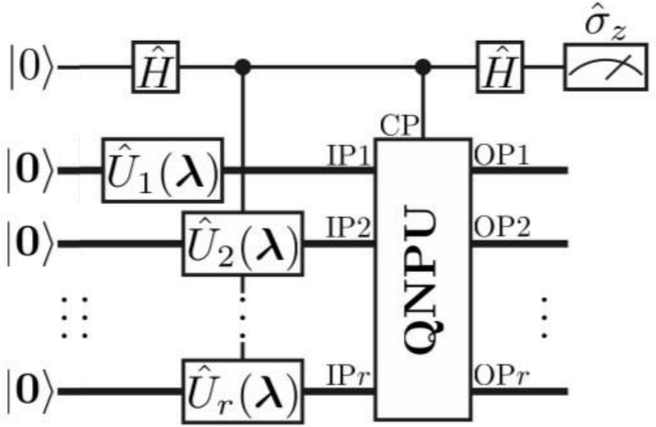
Lubasch et al

$$\mathcal{C}(\lambda_0, \boldsymbol{\lambda}) = \| \lambda_0 |\psi(\boldsymbol{\lambda})\rangle - (\mathbb{1} + \tau \hat{O}) \tilde{\lambda}_0 |\tilde{\psi}\rangle \|^2 \tag{S4}$$
$$= |\lambda_0|^2 - 2\Re\{\lambda_0 \tilde{\lambda}_0^* \langle \tilde{\psi} | (\mathbb{1} + \tau \hat{O}) | \psi(\boldsymbol{\lambda})\rangle\} + \text{const.}$$
$$= |\lambda_0|^2 - 2\Re\{\lambda_0 \tilde{\lambda}_0^* \langle \mathbf{0} | \hat{\tilde{U}}^\dagger (\mathbb{1} + \tau \hat{O}) \hat{U}(\boldsymbol{\lambda}) | \mathbf{0}\rangle\} + \text{const.},$$

Cost function ground state optimization

Correspondence with quantum circuits

- Read from left to right
- PDE nonlinearities are processed by the Quantum Nonlinear Processing Unit (QNPU)
- We obtain "readout" from the quantum circuit after performing time evolution for a fixed number of steps



Further spelling out quantum circuit components

5

Hadamard test

1

0 state

2

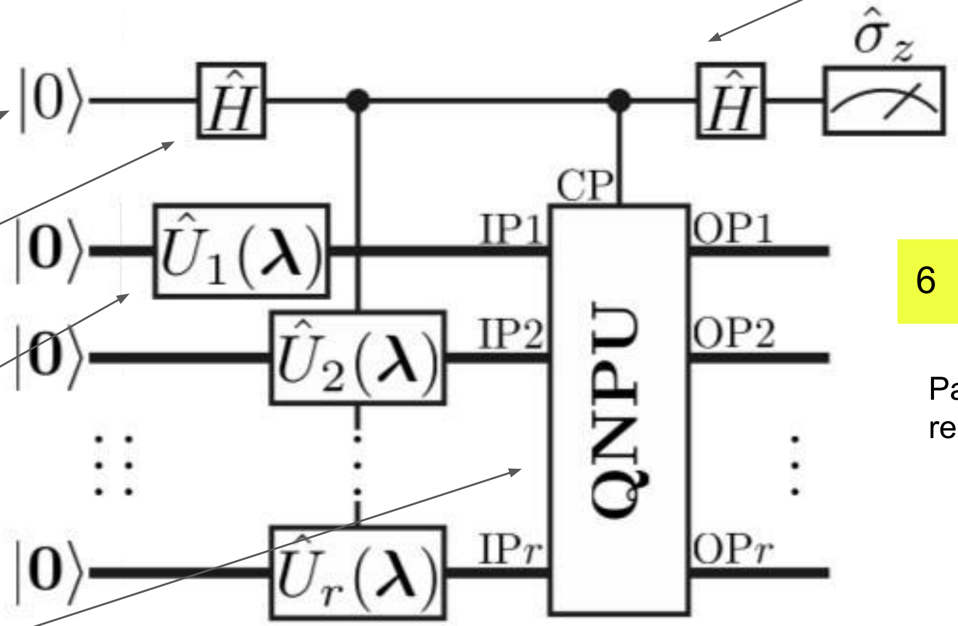
Hadamard test

3

Unitary transformation

4

Nonlinear processing unit



6

Pauli z-rotation for readout

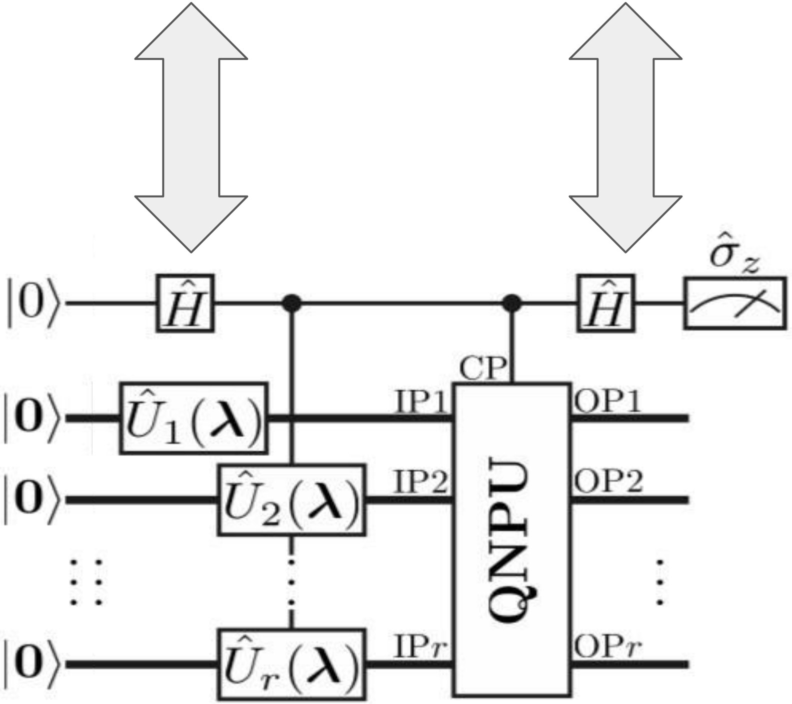
McMahon Lab

Lubasch et al

$$\mathcal{C}(\lambda_0, \boldsymbol{\lambda}) = \|\lambda_0|\psi(\boldsymbol{\lambda})\rangle - (\mathbb{1} + \tau\hat{O})\tilde{\lambda}_0|\tilde{\psi}\rangle\|^2 \tag{S4}$$
$$= |\lambda_0|^2 - 2\Re\{\lambda_0\tilde{\lambda}_0^*\langle\tilde{\psi}|(\mathbb{1} + \tau\hat{O})|\psi(\boldsymbol{\lambda})\rangle\} + \text{const.}$$
$$= |\lambda_0|^2 - 2\Re\{\lambda_0\tilde{\lambda}_0^*\langle\mathbf{0}|\hat{\tilde{U}}^\dagger(\mathbb{1} + \tau\hat{O})\hat{U}(\boldsymbol{\lambda})|\mathbf{0}\rangle\} + \text{const.},$$

<u>Backpropagation loop 1</u>

CF - QNPU



<u>Backpropagation loop 2</u>

QNPU - CF

Initializing, and variationally modifying, parameters for generating solutions to other PDE IVPs

Nonlinearity space of 8 PDEs

- Navier-Stokes
- Einstein
- Maxwell
- Boussinesq-type
- Lin-Tsien
- Camassa-Holm
- Drinfeld-Sokolov-Wilson (DSW)
- Hunter-Saxton

Initializing, and variationally modifying, parameters for generating solutions to other PDE IVPs

Cost function ground state space

Formulate cost function

Nonlinearity space of 8 PDEs

- Transform equation into a quantum state that we would like to perform time evolution on
- Form resultant quantum superposition for a Cost Function

Initializing parameters for generating solutions to other PDE IVPs

Cirq quantum circuit evaluation

Quantum circuit
automation

Nonlinearity space

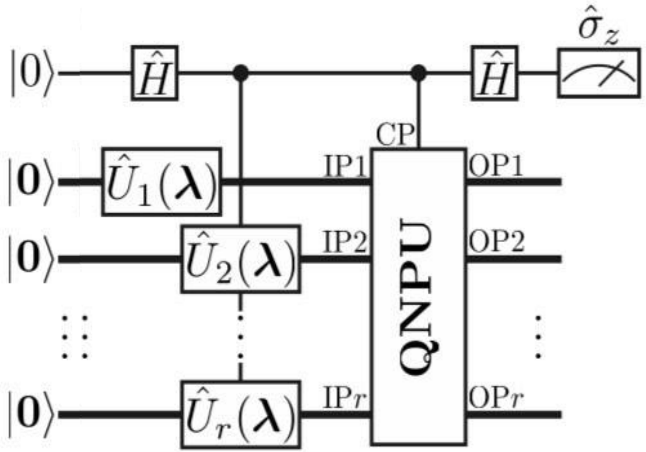
+

Cost function ground state space

Assemble solution approximations

We formulate our cost functions with a Python open-source package from Google

Automate evaluating the cost function



Assemble solution approximations

For different PDEs, each respective cost function can have a different number of terms that have to be simultaneously optimized

Optimizer pool

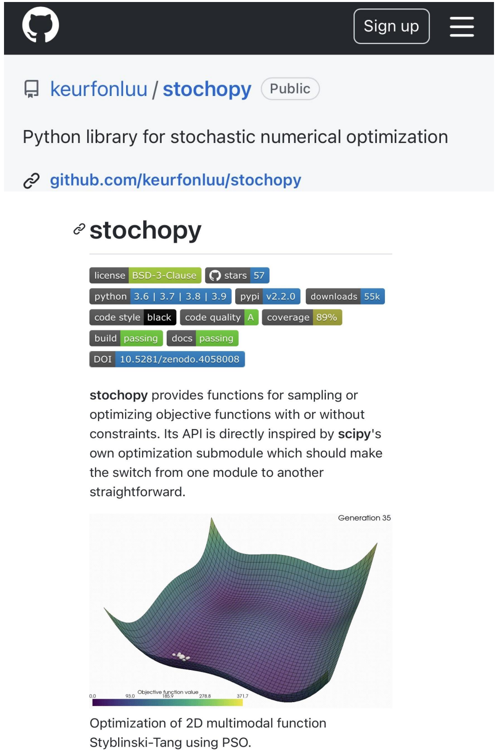
- Imfil
- Snobfit

Nevergrad

For the most part the deterministic optimizers shown here perform the best for most PDEs from the list shown

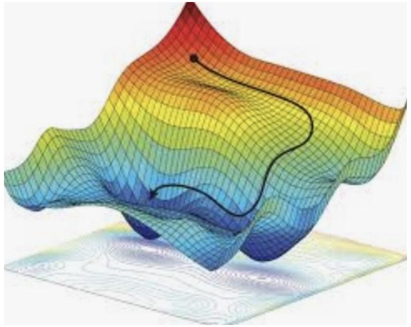
Optimizer pool

However, Stochastic and Gradient-based optimizers perform well in situations where deterministic optimizers require many time steps to reach the ground state



Stochastic optimizers

- Particle-swarm
- Competitive Particle-swarm
- Covariance matrix adaptation
- Neighborhood algorithm
- VD-CMA



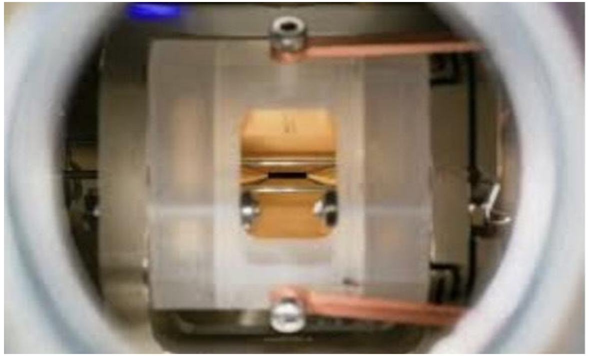
Gradient-based optimizer

- Sequential least squares

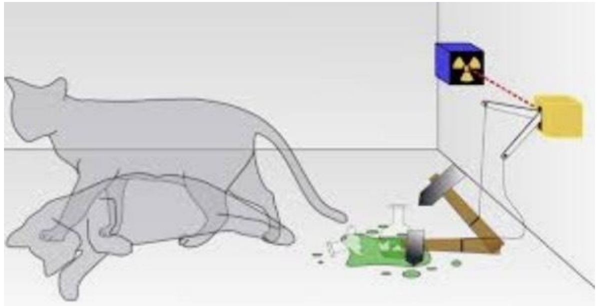
McMahon Lab

Measurement extraction for approximating PDE solutions

Quantum readout ⟺ Classically computed PDE solution

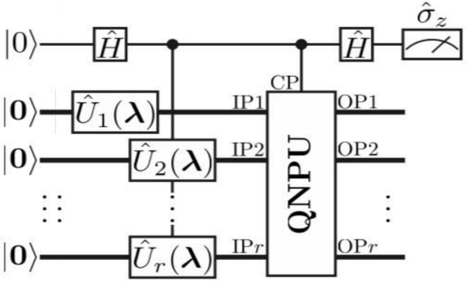


Physics world



Nature

Collapsing of the wavefunction with Pauli-z rotation



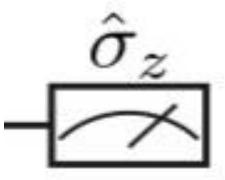
6

Pauli z-rotation for readout

McMahon Lab

For our implementation…

Sample a realization of one possible solution from the state vector




```
# allocate solution approximations
SA_1[i-1,0]=opt_2[10*i]
SA_2[i-1,0]=_2_2_opt[10*i]
SA_3[i-1,0]=_2_3_opt[10*i]
SA_4[i-1,0]=_2_4_opt[10*i]
SA_5[i-1,0]=_2_5_opt[10*i]
SA_6[i-1,0]= _2_6_opt[10*i]
SA_7[i-1,0]=_2_7_opt[10*i]
SA_8[i-1,0]=_2_8_opt[10*i]
```

Namely, only pass one scalar to the solution array from the set of all possibilities prepared in the ansatz state at the beginning of time evolution

Several components of the algorithm that we have discussed relate to the following areas

1

Machine Learning

- Formulating the cost function
- The cost function penalizes deviation from the optimal trajectory to reach the ground state

2

Classically determined PDE solving methods

- The quantum algorithm can produce solutions from redout in quantum circuits
- It is of interest to determine whether the Quantum algorithm can perform on a similar level as a Classical algorithm would

Several components of the algorithm that we have discussed relate to the following areas

1

Machine Learning

- Formulating the cost function
- The cost function penalizes deviation from the optimal trajectory to reach the ground state

2

Classically determined PDE solving methods

- The quantum algorithm can produce solutions from redout in quantum circuits
- It is of interest to determine whether the Quantum algorithm can perform on a similar level as a Classical algorithm would

3

Quantum Information

- We are interested in determining the best ansatz, which coincides with any possible initial conditions for a PDE IVP
- Relatedly, properties of the ansatz, which initially explores the Hilbert space which can maximize coverage

Several components of the algorithm that we have discussed relate to the following areas

1

Machine Learning

- Formulating the cost function
- The cost function penalizes deviation from the optimal trajectory to reach the ground state

2

Classically determined PDE solving methods

- The quantum algorithm can produce solutions from redout in quantum circuits
- It is of interest to determine whether the Quantum algorithm can perform on a similar level as a Classical algorithm would

3

Quantum Information

- We are interested in determining the best ansatz, which coincides with any possible initial conditions for a PDE IVP
- Relatedly, properties of the ansatz, which initially explores the Hilbert space which can maximize coverage

4

Computational Complexity

- Our ansatz being <u>classically simulable in polynomial time</u> allows for us to run numerical experiments for assessing performance
- However, it is still of interest to develop other ansatzae for exploring the Hilbert space in the future

Depending upon the interest/application…

4

Computational Complexity

2

Classically determined PDE solving methods

One can answer questions relating to:

- Whether quantum algorithms with polynomial runtime can generate solutions within a desired error tolerance
- Adjustment of variational parameters used to obtain readout

Alternatively, for other interests…

1

Machine Learning

4

Computational Complexity

One can answer questions relating to:

- Which optimizer fluctuates the least when approaching the ground state
- The rate of convergence of the optimizer to the ground state (ie, the number of steps in the time evolution after which the solution approximation no longer fluctuates)

Nonlinearity space

- Navier-Stokes
- Einstein
- Maxwell
- Boussinesq-type
- Lin-Tsien
- Camassa-Holm
- Drinfeld-Sokolov-Wilson (DSW)
- Hunter-Saxton

Let's focus only on this PDE

Nonlinearity space of 8 PDEs

Camassa-Holm PDE

$$u_t + 2\kappa u_x - u_{xxt} + 3uu_x = 2u_x u_{xx} + uu_{xxx}$$

Impose initial conditions on two derivatives of admissible solutions u

$$u_t + 2\kappa u_x - u_{xxt} + 3uu_x = 2u_x u_{xx} + uu_{xxx}$$

To execute the algorithm:

- Form a cost function for the PDE
- Explicitly code up the form of the cost function, with all expectation values, with Input & Output ports to the Nonlinear Processing Unit
- Repeatedly evaluate the cost function with values generated from Cirq

→

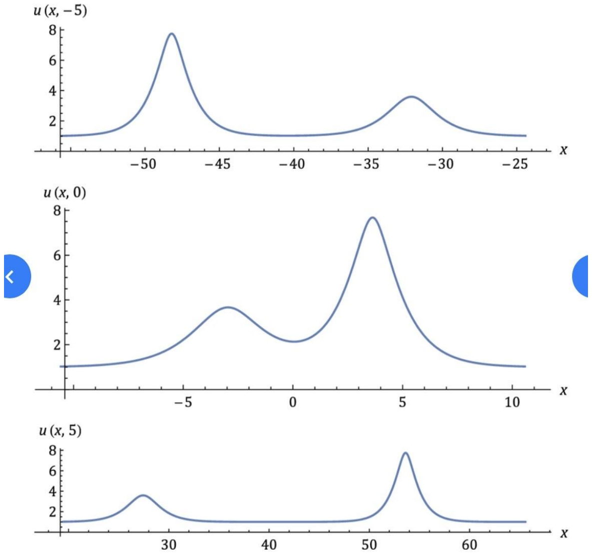
- <u>500, 2500, 8000, 15,000 & ~20,000</u> time steps of evolution

We turn towards classically determined solutions from the literature

DOI: 10.1016/J.JDE.2004.09.007
Corpus ID: 119989481

Traveling wave solutions of the Camassa-Holm equation

J. Lenells • Published 15 October 2005 • Mathematics • Journal of Differential Equations



Snapshots of the two-soliton solution of the Camassa–Holm equation (CH), for three values of

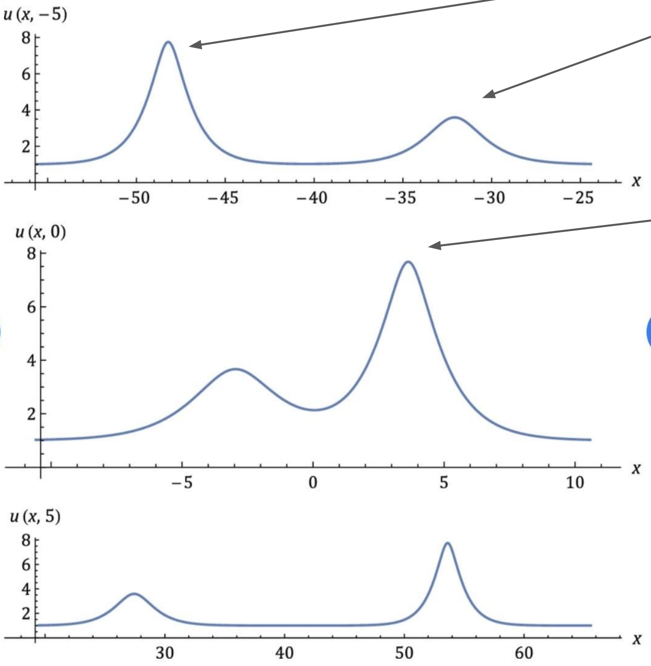
DOI: 10.1016/J.JDE.2004.09.007
Corpus ID: 119989481

Traveling wave solutions of the Camassa-Holm equation

J. Lenells • Published 15 October 2005 • Mathematics • Journal of Differential Equations

Different points of time at which the traveling wave attains maxima

For other solitons, other global maxima can be attained



Snapshots of the two-soliton solution of the Camassa–Holm equation (CH), for three values of

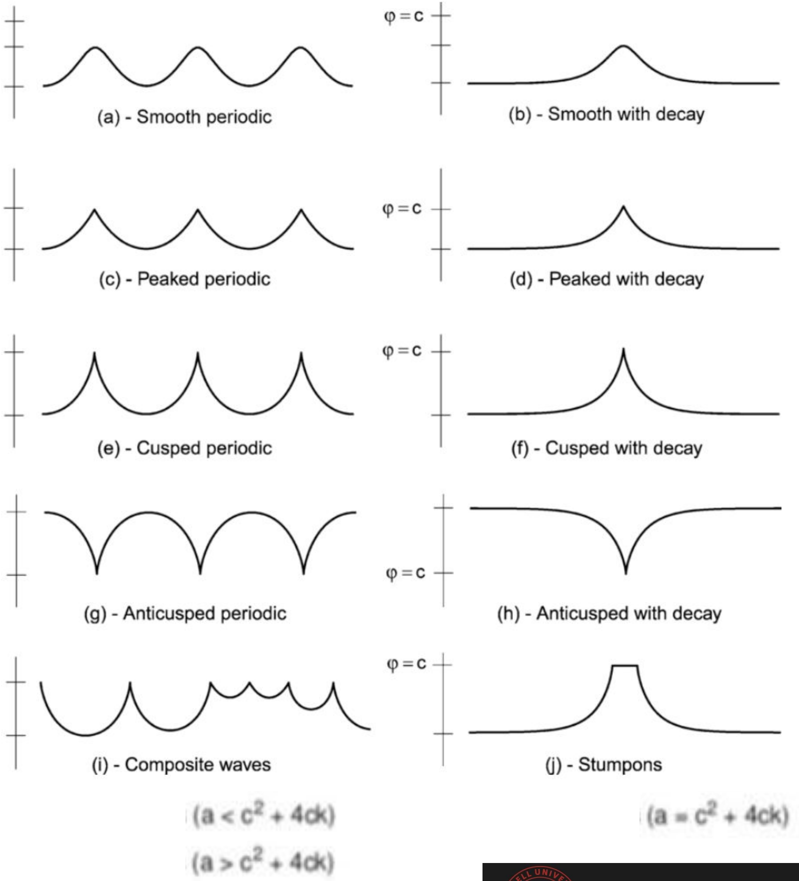
Another reference, more recently from 2020, also provides characteristics of solutions to the Camassa-Holm equation

Camassa–Holm Cuspons, Solitons and Their Interactions via the Dressing Method

Article | Full-text available

Feb 2020

Rossen I. Ivanov · Tony Lyons · Nigel Orr



Jonathan Lenells, Traveling wave solutions of the D-P equation (2005)

More details from 2020 reference

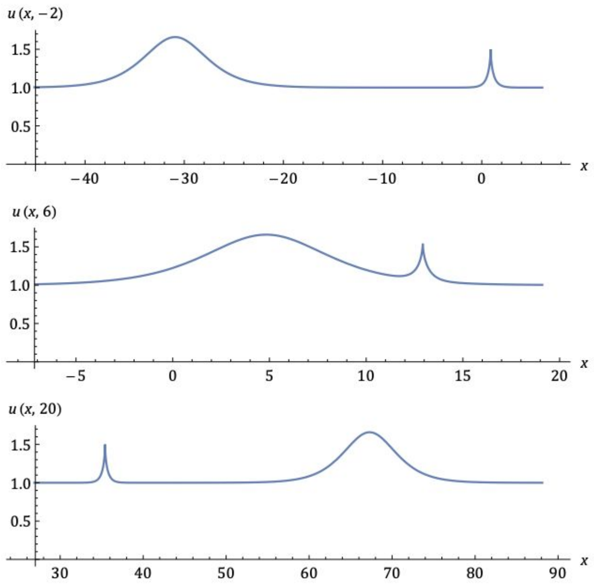
Camassa–Holm Cuspons, Solitons and Their Interactions via the Dressing Method

Article | Full-text available

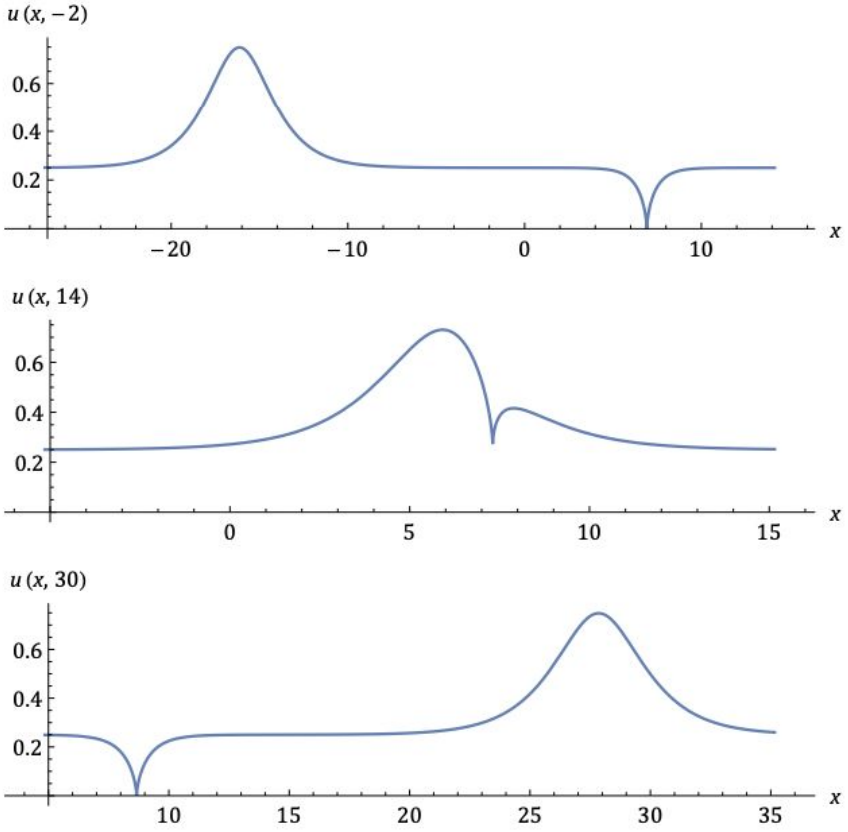
Feb 2020

Rossen I. Ivanov · Tony Lyons · Nigel Orr

24 ROSSEN IVANOV, TONY LYONS, AND NIGEL ORR



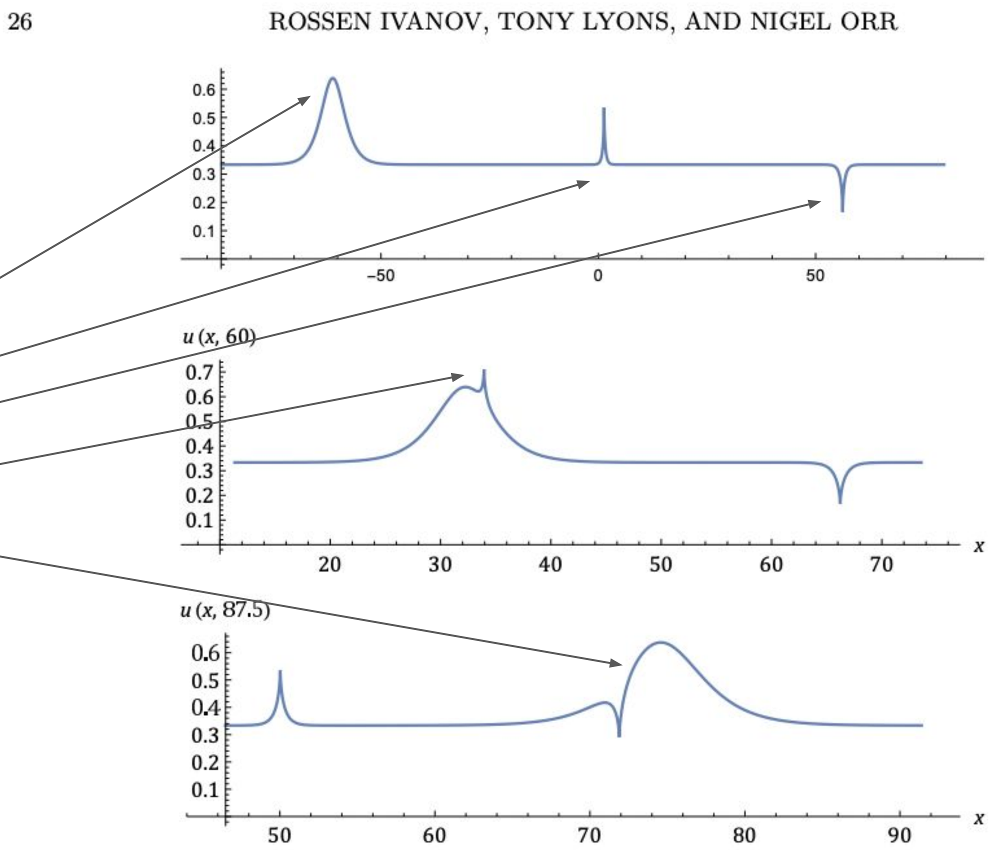
More details from 2020 reference



McMahon Lab

More details from 2020 reference

Camassa-Holm solutions can interpolate between smooth and cuspon behaviors



Camassa–Holm Cuspons, Solitons and Their Interactions via the Dressing Method

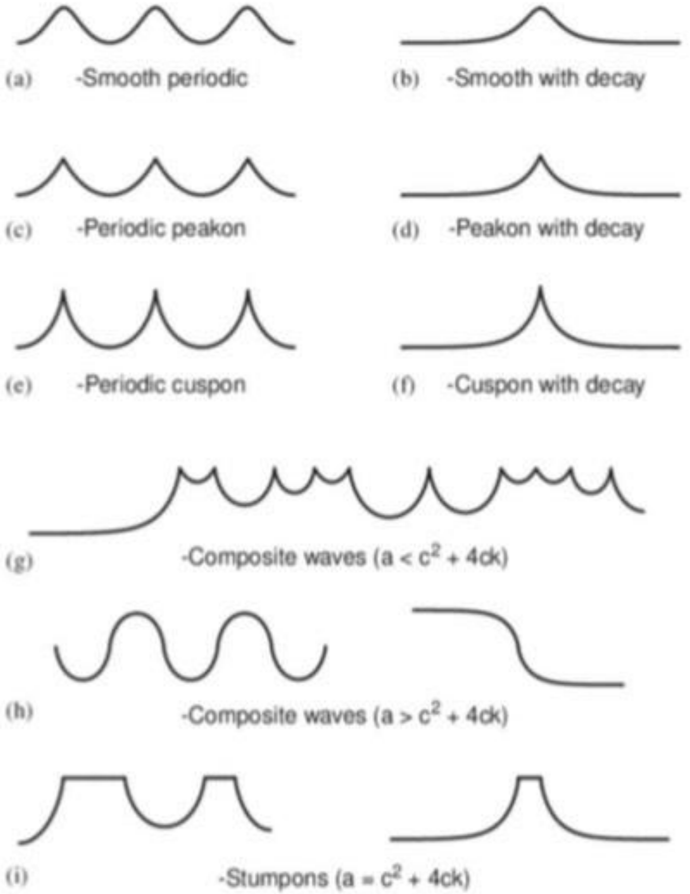
Article Full-text available

Feb 2020

Rossen I. Ivanov · Tony Lyons · Nigel Orr

From classically determined solutions to the Camassa-Holm equation,

- Can we recover behavior from (a)?
- Can we recover behavior from (b) or (d)?
- Or, more generally, can we recover behavior simultaneously from any of the categories that the authors from the 2020 reference provide?
  - Composite waves (g) + Smooth periodic
  - Composite waves (g) + Periodic cuspon
  - Composite waves (g) + Smooth with decay
  - Smooth with decay + Smooth periodic



McMahon Lab

Additional Camassa-Holm solutions that are unbounded

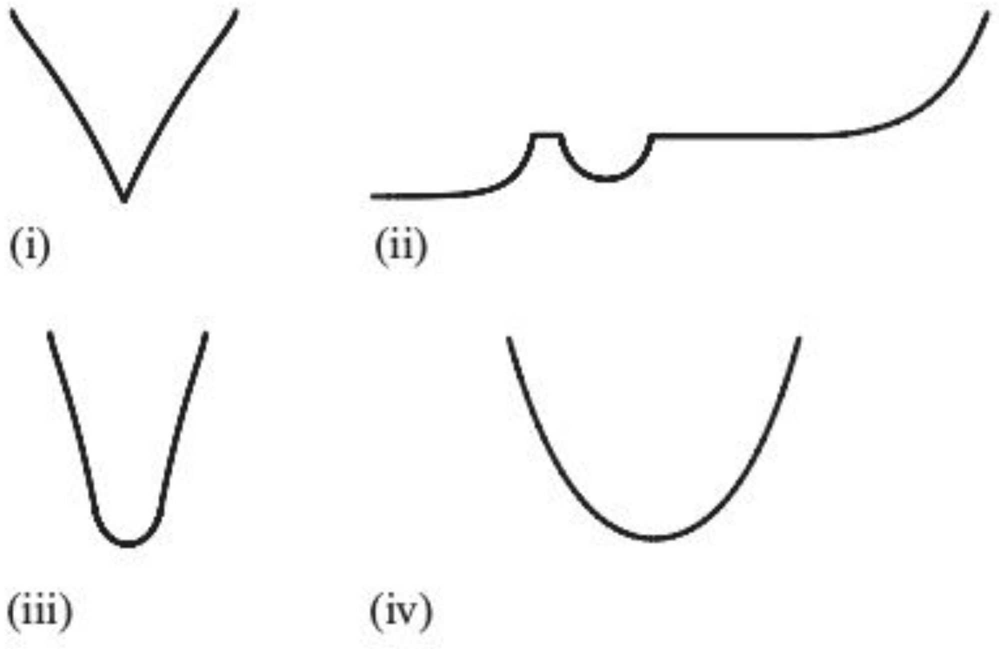
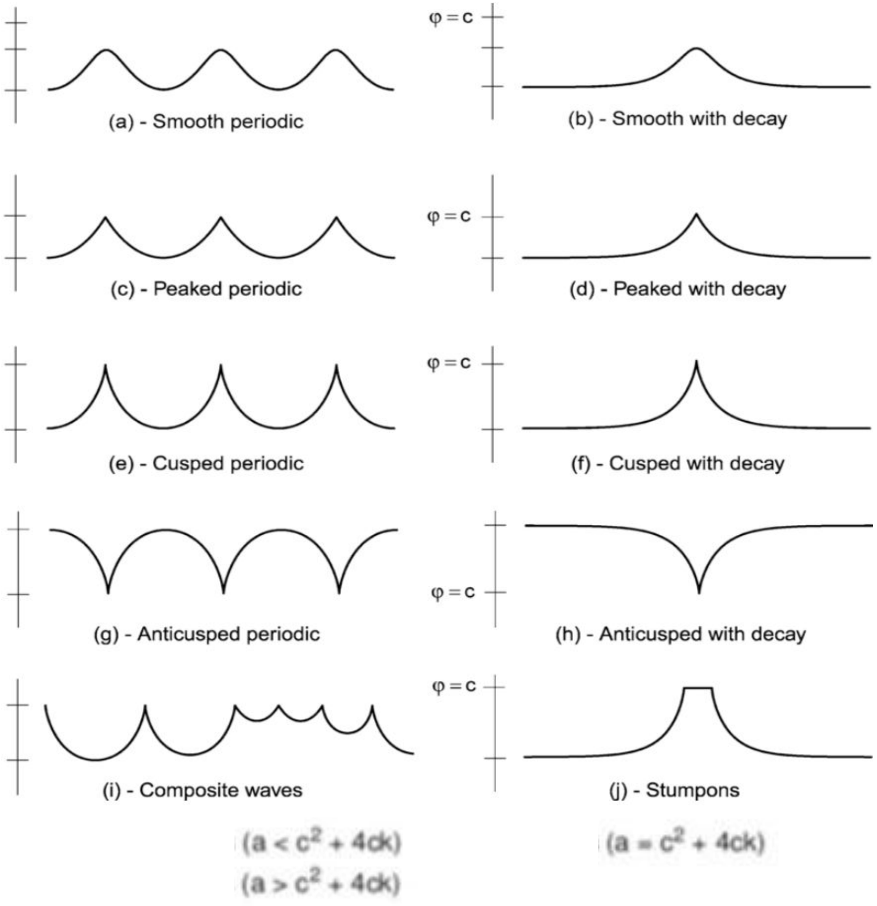


Fig. 4. Unbounded traveling waves of the Camassa–Holm equation.

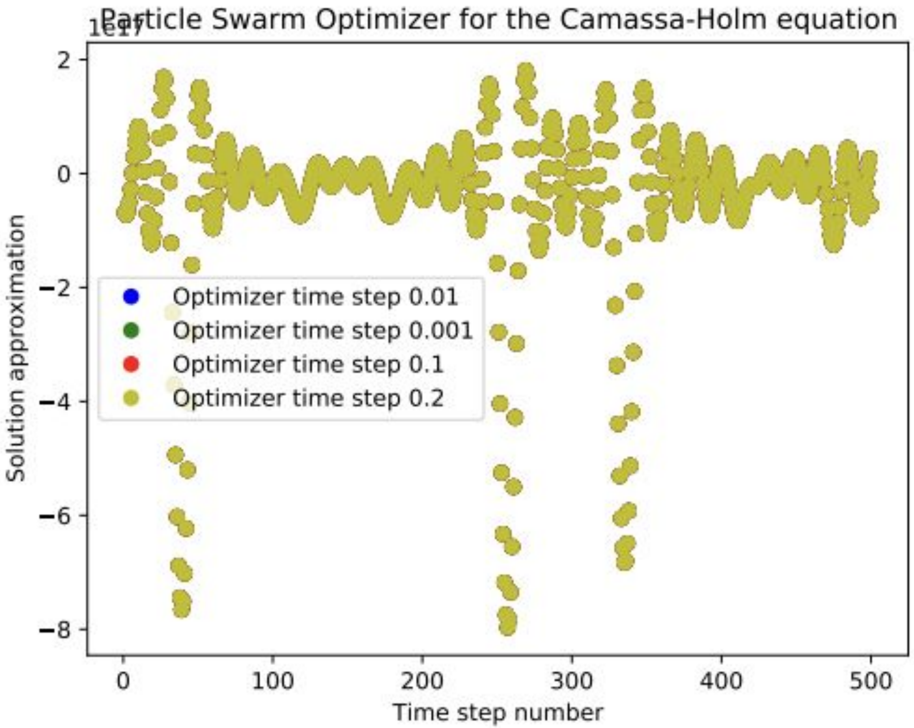
Goal for remainder of presentation

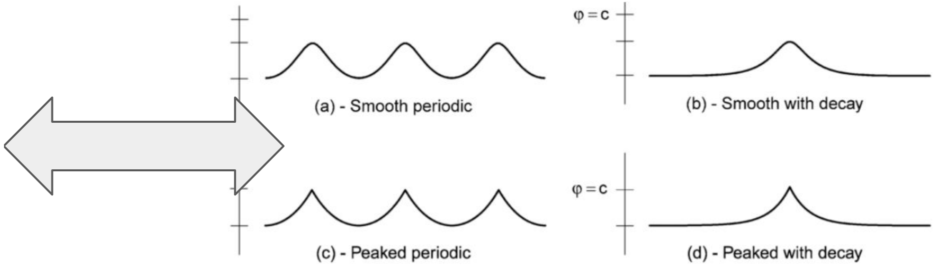
Which qualitative behaviors of Camassa-Holm solutions can be captured by the variational quantum algorithm?



McMahon Lab

Measurement extraction discussed in the APS April 2022 presentation

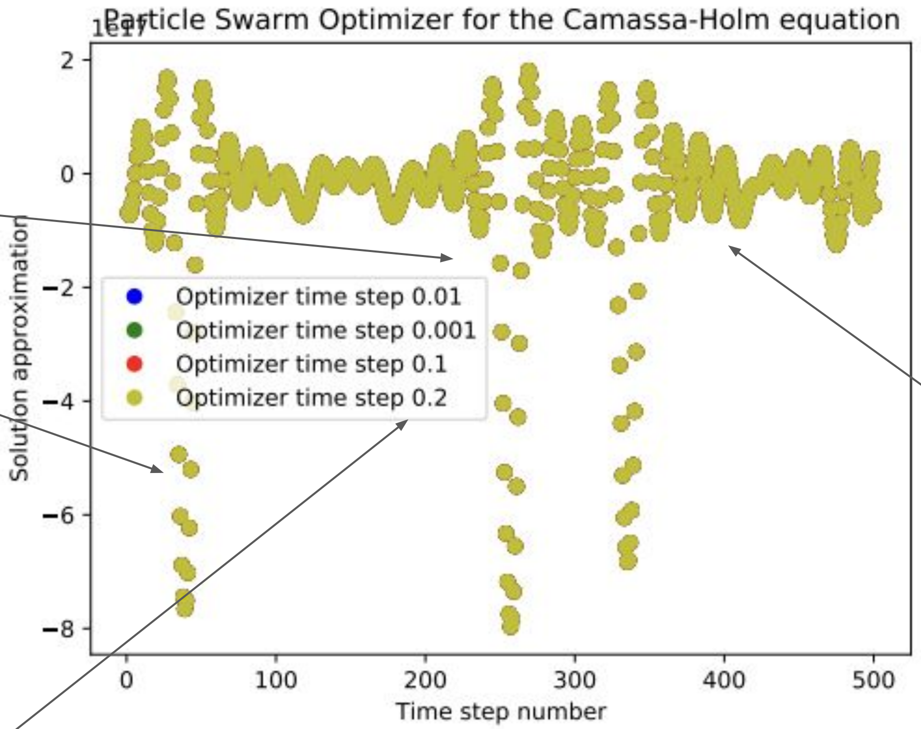




Resemblance with classically determined solutions

Measurement extraction discussed in the APS April 2022 presentation

PDE solution approximation <u>asymptotes</u> towards negative infinity



<u>Closer to the middle of the period of motion</u>, readout for solution approximations continue to fluctuate

For PDEs with more complicated nonlinearities, varying the time step of an optimizer <u>does not alter resolution</u> of the extracted measurements

McMahon Lab

A primary question of interest for simulating solutions to the Camassa-Holm PDE

How are features of standing wave and solitons realized? (ie, which nonlinearities are dominant?)

How is the time evolution dependent upon different initial conditions?

How much of the Hilbert space can we cover?

For the most part, we make use of the following optimizer

facebook research

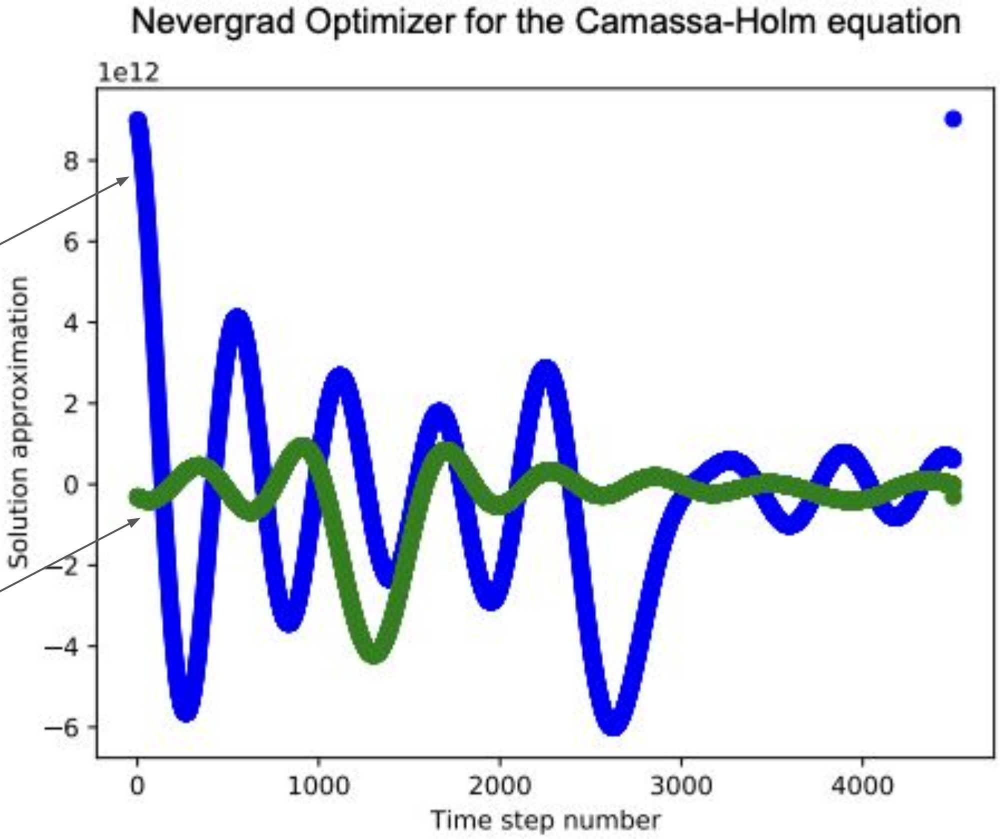
Nevergrad

One example of a solution simulated for time steps of evolution

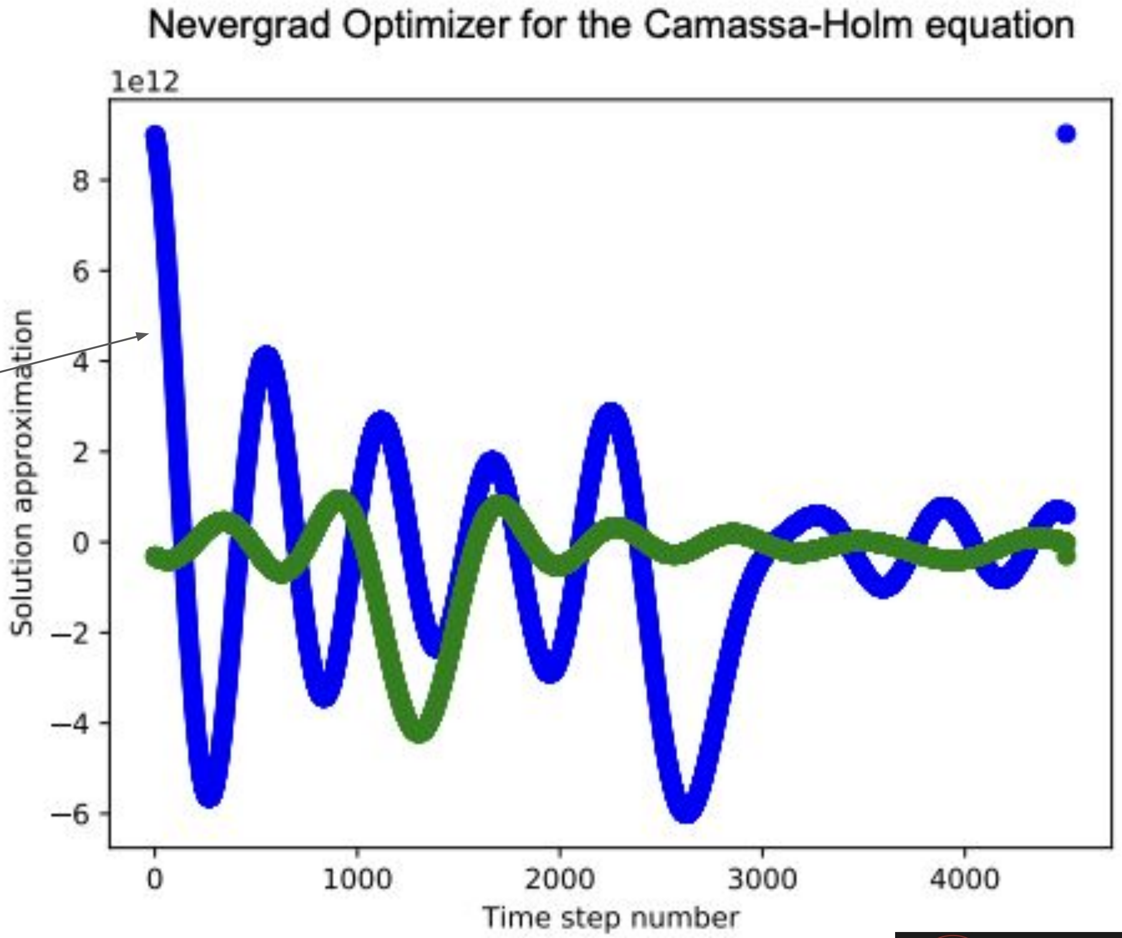
We form approximations to standing wave solutions of the Camassa-Holm equation

Approximation 1

Approximation 2



Taking a closer look

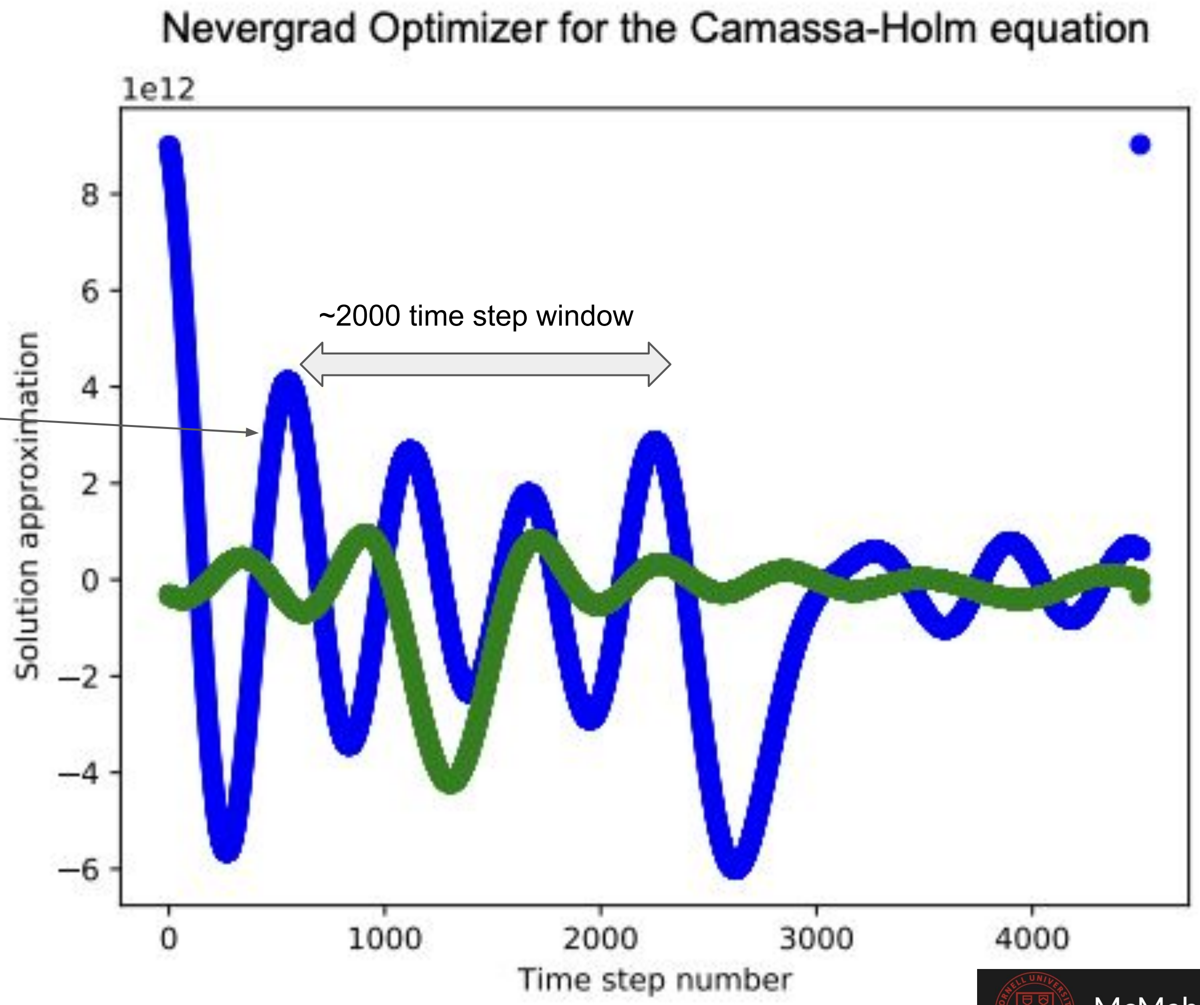


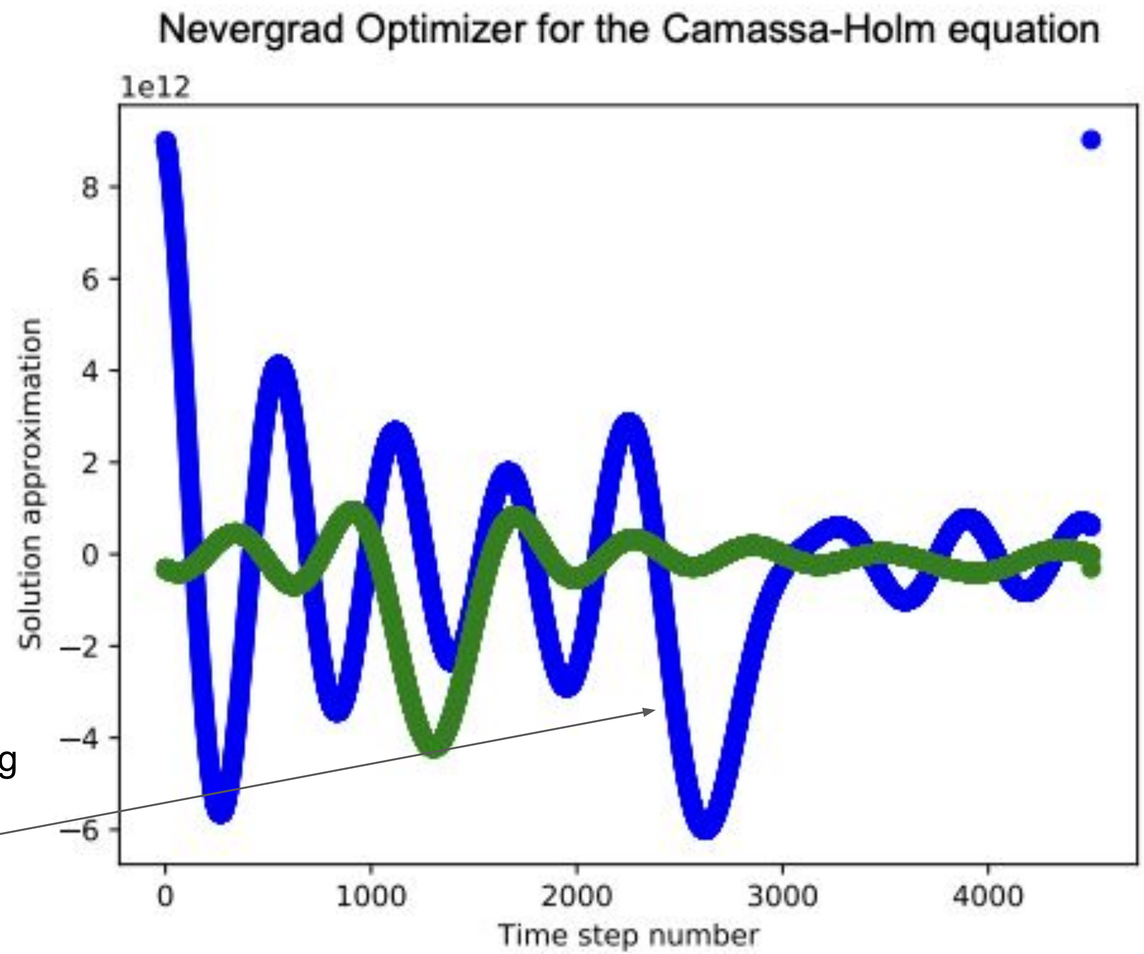
Approximation 1

Solution approximation for the traveling wave decreases in amplitude after the first period of motion

Approximation 1

Magnitude of approximation to standing wave stabilizes





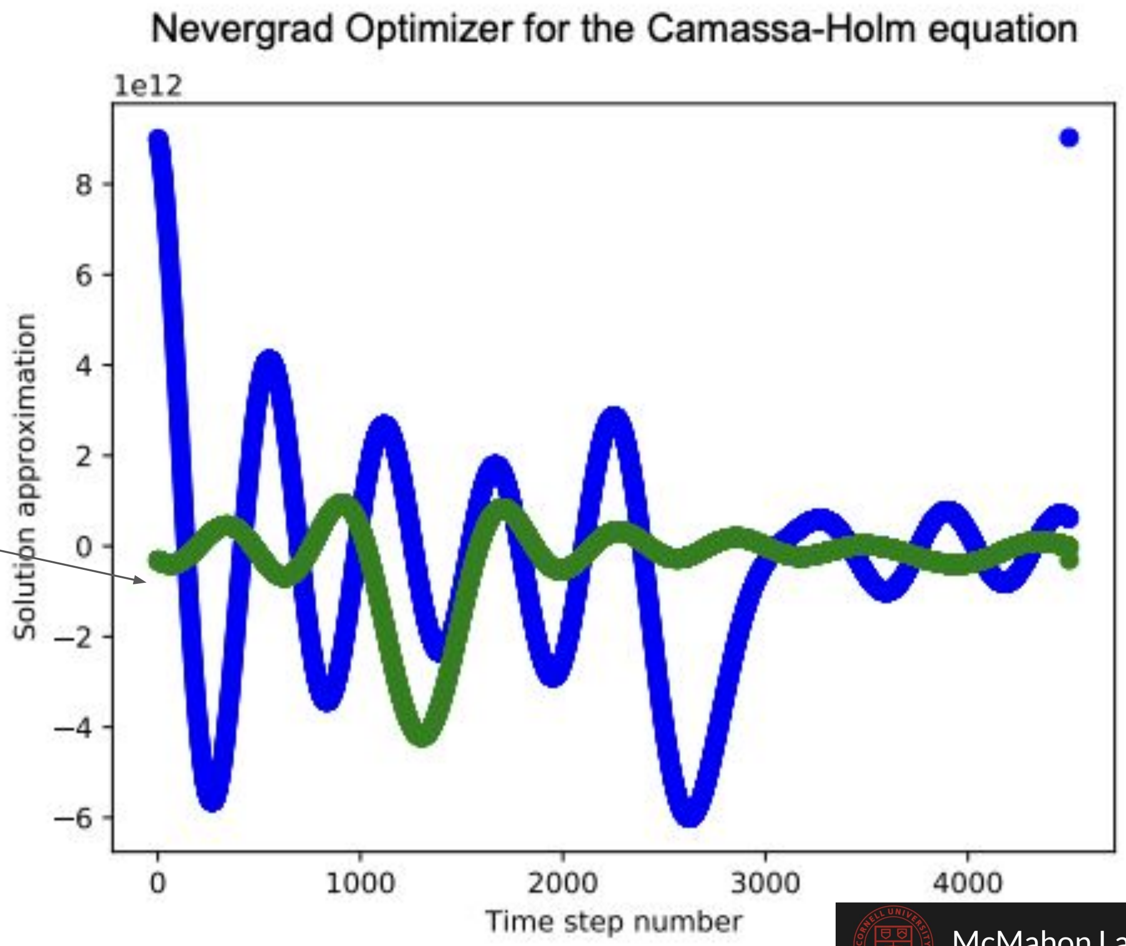
Approximation 1

Magnitude of approximation to standing wave decreases significantly around Time Step 3000, and, afterwards stabilizes

Besides Approximation 1…

<u>Approximation 2</u>

Initial condition of standing wave velocity is set to be almost 0

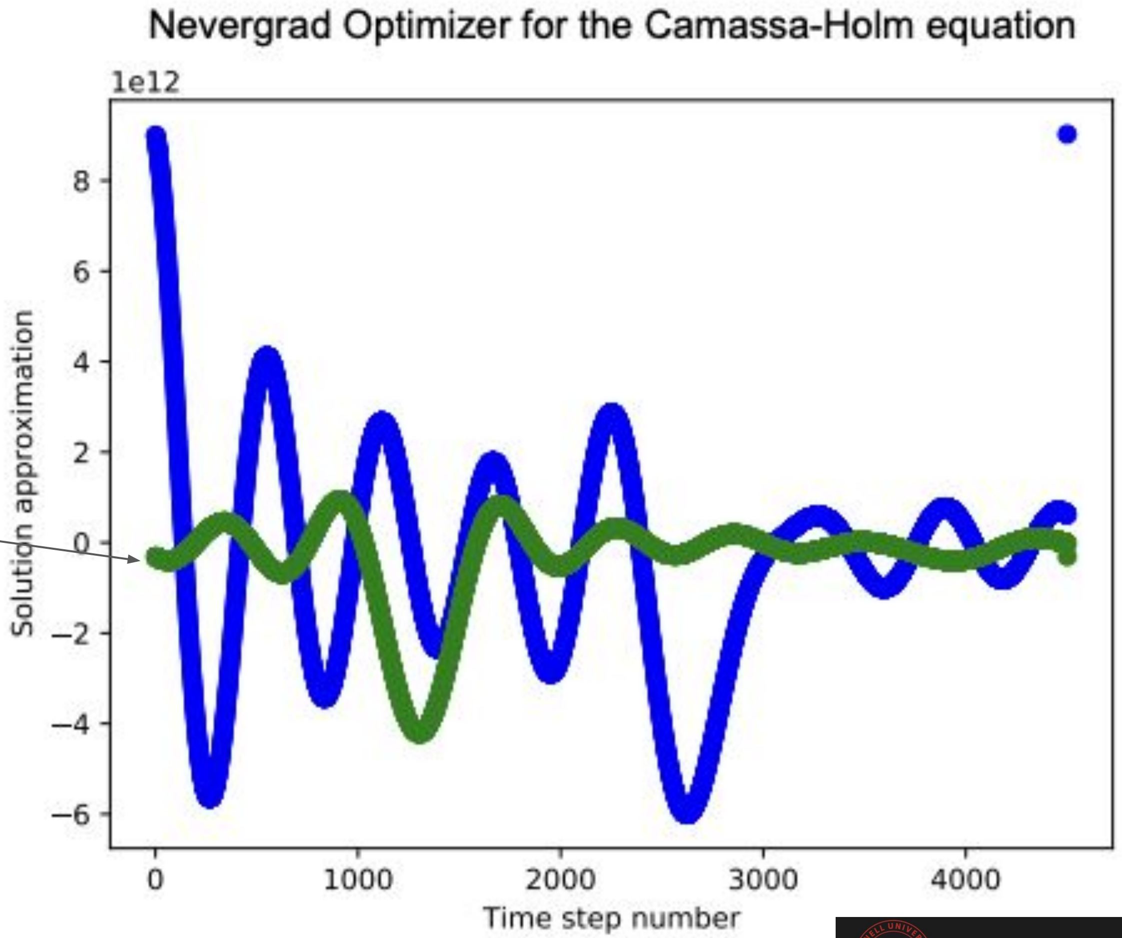


McMahon Lab

Besides Approximation 1…

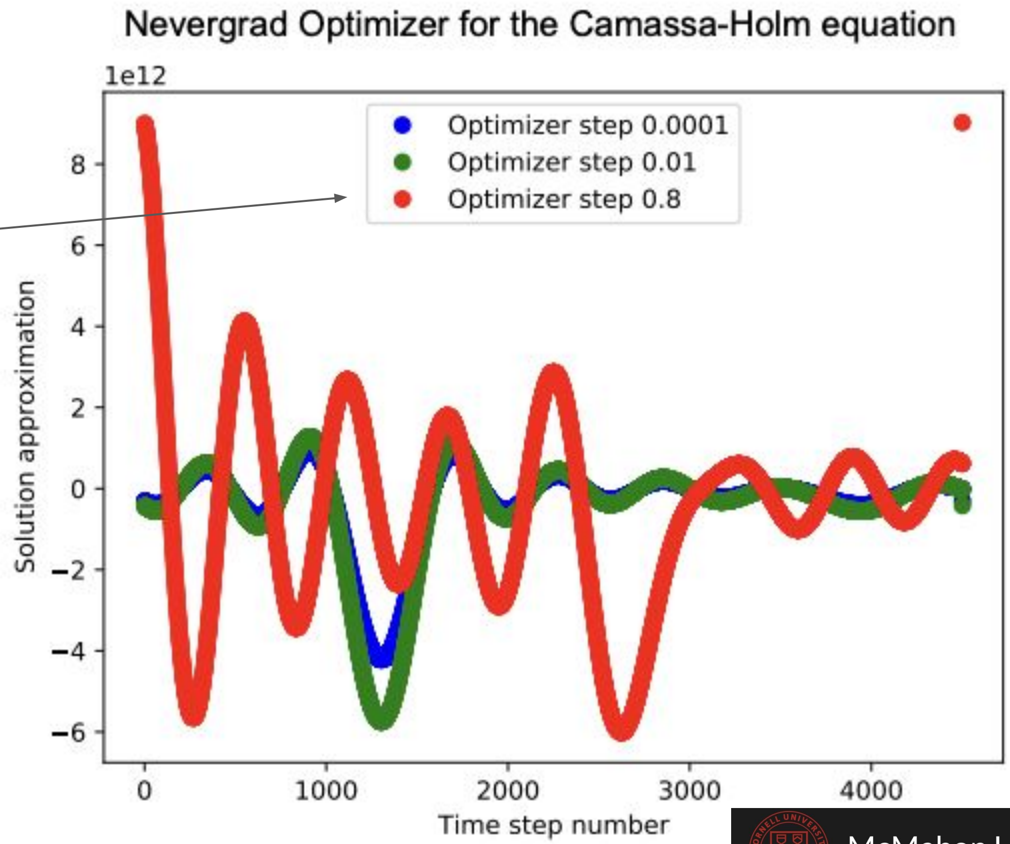
<u>Approximation 2</u>

Majority of the relaxation dynamics to the ground state from the cost function fluctuates above and below 0



With other results from the Nevergrad Optimizer, we observe:

Obtain other solutions by running the Optimizer for a range of time step magnitudes

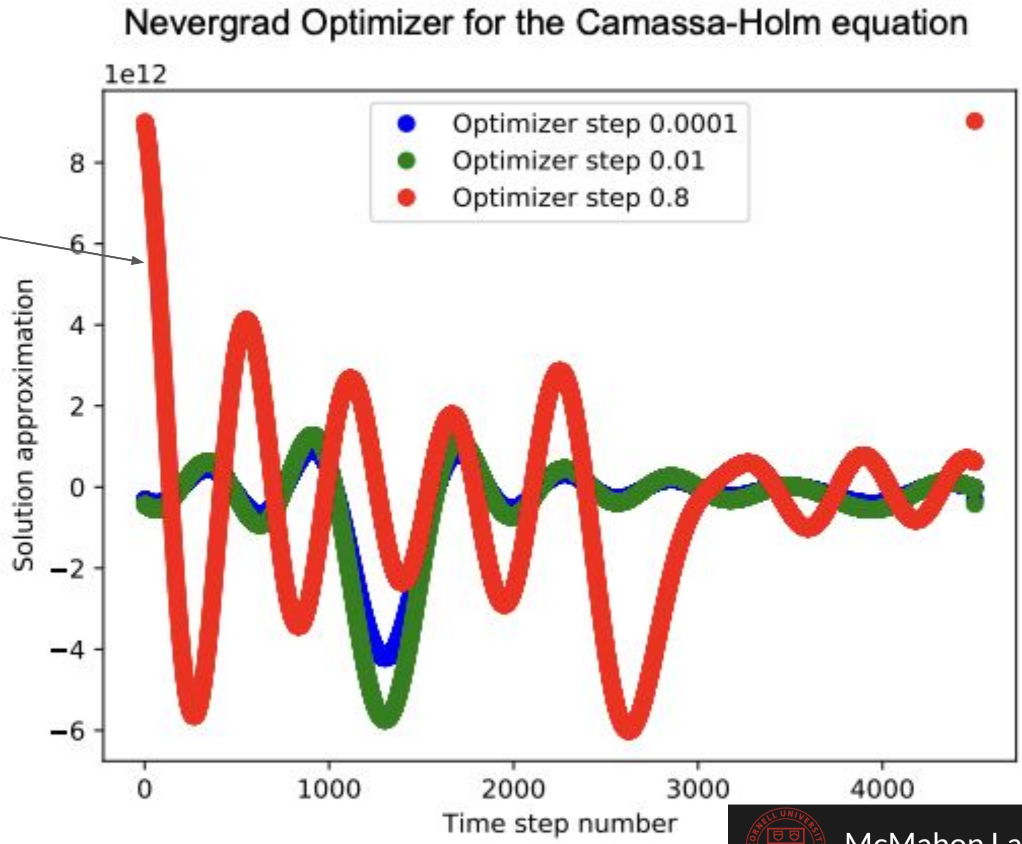


McMahon Lab

More specifically:

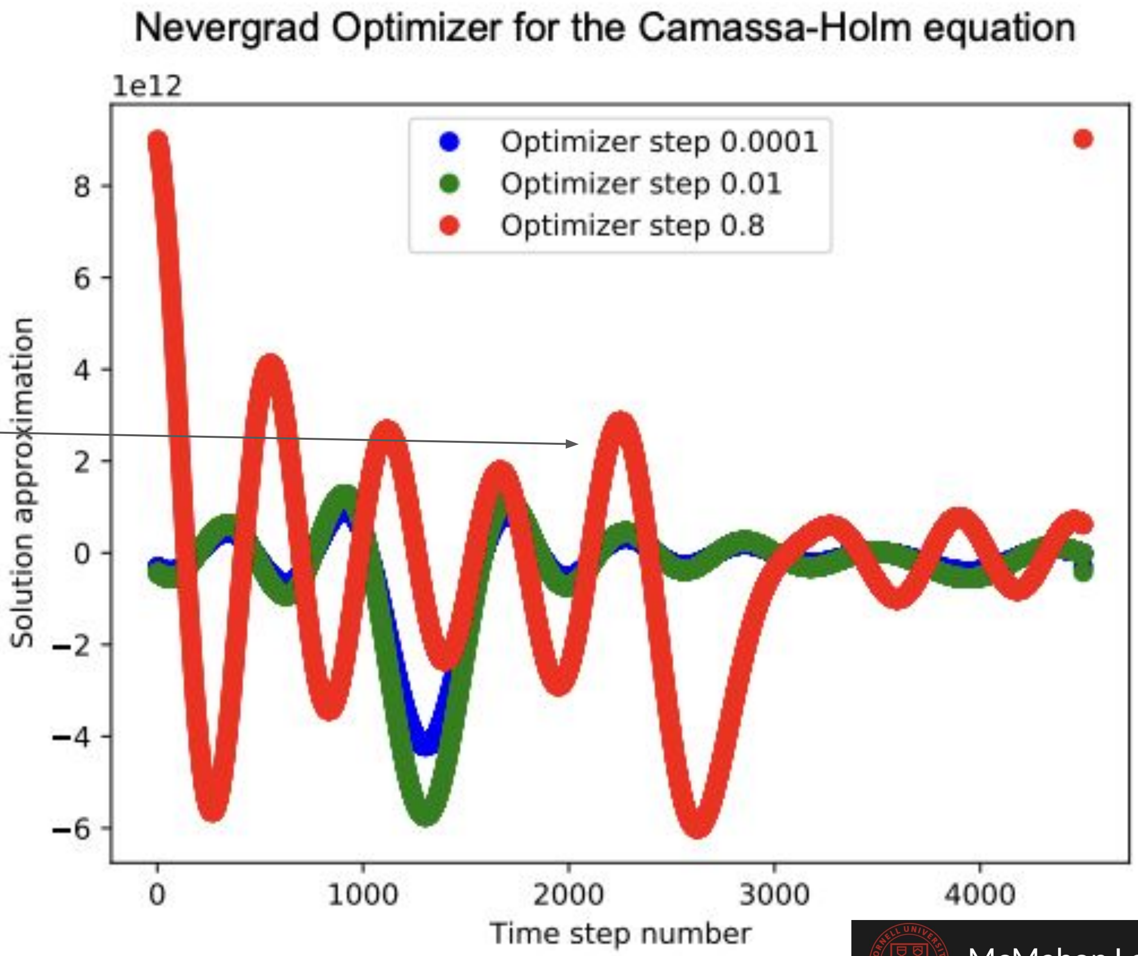
<u>Approximation 1</u>

Initial condition of solution is set to a high velocity



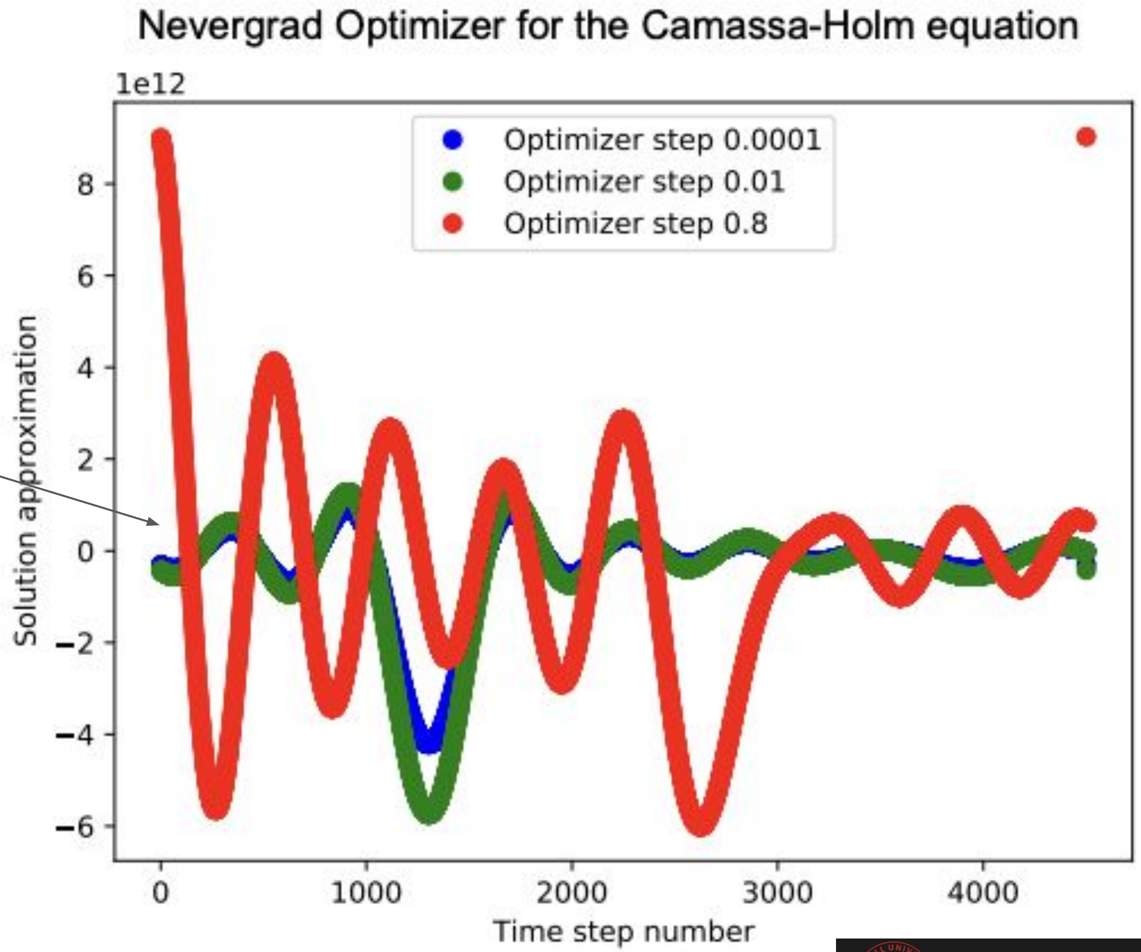
Approximation 1

- Similar relaxation dynamics to previously shown
- Solution approximation captures features of Standing wave as the amplitude of motion decreases with the number of time steps



Approximations 2 & 3

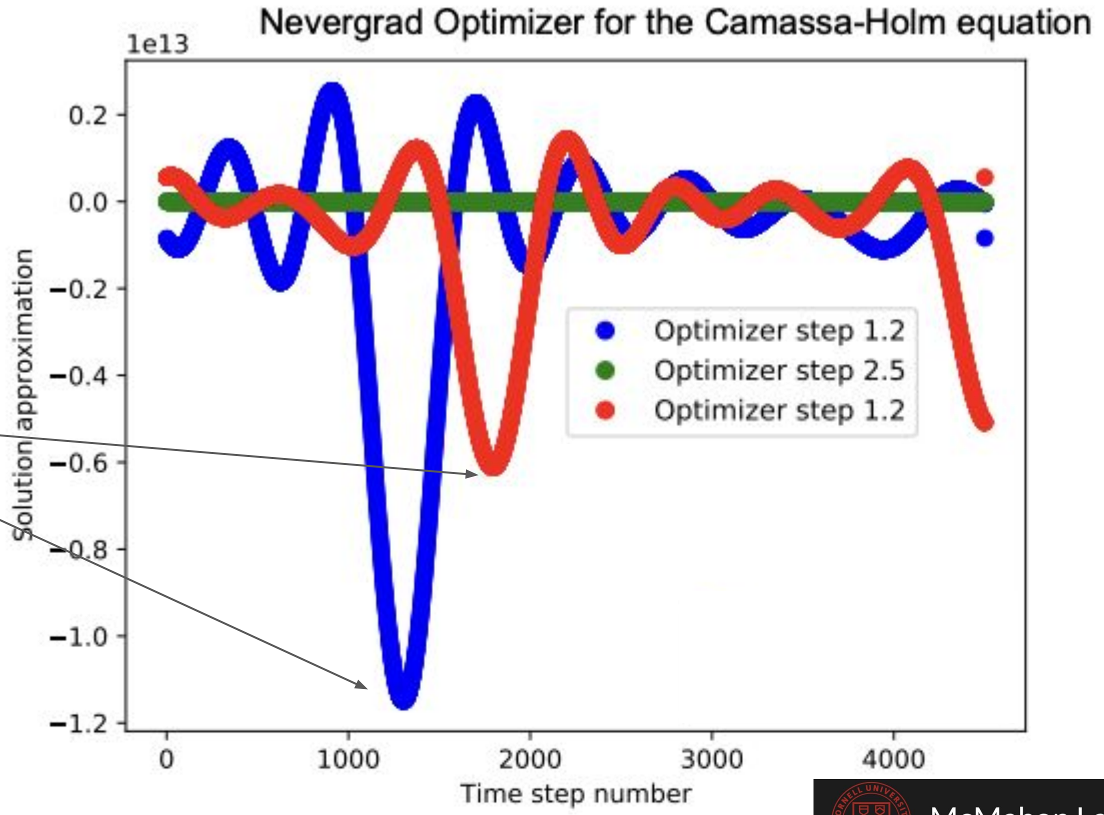
Making use of the Nevergrad Optimizer with time steps, of respective size 0.001 and 0.1, exhibits little difference in minimizing to the ground state



McMahon Lab

Regardless, Hilbert space coverage can differ more widely for different ZGR-QFT ansatzae, and ICs

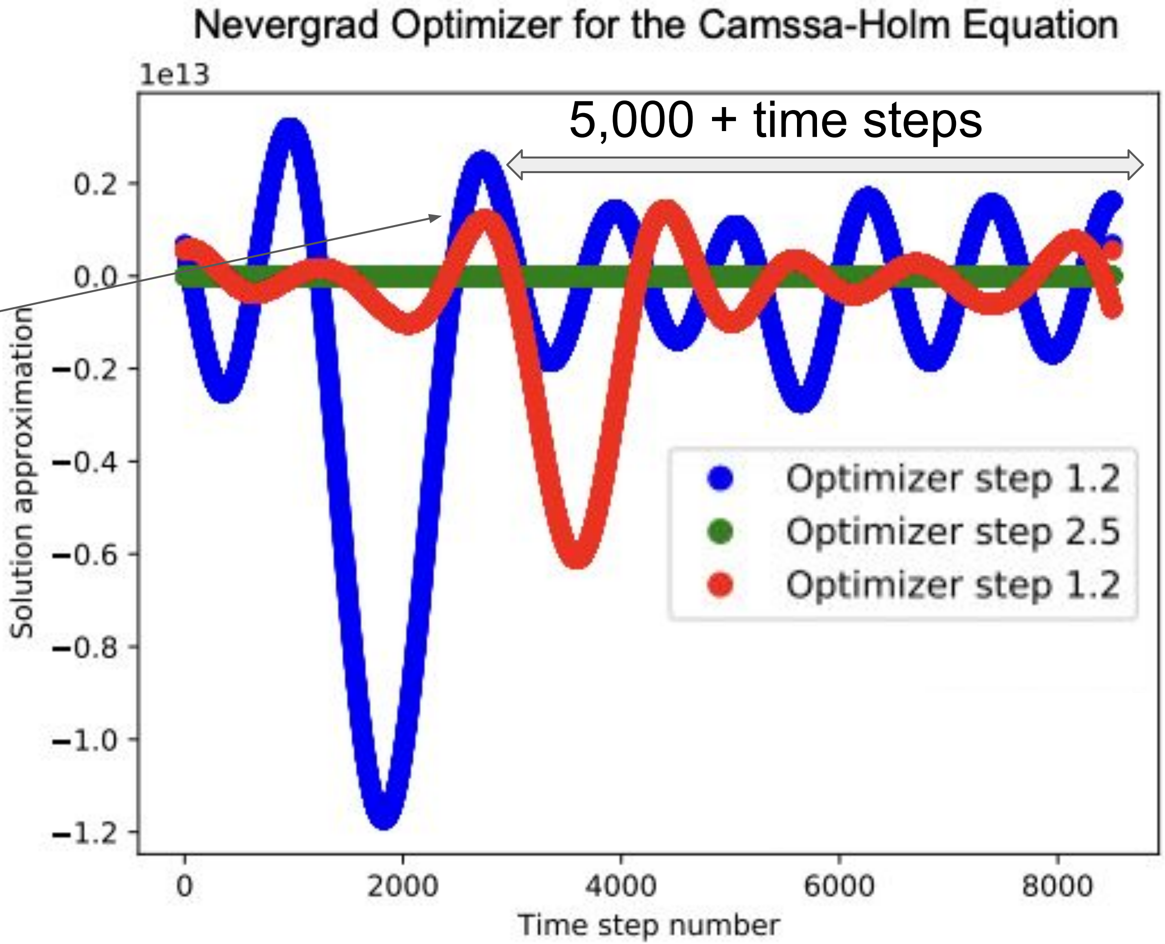
Absolute minima of two solutions can occur at completely different points of the evolution



Running the QSim simulator + Nevergrad Optimizer for more than 4,000 time steps of evolution

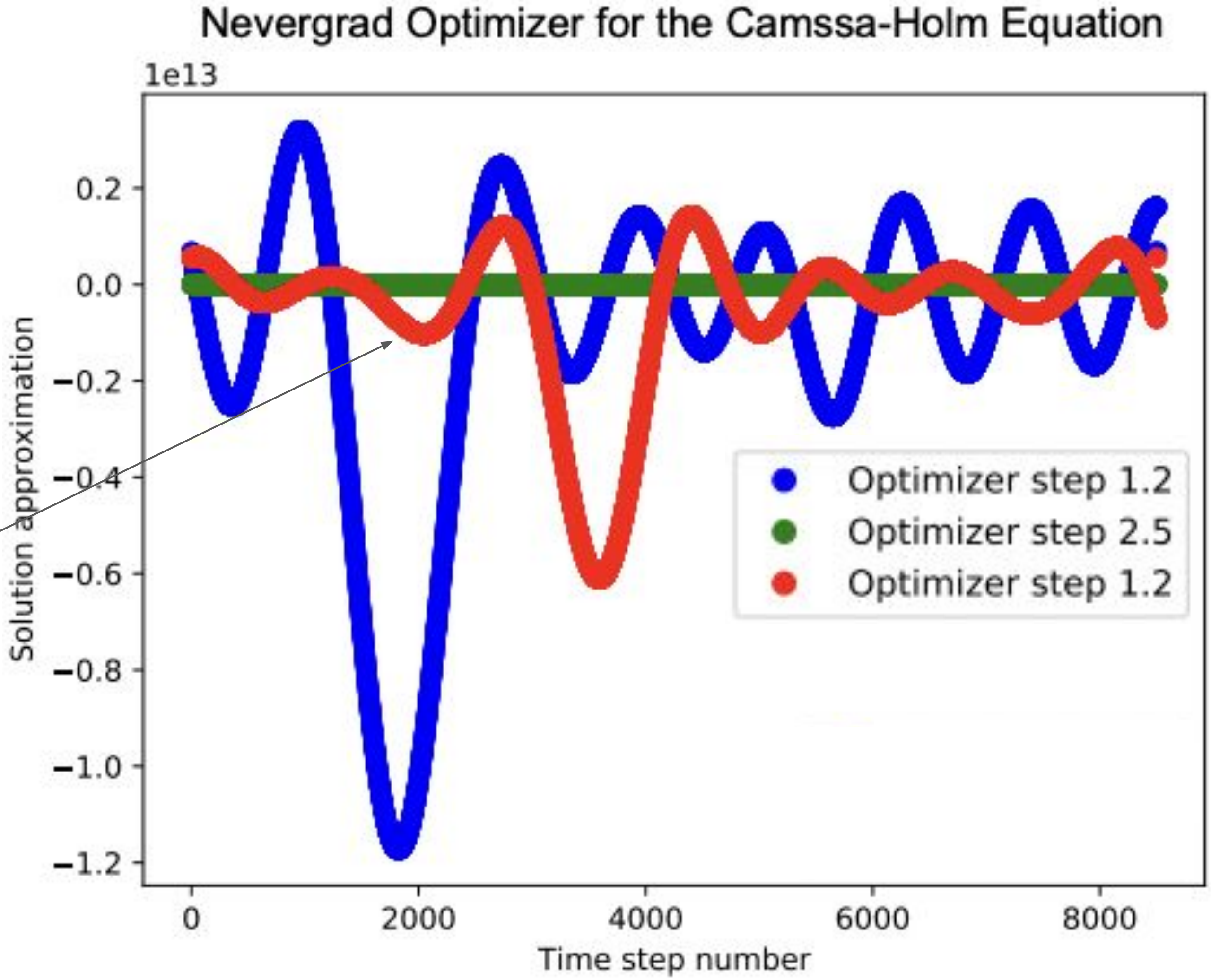
8,000 + time steps of evolution provide:

- Longer periods of motion in which the solution amplitude remains constant

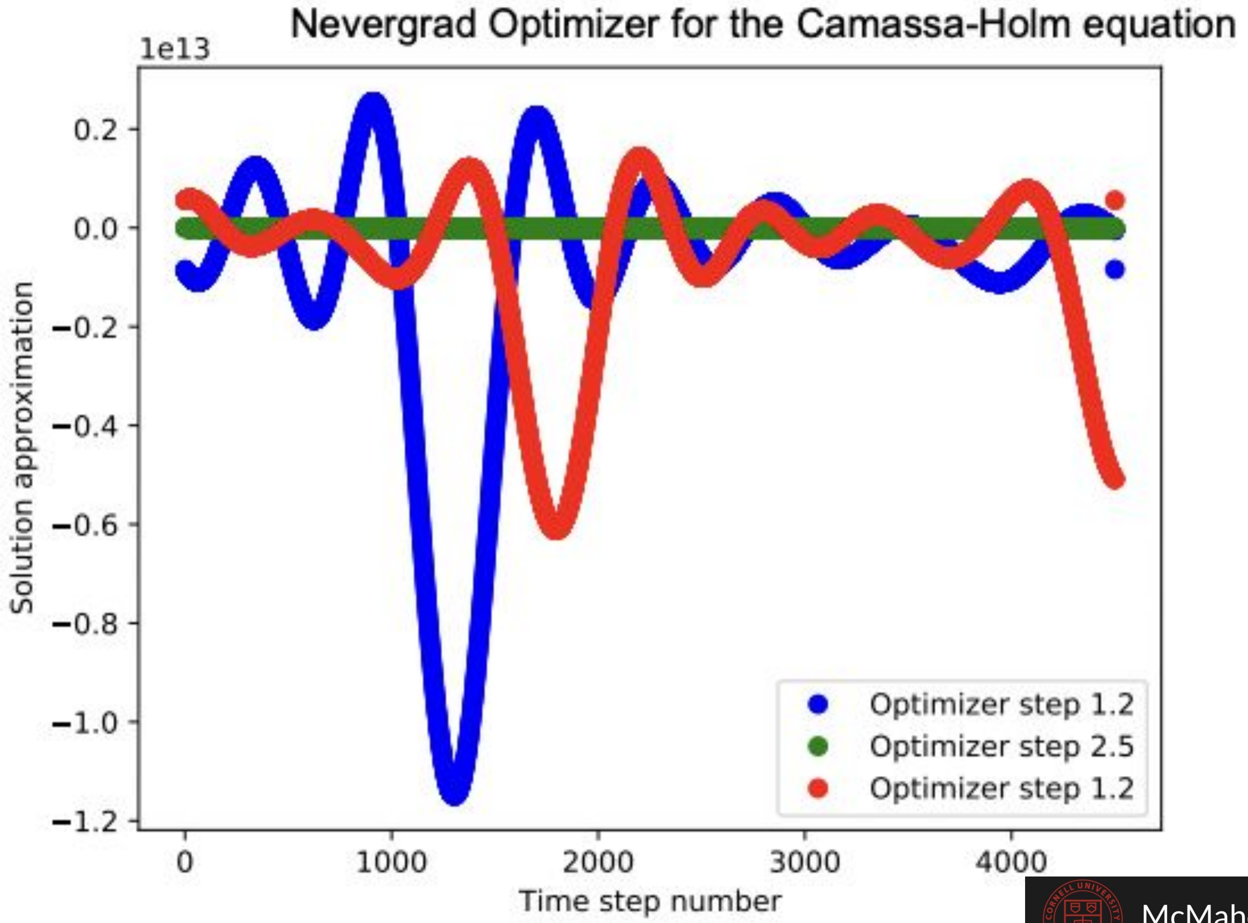


For other time step magnitudes with the Nevergrad Optimizer:

- Time evolution with time steps of size 1.2 exhibit faster periods of motion

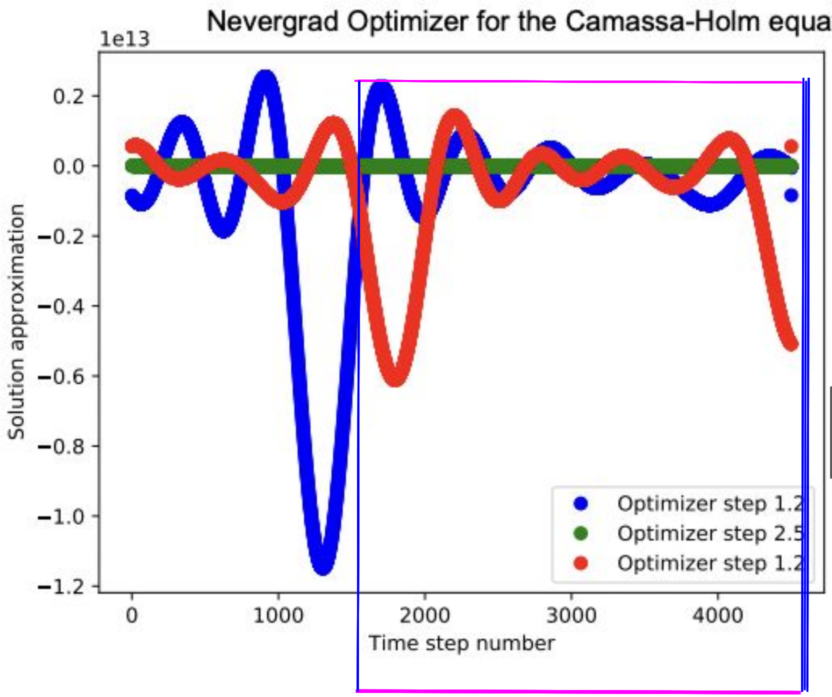


Albeit differences in the Optimizer step and IC, different time evolutions can overlap in Hilbert Space

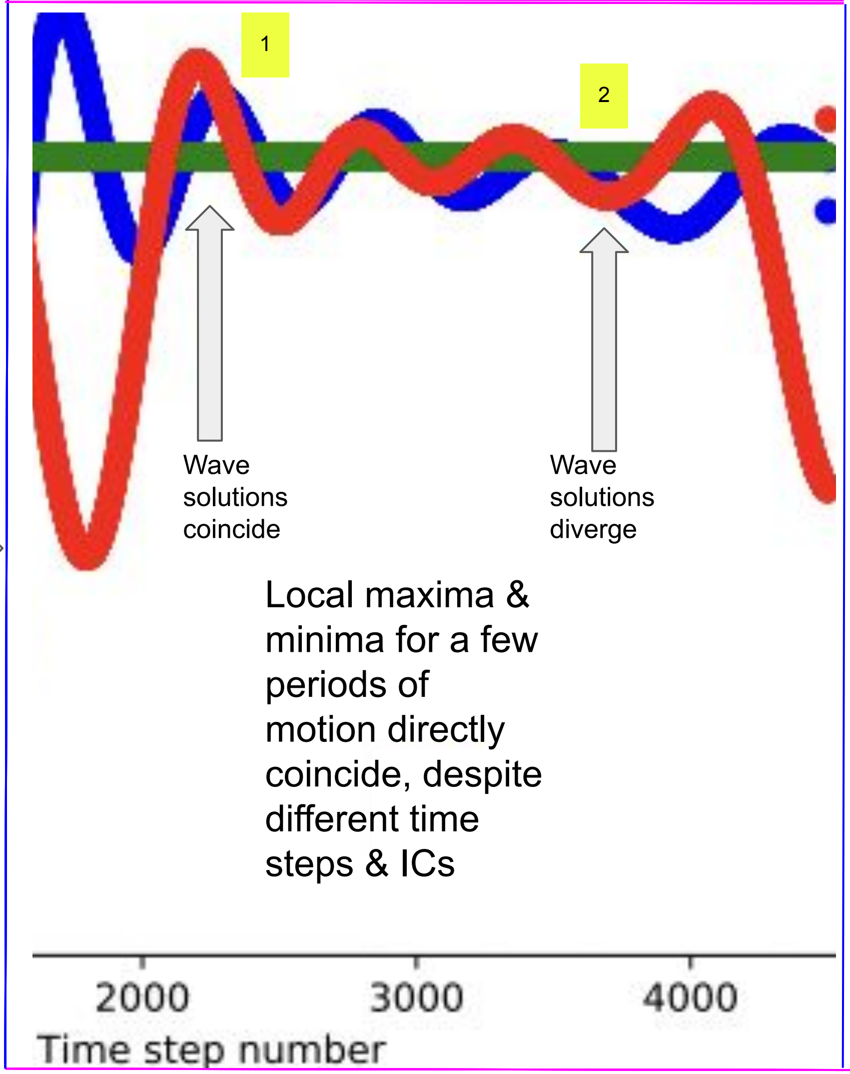


McMahon Lab

More specifically…



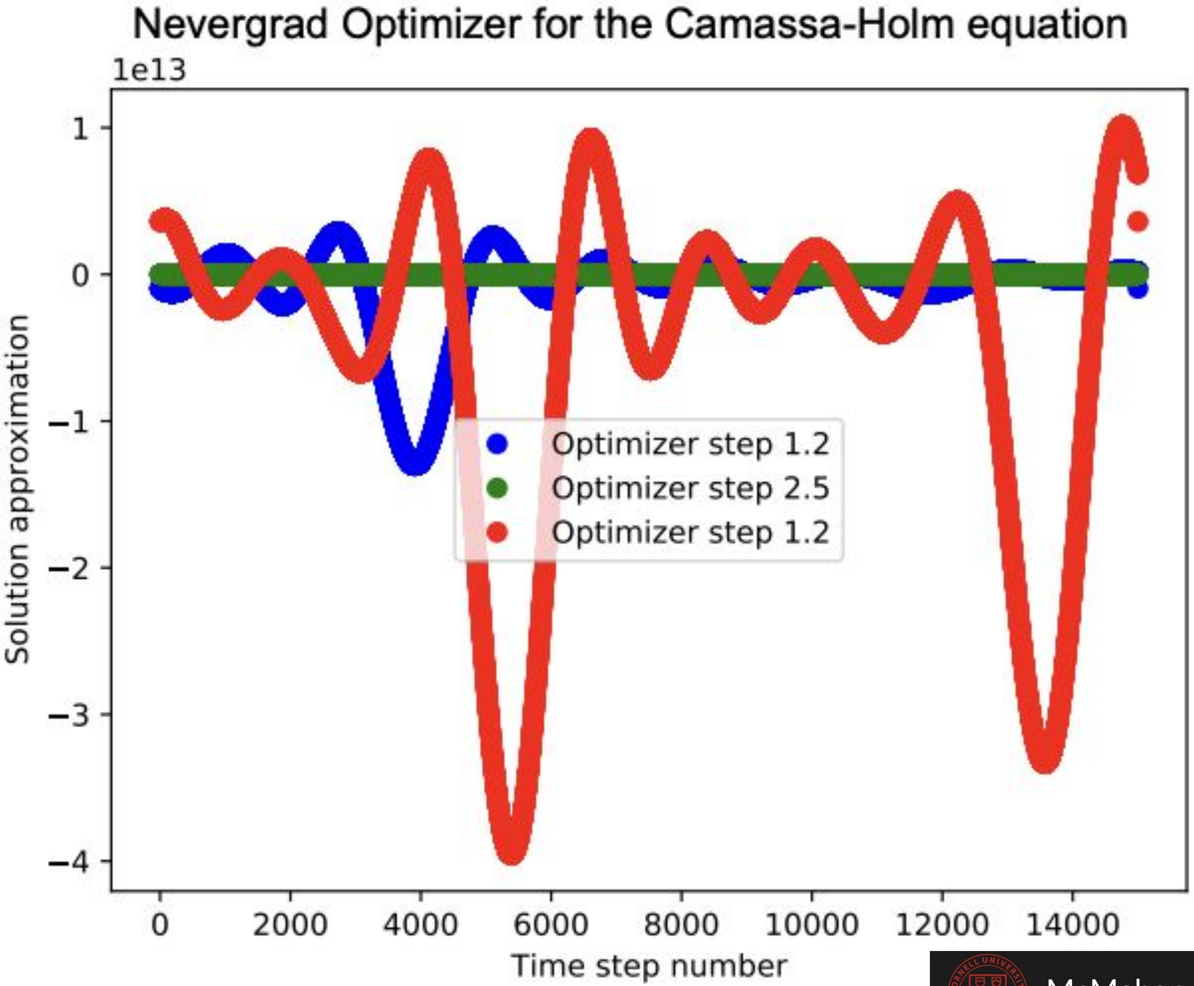
Restriction of time evolution within blue window

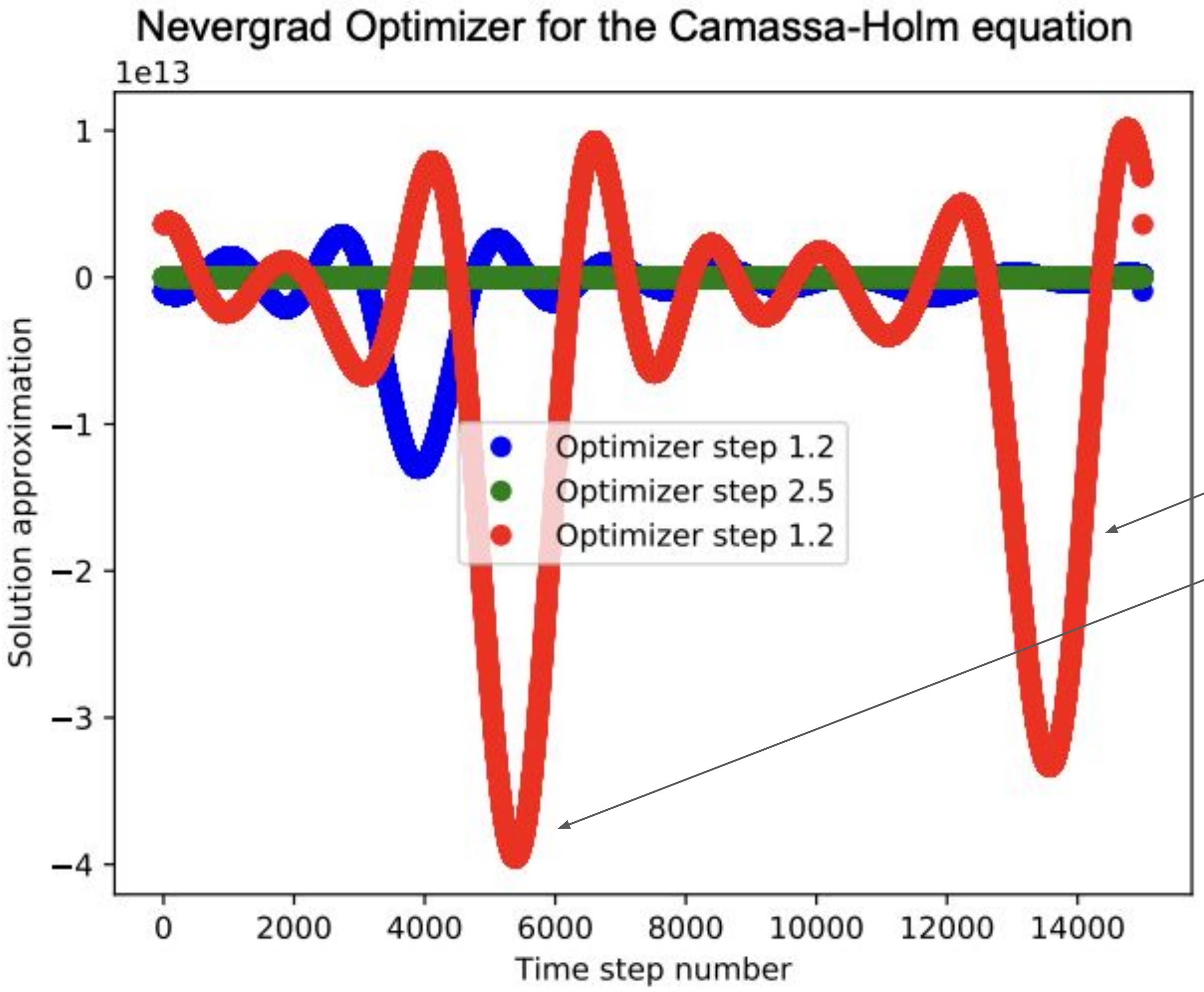


Local maxima & minima for a few periods of motion directly coincide, despite different time steps & ICs

A final example

15,000+ time steps of evolution





Recurring coverage of the Hilbert Space that are separated by ~9,000 time steps of evolution

Straddle between two fields

Pure Mathematics ⟷ Applied Mathematics

- Regularity conditions from Sobolev Space
- Well-posed Initial Value problem
- Existence of fixed points
- Duhamel formulation

- Algorithmic hardness
- Well-posed Optimization problem & constraints
- Classically simulable in polynomial time

Conclusions

- Provided an overview of the quantum algorithm
- Discussed the nonlinearity space to which the algorithm is applied
- Observed similarities between simulation results from Camassa-Holm PDE & different ranges of behaviors from a Proof in the 2020 reference below

Camassa–Holm Cuspons, Solitons and Their Interactions via the Dressing Method

Article | Full-text available

Feb 2020

Rossen I. Ivanov · Tony Lyons · Nigel Orr

Quantum Physics

[Submitted on 16 Sep 2022]

Variational quantum algorithm for measurement extraction from the Navier-Stokes, Einstein, Maxwell, Boussniesq-type, Lin-Tsien, Camassa-Holm, Drinfeld-Sokolov-Wilson, and Hunter-Saxton equations

Pete Rigas

Classical-quantum hybrid algorithms have recently garnered significant attention, which are characterized by combining quantum and classical computing protocols to obtain readout from quantum circuits of interest. Recent progress due to Lubasch et al in a 2019 paper provides readout for solutions to the Schrodinger and Inviscid Burgers equations, by making use of a new variational quantum algorithm (VQA) which determines the ground state of a cost function expressed with a superposition of expectation values and variational parameters. In the following, we analyze additional computational prospects in which the VQA can reliably produce solutions to other PDEs that are comparable to solutions that have been previously realized classically, which are characterized with noiseless quantum simulations. To determine the range of nonlinearities that the algorithm can process for other IVPs, we study several PDEs, first beginning with the Navier-Stokes equations and progressing to other equations underlying physical phenomena ranging from electromagnetism, gravitation, and wave propagation, from simulations of the Einstein, Boussniesq-type, Lin-Tsien, Camassa-Holm, Drinfeld-Sokolov-Wilson (DSW), and Hunter-Saxton equations. To formulate optimization routines that the VQA undergoes for numerical approximations of solutions that are obtained as readout from quantum circuits, cost functions corresponding to each PDE are provided in the supplementary section after which simulations results from hundreds of ZGR-QFT ansatzae are generated.

Comments: 144 pages, 100 figures

Subjects: **Quantum Physics (quant-ph)**; Information Theory (cs.IT); Numerical Analysis (math.NA); Optimization and Control (math.OC); Computational Physics (physics.comp-ph)

Cite as: arXiv:2209.07714 **[quant-ph]**
(or arXiv:2209.07714v1 **[quant-ph]** for this version)
https://doi.org/10.48550/arXiv.2209.07714

Submission history

From: Pete Rigas [view email]
[v1] Fri, 16 Sep 2022 04:44:19 UTC (38,329 KB)

Acknowledgments

Peter McMahon

Abhi Sarma

Yilian Liu

McMahon Lab

Are there any questions?