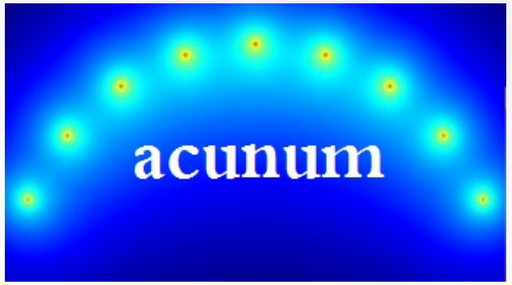
An Accelerated Weeks Method for Numerical Laplace Transform Inversion

Application
to
Viscoelastic Beam Modeling

Patrick O. Kano, Ph.D.

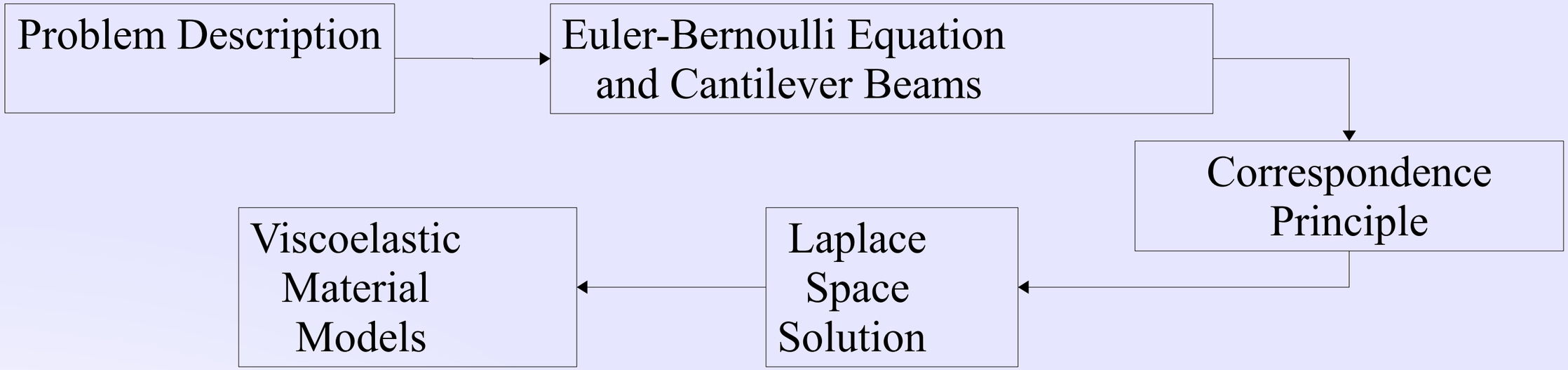
May 16, 2012



Acunum Algorithms and Simulations, LLC
Acute Numerical Algorithms And Efficient Simulations

Presentation Outline

I. Viscoelastic Beams and Modeling Concept



II. The Weeks Method for Numerical Laplace Transform Inversion

III. GPU Acceleration of the Weeks Method

IV. Application to Beam Modeling

V. Potential Impact and Future Research Directions

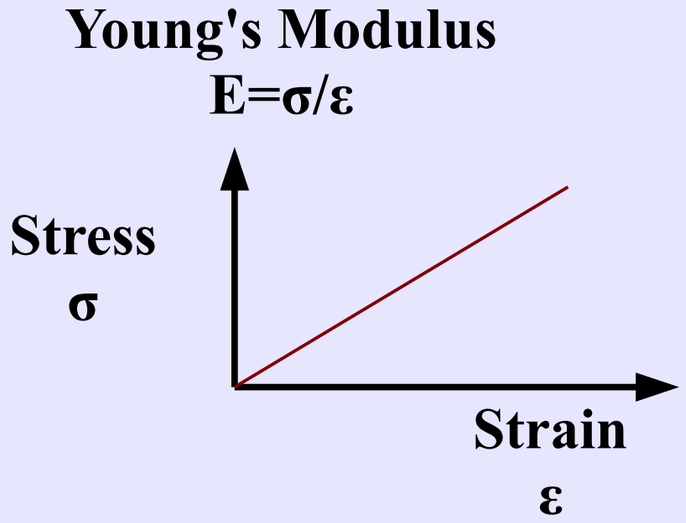
Viscoelastic Beams and Modeling Concept

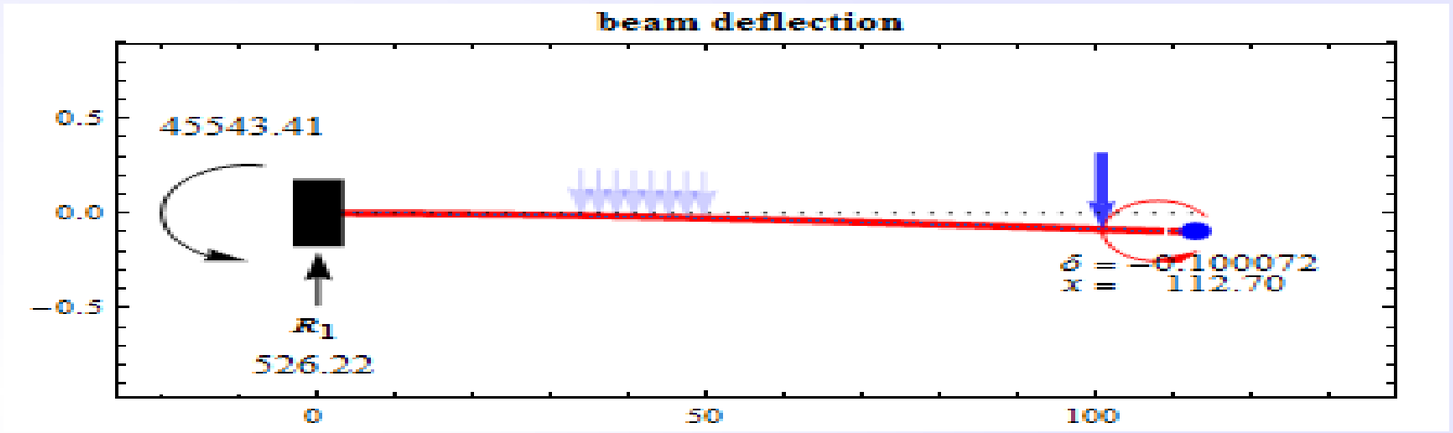
Beam deflection analysis is a fundamental problem in mechanical and structural engineering.

Linear elastic beams are well understood.

A simple linear elastic beam immediately responds to an applied stress.

Numerous software packages exist to model these beams.



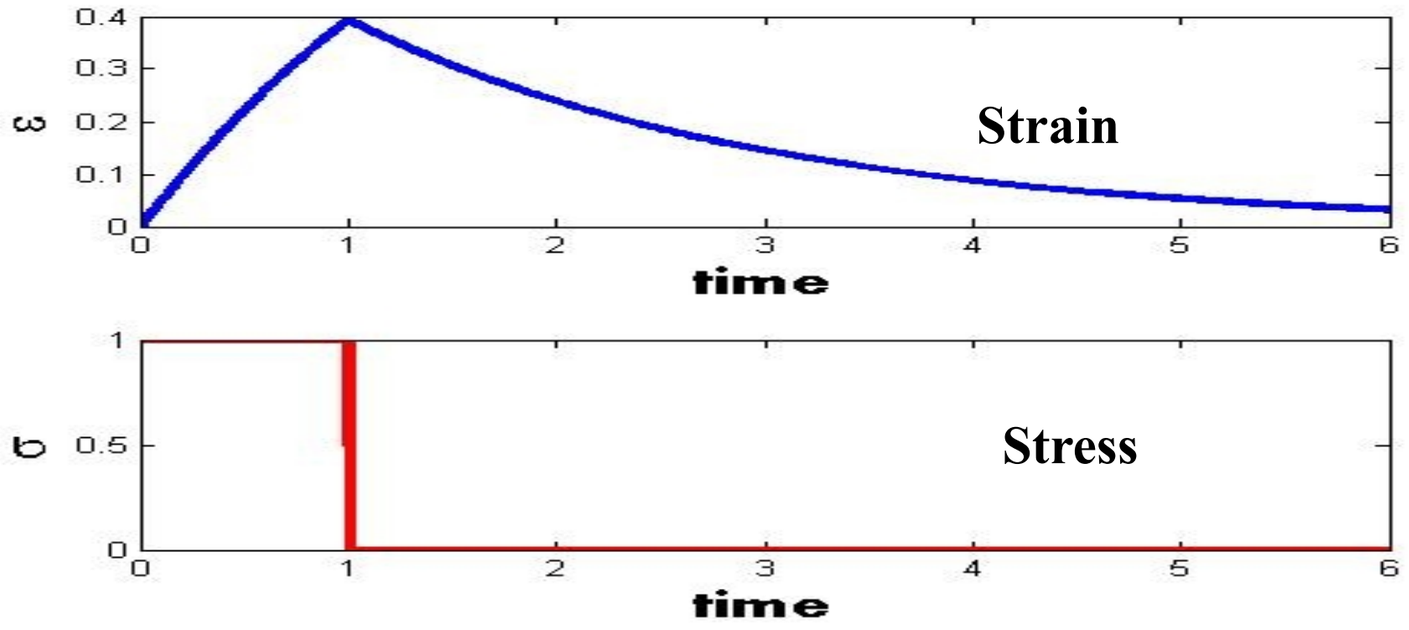


Wolfram Demonstrations Project
Single-Span Beam

Viscoelastic Beams and Modeling Concept

Viscoelastic materials possess both elastic and viscous properties.

These materials exhibit a time dependent response to a stress.



The deflection of a viscoelastic beam is much more difficult to predict.

Goal
Develop software to predict the deflection of beams with arbitrary viscoelastic material properties.

Euler-Bernoulli Equation

The Euler-Bernoulli equation provides a first order description of the deflection *w(x,t)* of an elastic beam.

$$\frac{\partial^4 w}{\partial x^4} + \frac{1}{a^2}\frac{\partial^2 w}{\partial t^2} = 0$$

$$a^2 = \frac{EI}{\rho A}$$

E = Young's modulus

I = area moment of inertia

A = beam cross-sectional area

ρ = beam density

Boundary Conditions for a Cantilever Beam

Fixed at $x = 0$

$$w(0,t) = 0$$

$$\frac{\partial w}{\partial x}(0,t) = 0$$

Free at $x = L$

$$\frac{\partial^2 w}{\partial x^2}(L,t) = 0$$

$$-EI\frac{\partial^3 w}{\partial x^3}(L,t) = F(t) - m\frac{\partial^2 w}{\partial t^2}(L,t)$$

Initial Conditions

$$w(x,0) = 0$$

$$\frac{\partial w}{\partial t}(x,0) = 0$$

Fixed x=0

Cantilever Beam

Free and Forced x=L

Correspondence Principle

The displacement *w(t)* of a viscoelastic beam is obtained from:

- solving a corresponding *elastic* problem in Laplace transform space for *W(x,s)*
- replacing the Young's modulus E with complex function E(s)
- inverting back to the time domain *w(x,t)*

The principle uses the Laplace transform.

$$F(s) = \int_0^\infty e^{-st} f(t)\,dt$$

$$f(t) = \frac{1}{2\pi i} \oint_\Gamma e^{st} F(s)\,ds$$

$$F(s) : \mathbb{C} \longrightarrow \mathbb{C}$$

$$f(t) : \mathbb{R} \longrightarrow \mathbb{R}$$

Convert the problem to Laplace Transform Space

↓

Solve the Euler-Bernoulli Equation for W(s;E)

↓

Replace elastic modulus E with a viscoelastic function E(s)

↓

Inverse Laplace Transform W(s;E(s)) → w(t)

The last step is the most difficult.

Laplace Space Solution

Applying the Laplace transform yields ordinary differential equations in space:

$$\frac{d^4W}{dx^4} + \frac{1}{a^2}\left[s^2W(x,s) - \frac{dw}{dt}(x,0) - s\cdot w(x,0)\right] = 0$$

$$W(0,s) = 0$$

$$\frac{dW}{dx}(0,s) = 0$$

$$\frac{d^2W(L,s)}{dx^2} = 0$$

$$-EI\frac{d^3W(L,s)}{dx^3} = F(s) - m\left(s^2W(L,s) - \frac{dw}{dt}(L,0) - s\cdot w(L,0)\right)$$

Convert the problem to Laplace Transform Space

The general solution has the form:

$$W(x,s) = k_1e^{r_1\sqrt{\frac{s}{a}}x} + k_2e^{r_2\sqrt{\frac{s}{a}}x} + k_3e^{r_3\sqrt{\frac{s}{a}}x} + k_4e^{r_4\sqrt{\frac{s}{a}}x}$$

$$r_j = e^{i\left(\frac{\pi}{4}+(j-1)\frac{\pi}{2}\right)}$$

$$j = 1,2,3,4$$

Laplace Space Solution

The four boundary conditions can be applied to determine the four unknown coefficients $\{k\}$.

$$W(x,s) = F(s)\frac{b_1 e^{r_1\sqrt{\frac{s}{a}}x} + b_2 e^{r_2\sqrt{\frac{s}{a}}x} + b_3 e^{r_3\sqrt{\frac{s}{a}}x} + b_4 e^{r_4\sqrt{\frac{s}{a}}x}}{b_1 h_1 + b_2 h_2 + b_3 h_3 + b_4 h_4}$$

$$r_j = e^{i\left(\frac{\pi}{4}+(j-1)\frac{\pi}{2}\right)}$$

$$h_j = \left(ms^2 - EI\left(\frac{s}{a}\right)^{3/2} r_j^3\right) e^{r_j\sqrt{\frac{s}{a}}L}$$

$$g_j = r_j^2 e^{r_j\sqrt{\frac{s}{a}}L}$$

$$b_1 = r_4 g_3 + r_2 g_4 + r_3 g_2 - r_3 g_4 - r_4 g_2 - r_2 g_3$$

$$b_2 = r_3 g_4 + r_4 g_1 + r_1 g_3 - r_4 g_3 - r_1 g_4 - r_3 g_1$$

$$b_3 = r_4 g_2 + r_1 g_4 + r_2 g_1 - r_2 g_4 - r_4 g_1 - r_1 g_2$$

$$b_4 = r_2 g_3 + r_3 g_1 + r_1 g_2 - r_3 g_2 - r_1 g_3 - r_2 g_1$$

Solve the Euler-Bernoulli Equation for W(s;E)

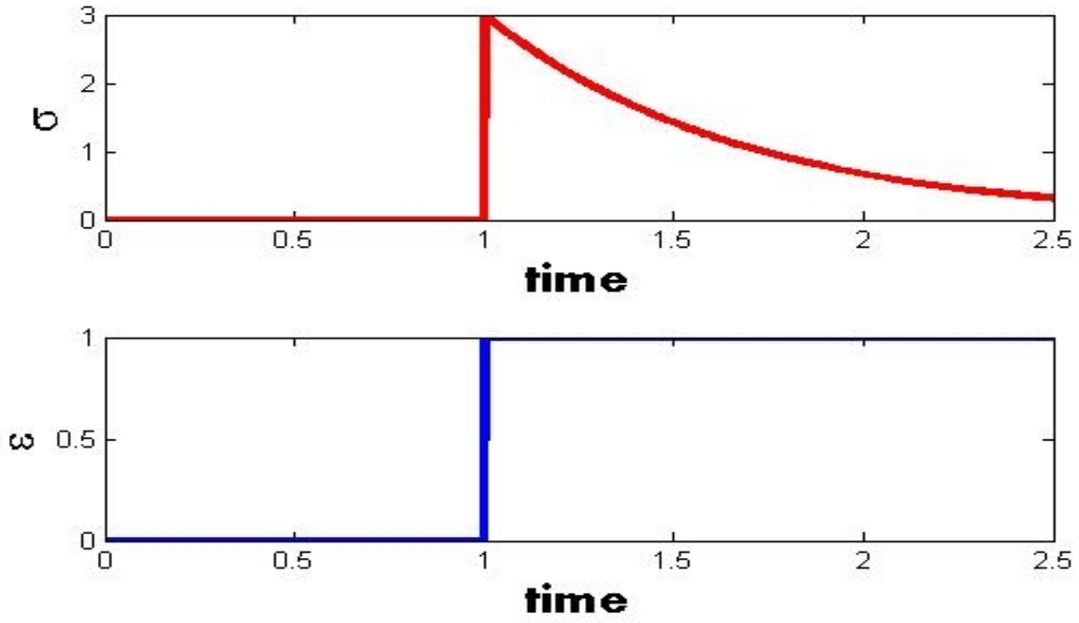
This expression for W(x,s) is difficult to invert analytically.

Viscoelastic Material Models

Correspondence principle →Replace constant Young's modulus E with viscoelastic E(s)

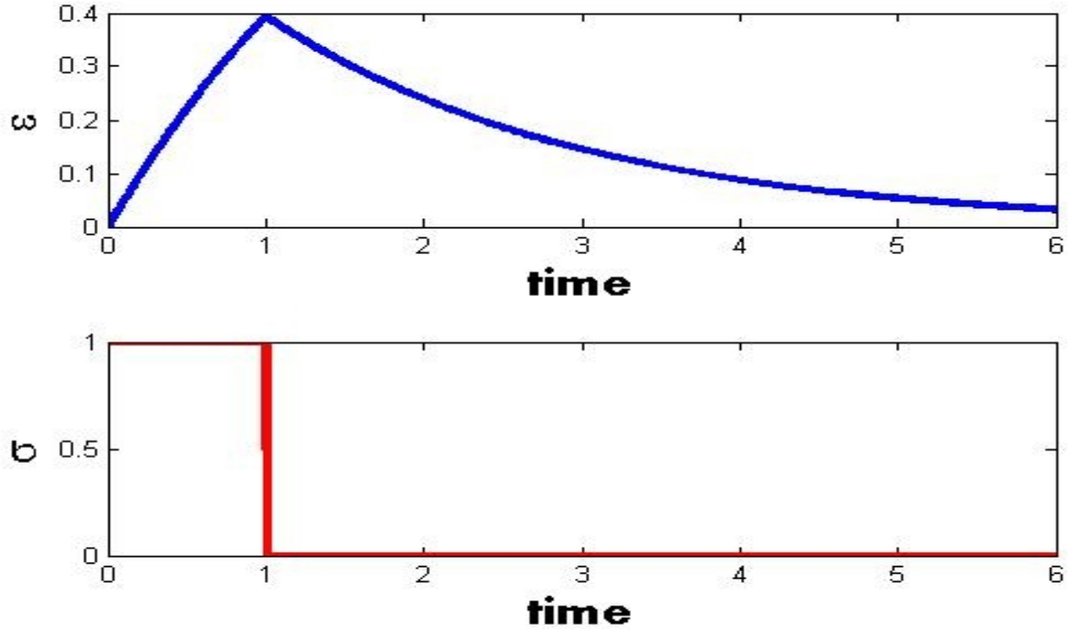
Real world models for E(s) are complicated.
Mechanical models can capture fundamental viscoelastic phenomena.

Stress-Relaxation



stress release under constant strain

Creep



nonlinear strain under applied stress

Visceolastic Material Models

spring $\sigma = E\epsilon$

$\sigma = \eta \frac{d\epsilon}{dt}$ damper

Series Combination

Parallel Combination

Maxwell Model

predicts **stress relaxation**

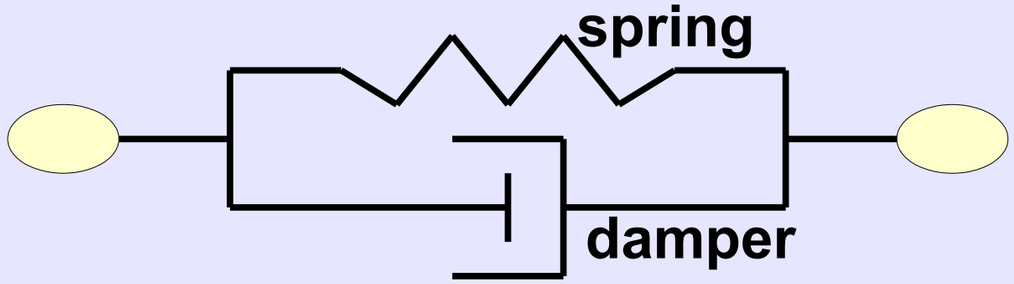
$$\frac{d\epsilon}{dt} = \frac{\sigma}{\eta} + \frac{1}{E}\frac{d\sigma}{dt}$$

$$s\hat{\epsilon} = \left(\frac{1}{\eta} + \frac{s}{E}\right)\hat{\sigma}$$

$$E_{eff}(s) = \frac{s}{\frac{1}{\eta} + \frac{s}{E}}$$

Replace elastic modulus E with a viscoelastic function E(s)

Kelvin-Voight Model



predicts **creep behavior**

$$\sigma = E\epsilon + \eta\frac{d\epsilon}{dt}$$

$$\hat{\sigma}(s) = (E + s\eta)\hat{\epsilon}$$

$$E_{eff}(s) = E + s\eta$$

E(s) substituted for E into *W(x,s)*
yields an expression that no one can analytically invert.
A numerical Laplace transform inversion method is required.

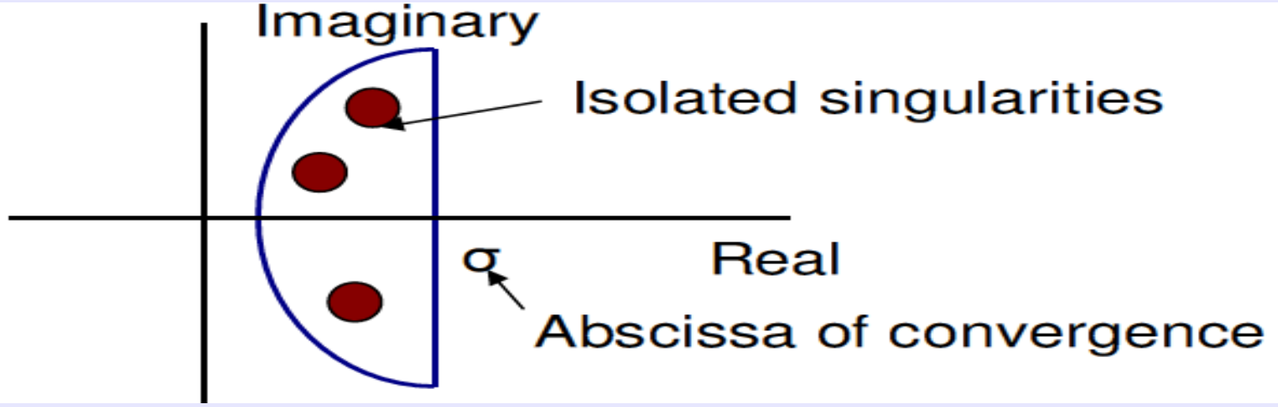
Inverse Laplace Transform

Inverse Laplace Transform
W(s;E(s)) → w(t)

Analytic inversion of the Laplace transform is defined via contour integration in the complex plane.

The *Bromwich* contour is commonly chosen.

$$f(t) = \frac{1}{2\pi i}\int_{\sigma-i\infty}^{\sigma+i\infty} F(s)e^{st}ds$$



For simple F(s), Cauchy's residue theorem can be employed.

$$\int_{\sigma-i\infty}^{\sigma+i\infty} F(s)e^{st}ds = 2\pi i\sum_j r_j$$

$$r_j = \frac{1}{(m-1)!}\lim_{s\to s_j}\frac{d^{m-1}}{ds^{m-1}}\left[(s-s_j)^m F(s)e^{st}\right]$$

→ f(t) is the sum of the residues

For complicated F(s), this approach can be too cumbersome to perform even in symbolic software (Maple or Mathematica).

Numerical Inversion Methods

Numerical Laplace transform inversion
is an inherently sensitive problem.

The exponential term leads rapid growth of numerical errors.

$$f(t) = \frac{1}{2\pi i} \int_{\sigma-i\infty}^{\sigma+i\infty} F(s) e^{st} ds$$

Post's Formula (1930)

- Based on asymptotic expansion (Laplace's method)
- Post (1930), Gaver (1966), Valko-Abate (2004)

Weeks Method (1966)

- Laguerre polynomial expansion method
- Ward (1954), Weeks (1966), Weideman (1999)

Talbot's Method (1979)

- Deformed contour method
- Talbot (1979), Weideman & Trefethen (2007)

Numerical inverse Laplace transform algorithm development is a long standing problem.

Weeks' Method

The Weeks method is one of the most well known algorithms for the numerical inversion of a Laplace space function.

It returns an explicit expression for the time domain function as an expansion in Laguerre polynomials.

$$f(t) = \lim_{N\to\infty} f_N(t)$$

$$f_N(t) = e^{\sigma t} \sum_{n=0}^{N-1} a_n e^{-bt} L_n(2bt)$$

$$L_n(x) = \frac{e^x}{n!}\frac{d^n}{dx^n}\left(e^{-x}x^n\right)$$

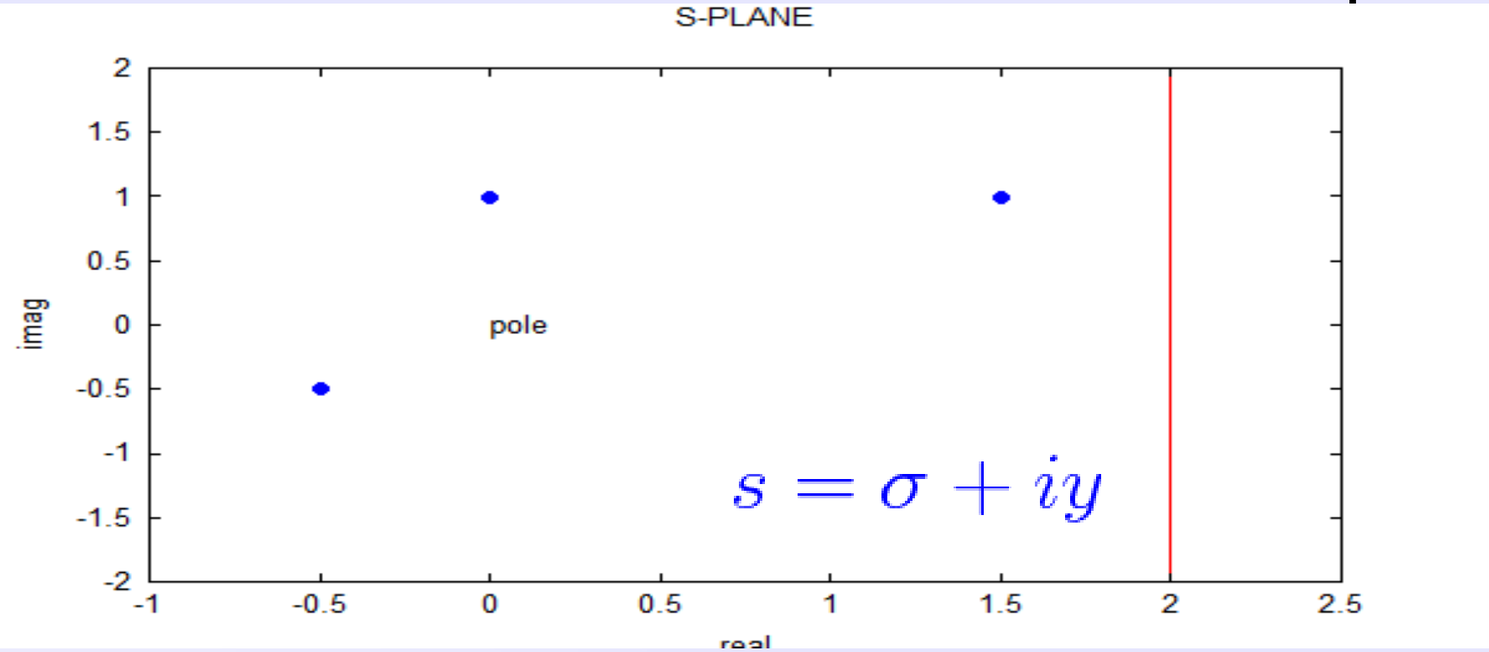
The coefficients $\{a_n\}$

1. contain the information particular to the Laplace space function
2. may be complex scalars, vectors, or matrices
3. time independent

Two free scaling parameters **σ and *b***,
must be selected according to the constraints that:

- *b>0 [Time scale factor]* ensures that the Laguerre polynomials are well behaved for large t
- $\sigma>\sigma_0$ *[Exponential factor]* greater than the abscissa of convergence

Coefficients Computation



$$f(t) = \frac{1}{2\pi i} \int_{\sigma - i\infty}^{\sigma + i\infty} F(s) e^{st} ds$$

The weighted Laguerre polynomials have a nice Fourier representation:

$$\sum_{n=0}^{\infty} a_n e^{-bt} L_n(2bt) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{iyt} F(\sigma + iy) dy$$

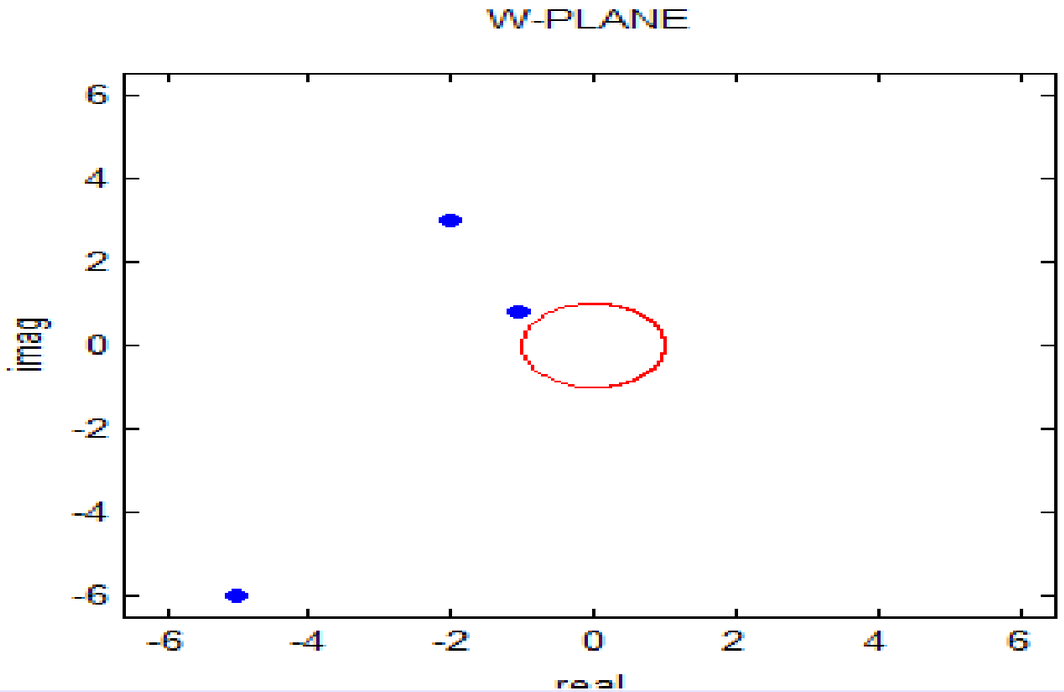
$$e^{-bt} L_n(2bt) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{iyt} \frac{(iy - b)^n}{(iy + b)^{n+1}} dy$$

$$\sum_{n=0}^{\infty} a_n \frac{(iy - b)^n}{(iy + b)^{n+1}} = F(\sigma + iy)$$

Möbius Transformation s→w

Instead of integration on the y-line of *s*, *integrate on the circular contour* in *w*.

$$w = \frac{s - \sigma - b}{s - \sigma + b}$$



With the change of variables, one obtains a **power series** in *w*.

$$\sum_{n=0}^{\infty} a_n w^n = \frac{2b}{1-w} F\left(\sigma - b\frac{w+1}{w-1}\right)$$

The unit circle parametrized by θ as an integration path. $w = e^{i\theta}$

The coefficients are dependent on (σ,b).

$$a_n = \frac{1}{2\pi}\int_{-\pi}^{\pi} e^{-in\theta}\frac{2b}{1-e^{i\theta}} F\left(\sigma - b\frac{e^{i\theta}+1}{e^{i\theta}-1}\right) d\theta$$

Weeks' Method Error Estimate

A straight forward error estimate yields three contributions:

1. Discretization (D_E) – Finite integral sampling
2. Truncations (T_E) – Finite number of Laguerre polynomials
3. Round-off (R_E) – Finite computation precision

The integration on the circular w-space contour converges quickly →
The discretization error can be neglected.

$$E_{total} \leq e^{\sigma t}(T_E + R_E)$$

$$E_{total} \leq e^{\sigma t}\left(\sqrt{\sum_{n=N}^{\infty} ||a_n|\,|^2} + \epsilon\sqrt{\sum_{n=0}^{N-1} ||a_n|\,|^2}\right)$$

$$||a_n|\,|^2 \leq \frac{||F(s)|\,|^2}{r^n}$$

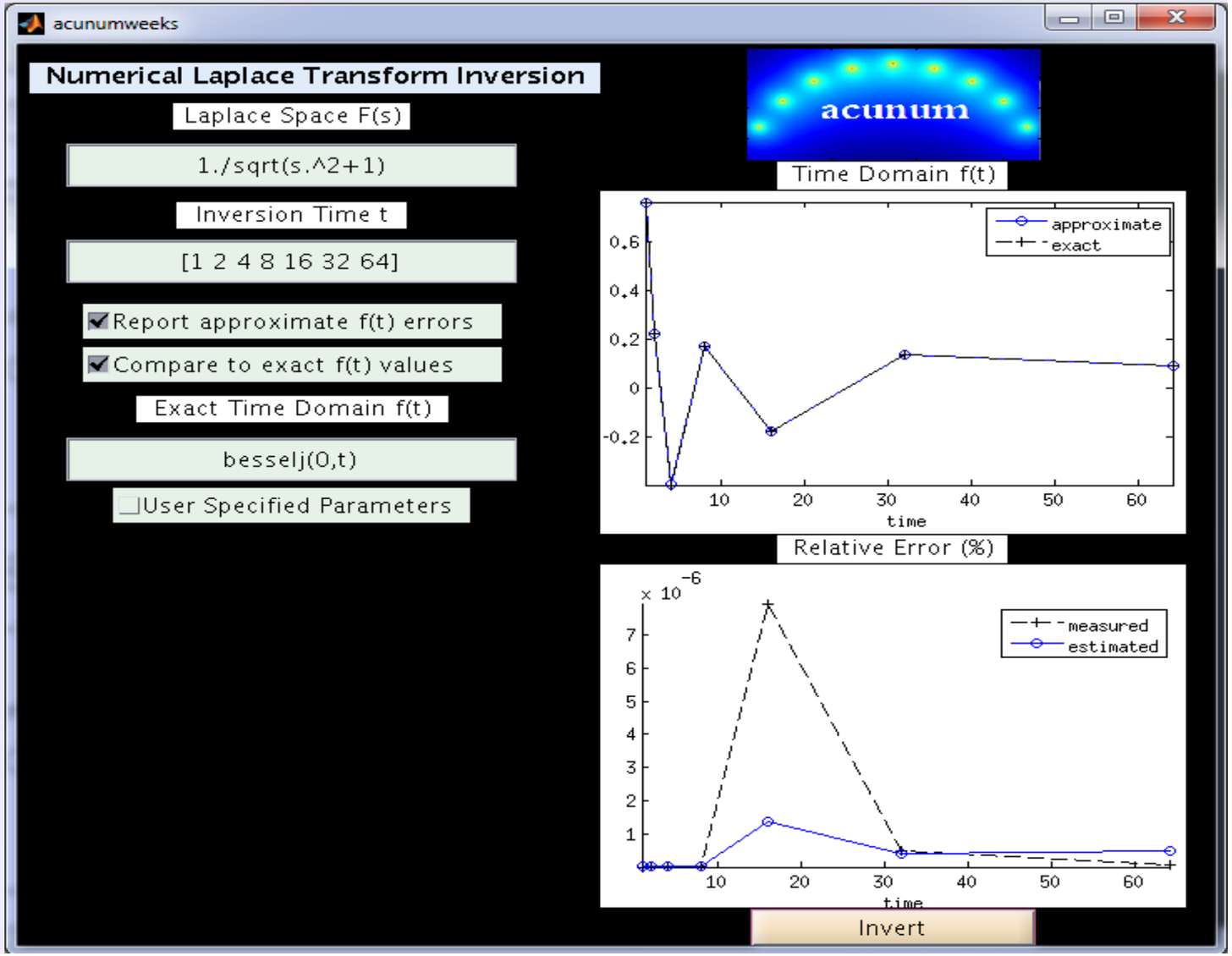
Inverse Laplace Transform
W(s;E(s)) → w(t)

Minimizing this error estimate yields optimal σ and b parameters.

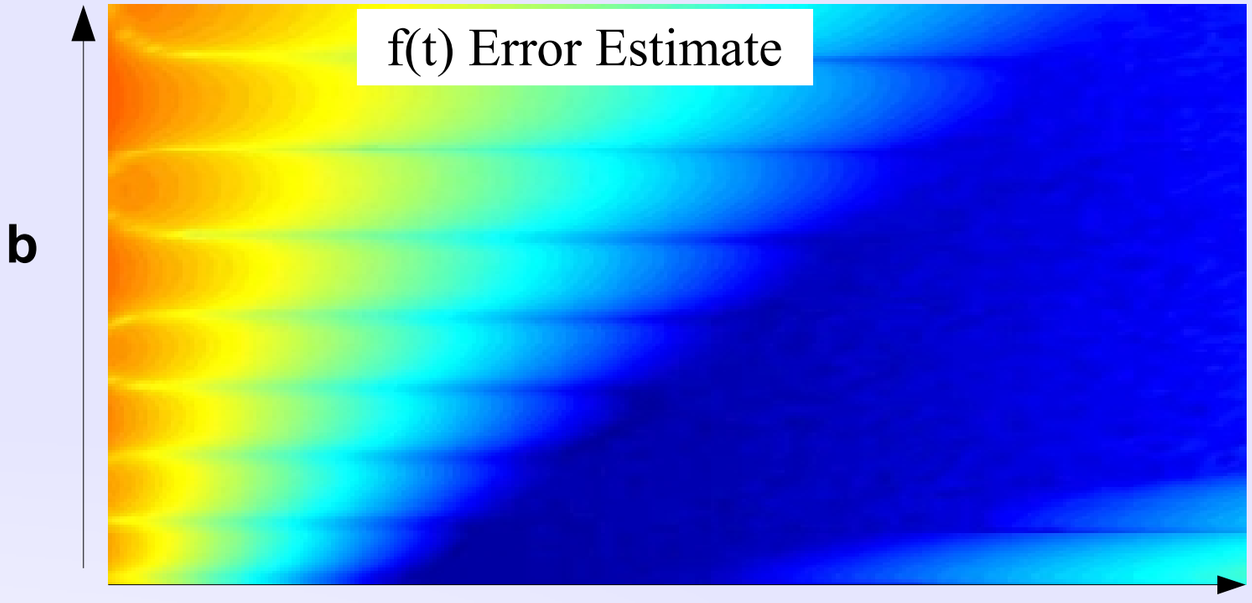
Weeks Method GPU Accelerated Tool

Available on the MATLAB file exchange

- The codes are freely available under a [BSD] license in:
  - MATLAB using JACKET from AccelerEyes, Inc.
  - C/C++ with CUDA on the Acunum website
- The MATLAB tool includes a GUI 'acunumweeks' or can be run from the MATLAB environment.



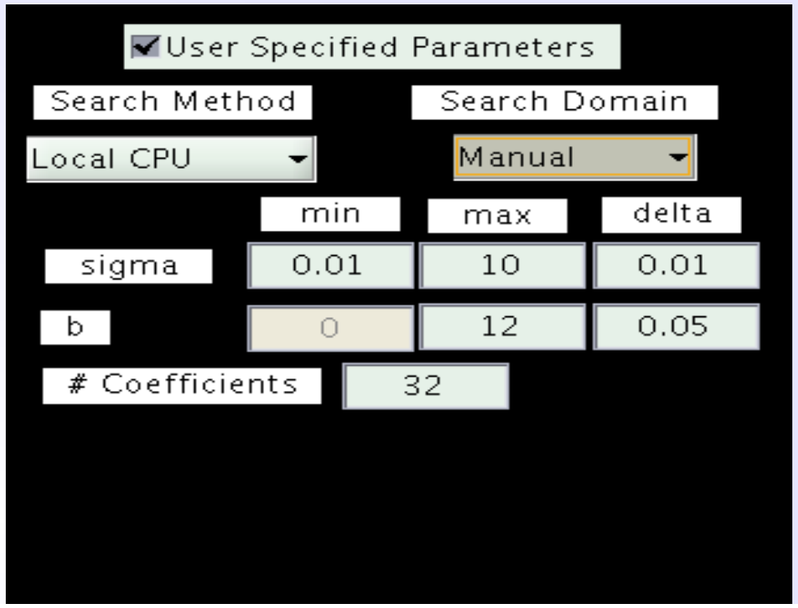
Weeks Method GPU Accelerated Tool

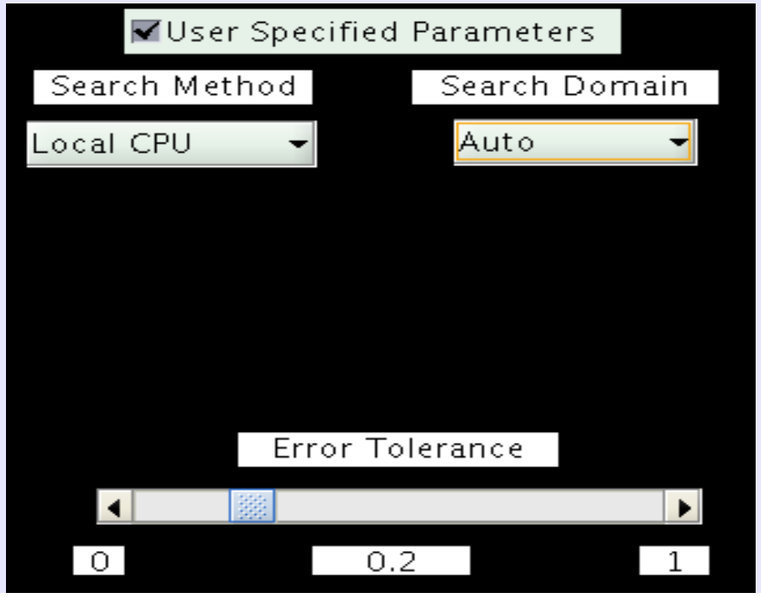


Minimize an error estimate of f(t;σ,b) to obtain optimal (σ,b) parameters.

Use graphics processing unit [GPU] parallelization to perform a global minimization of the error estimate.

Manual (σ,b) Ranges





Auto (σ,b) Ranges

Weeks Method GPU Accelerated Tool

Acunum has developed a C++ & MATLAB tool that performs the computations on NVIDIA graphics processors.

MATLAB

Dominant environment for scientific computing and algorithm development.

NVIDIA Graphics Processing Units

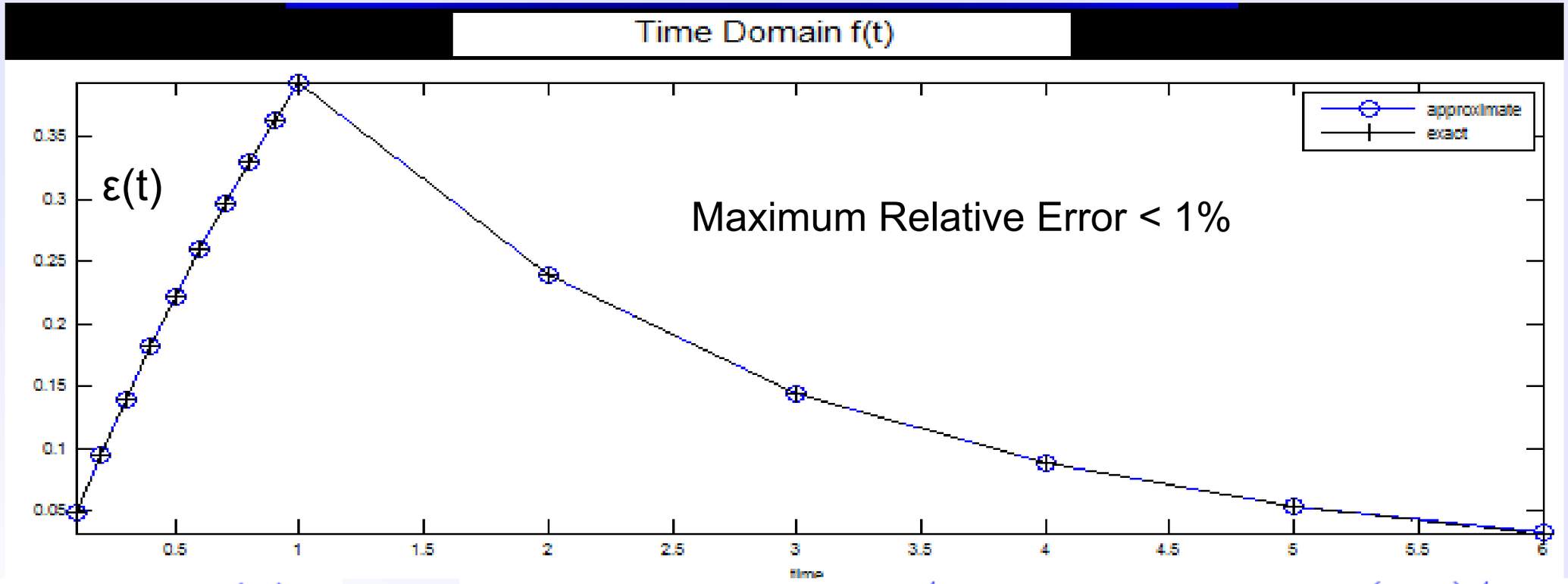
Low cost and widely distributed graphics processors allow for wide tool use.

ACUNUM Software

Acunum's MATLAB tools uses JACKET from AccelerEyes, Inc. for the MATLAB-to-GPU interface.

Kelvin-Voight Model Test

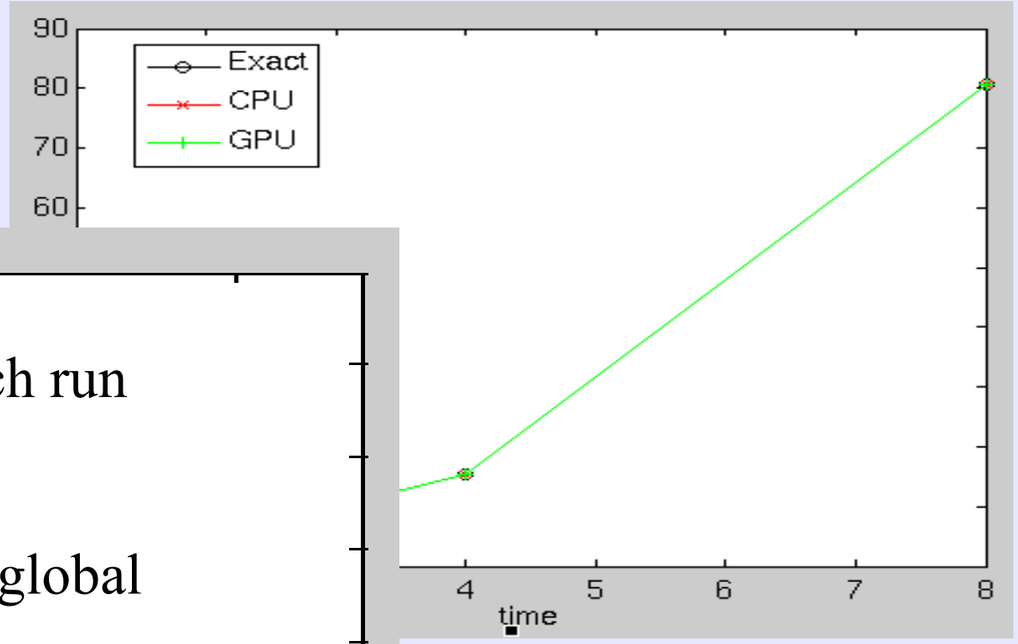
- The simple Maxwell & Kelvin-Voight models
  - are too simplistic to quantitatively describe viscoelastic materials.
  - provide test cases for code validation.



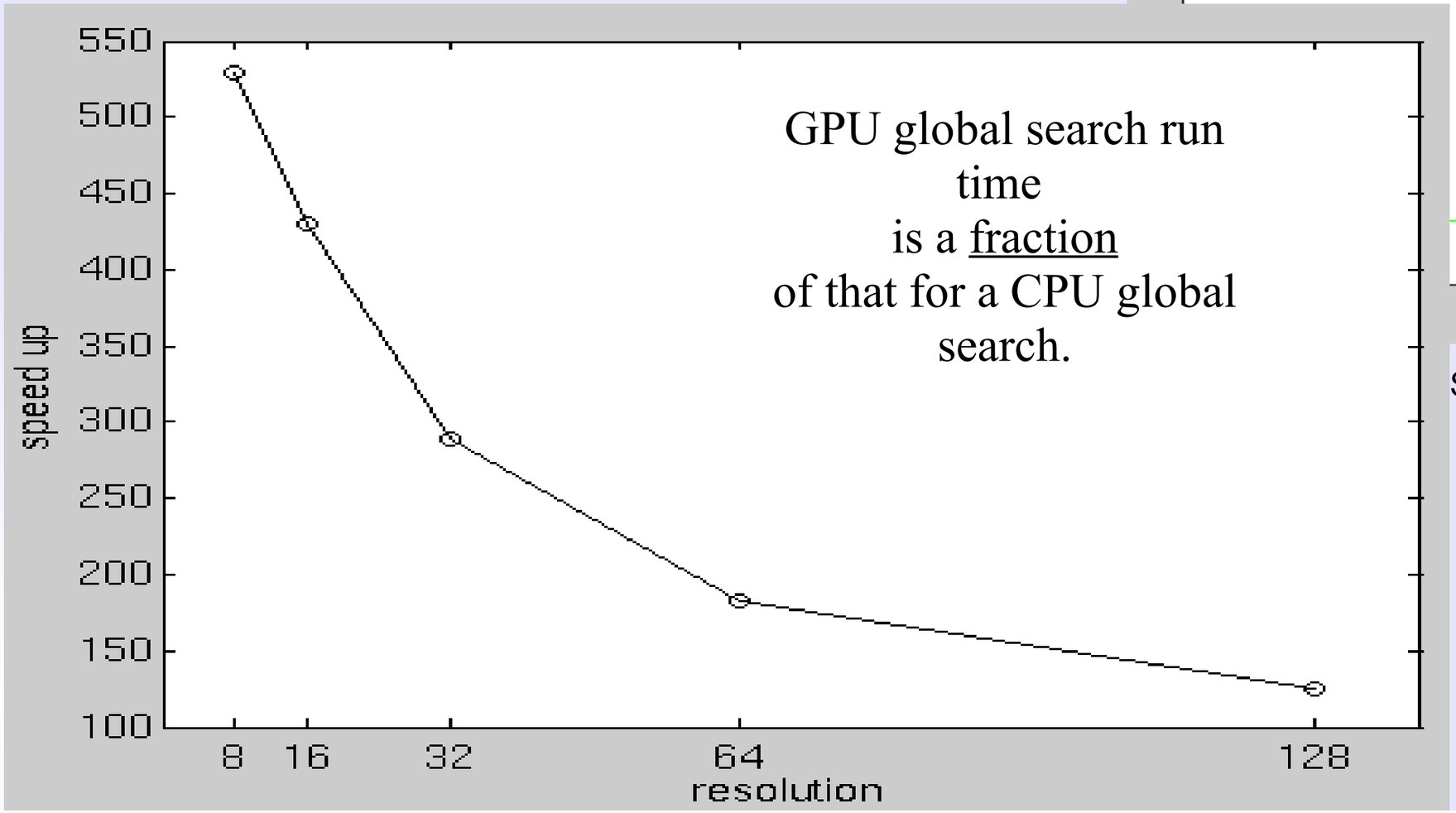
Creep

$$\epsilon(s) = \frac{\sigma(s)}{E + s\eta} \longrightarrow \begin{aligned} \epsilon(t) &= (1 - e^{-t/2}) + H(t-1)(e^{-(t-1)/2} - 1) \\ \sigma(t) &= H(t-1) \end{aligned}$$

Speed Test



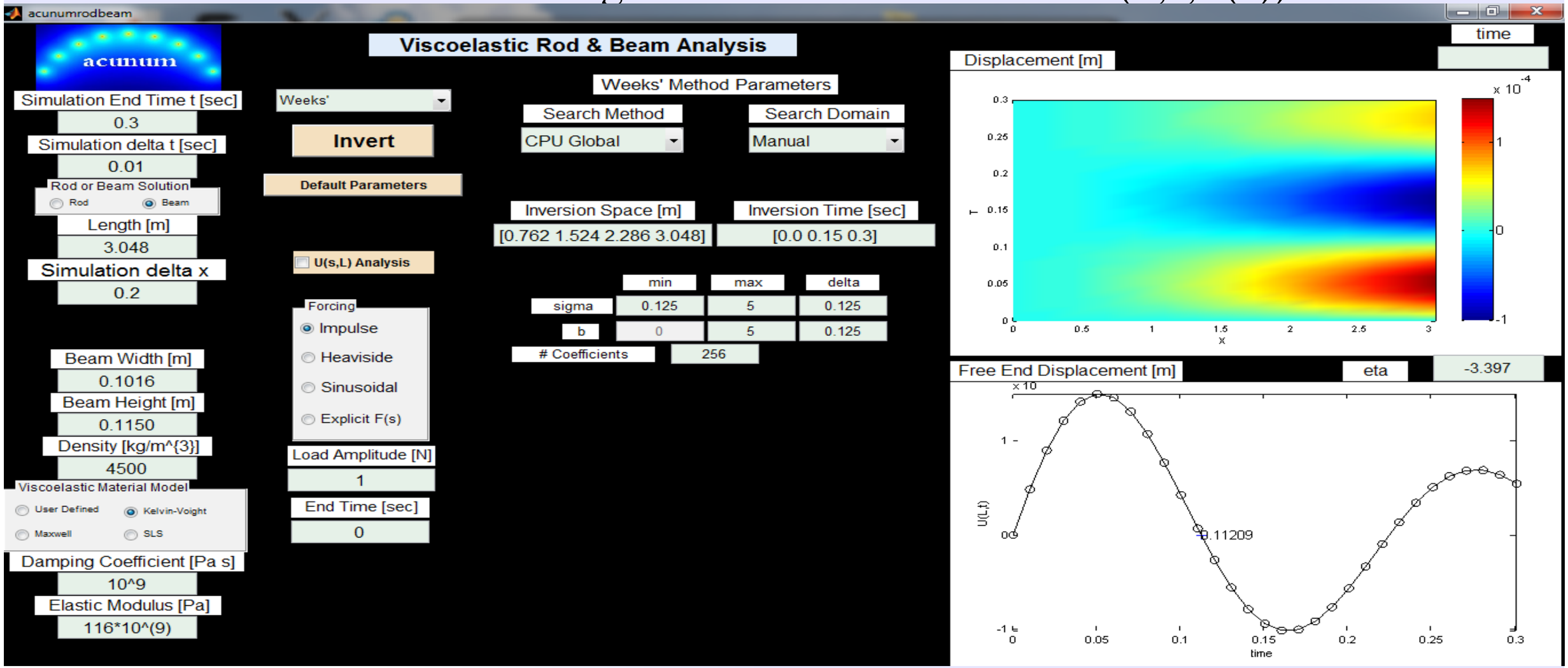
Solutions are the same.



GPU global search run time
is a fraction
of that for a CPU global search.

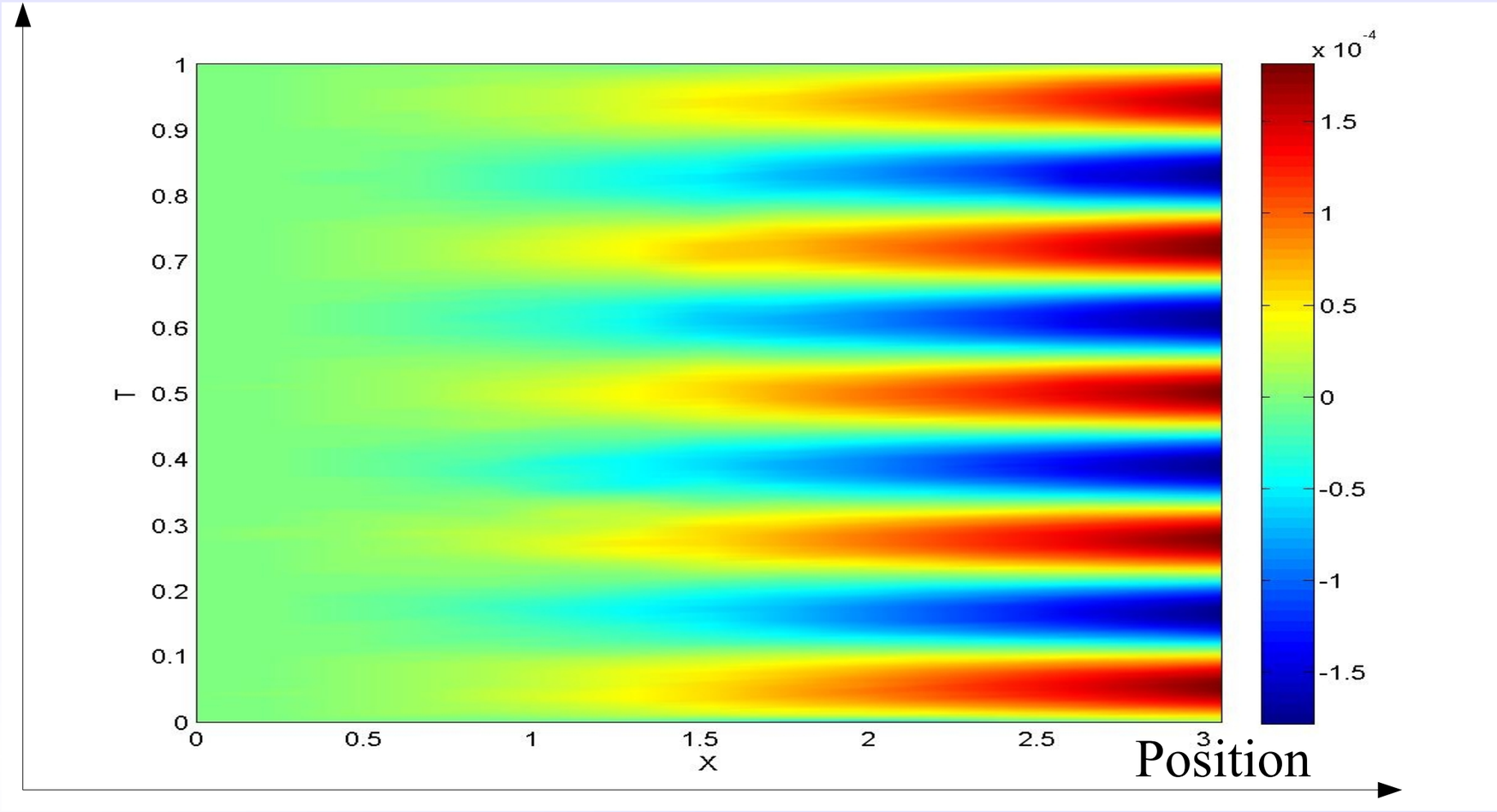
Application to Beam Modeling

Weeks method engine can be used to invert W(x,s;E(s))



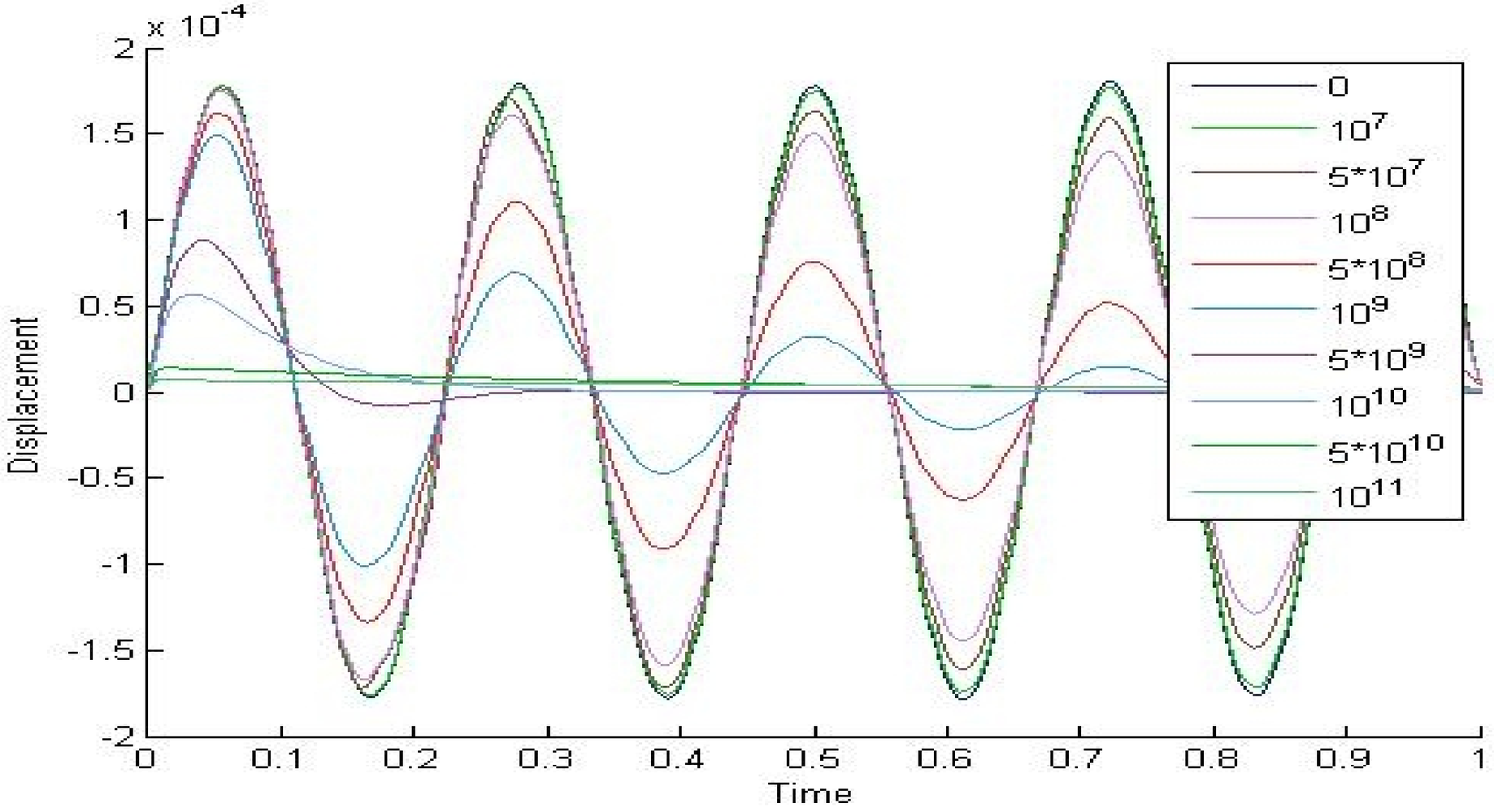
Beam Impulsive Force Impact

Oscillation of an elastic cantilever beam struck by an impulsive force.



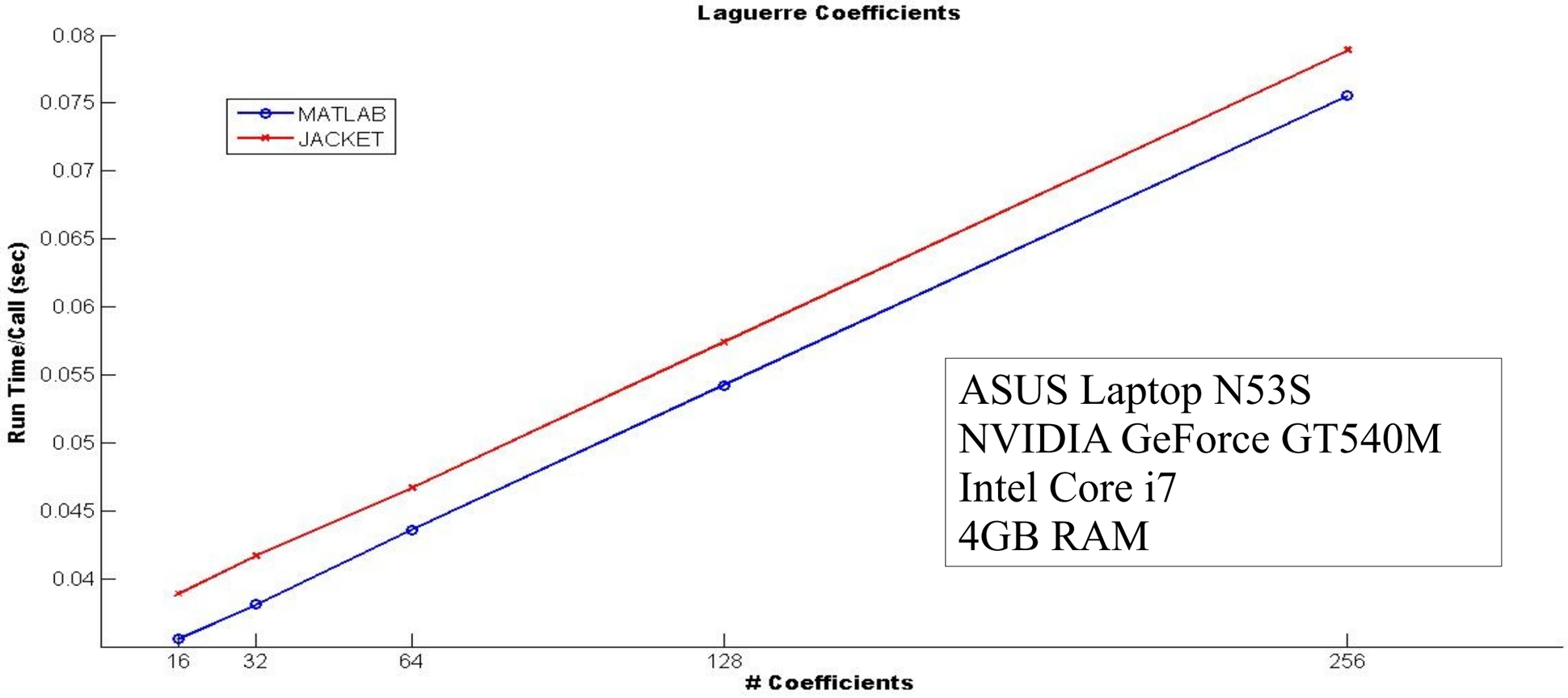
Beam Oscillation versus Viscosity

A transition from underdamped to overdamped oscillations is observed as the viscoelastic material viscosity is increased.



Expansion Coefficient Timing Tests

Computation of the expansion coefficients {a} is slightly slower with JACKET than standard MATLAB.

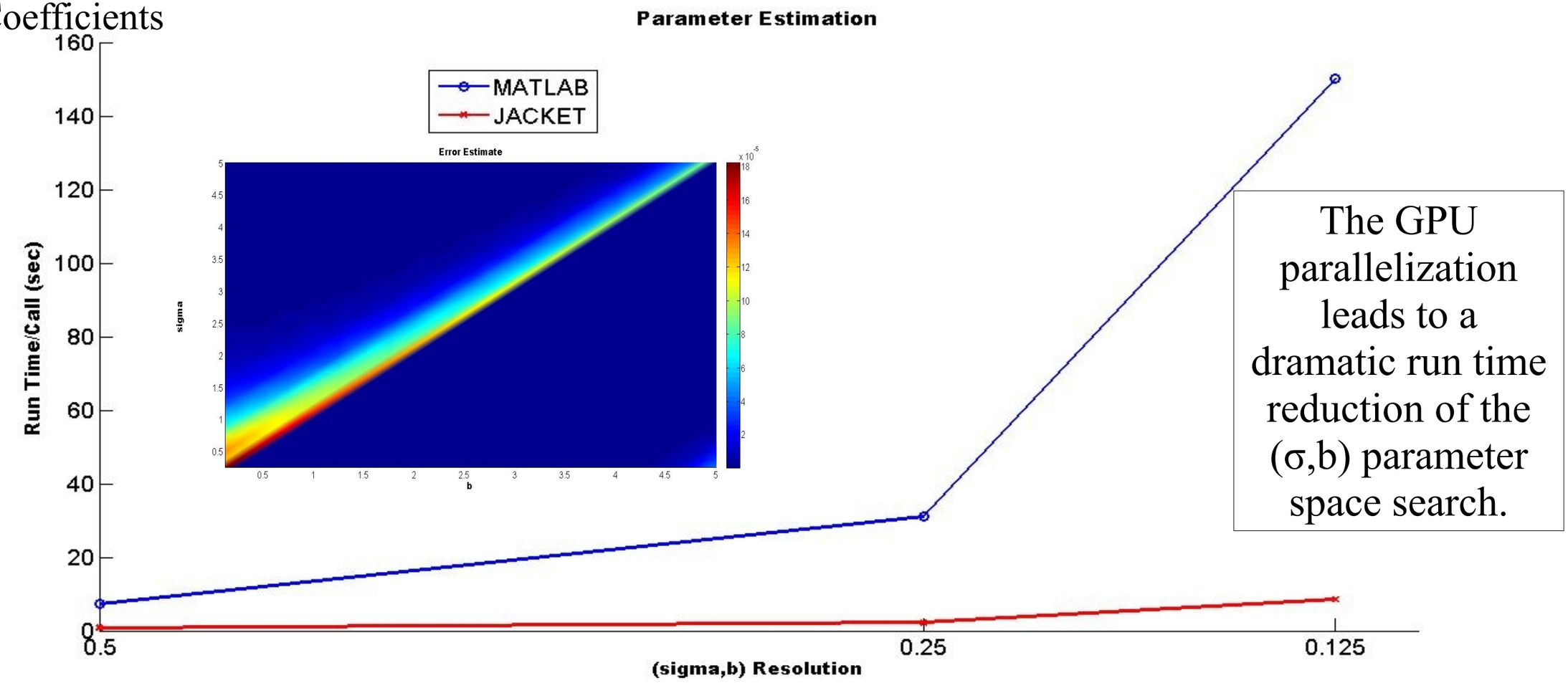


ASUS Laptop N53S
NVIDIA GeForce GT540M
Intel Core i7
4GB RAM

Run time grows linearly with the number of coefficients.

Parameter Estimation Timing Tests

256 Coefficients



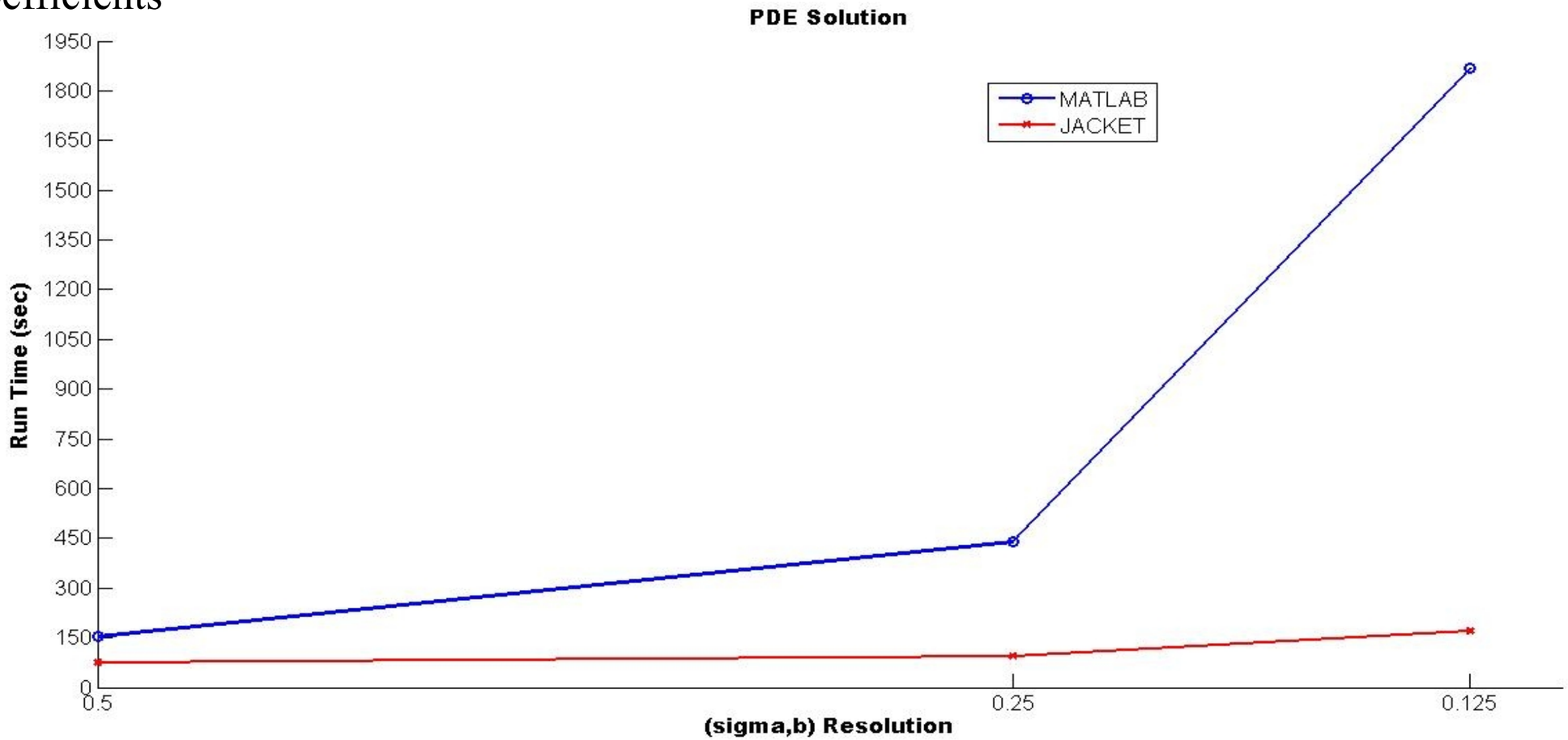
The GPU parallelization leads to a dramatic run time reduction of the (σ,b) parameter space search.

The GPU parallelization more than compensates for the small increase in Laguerre coefficient calculation run time incurred using JACKET.

PDE Simulation Timing Tests

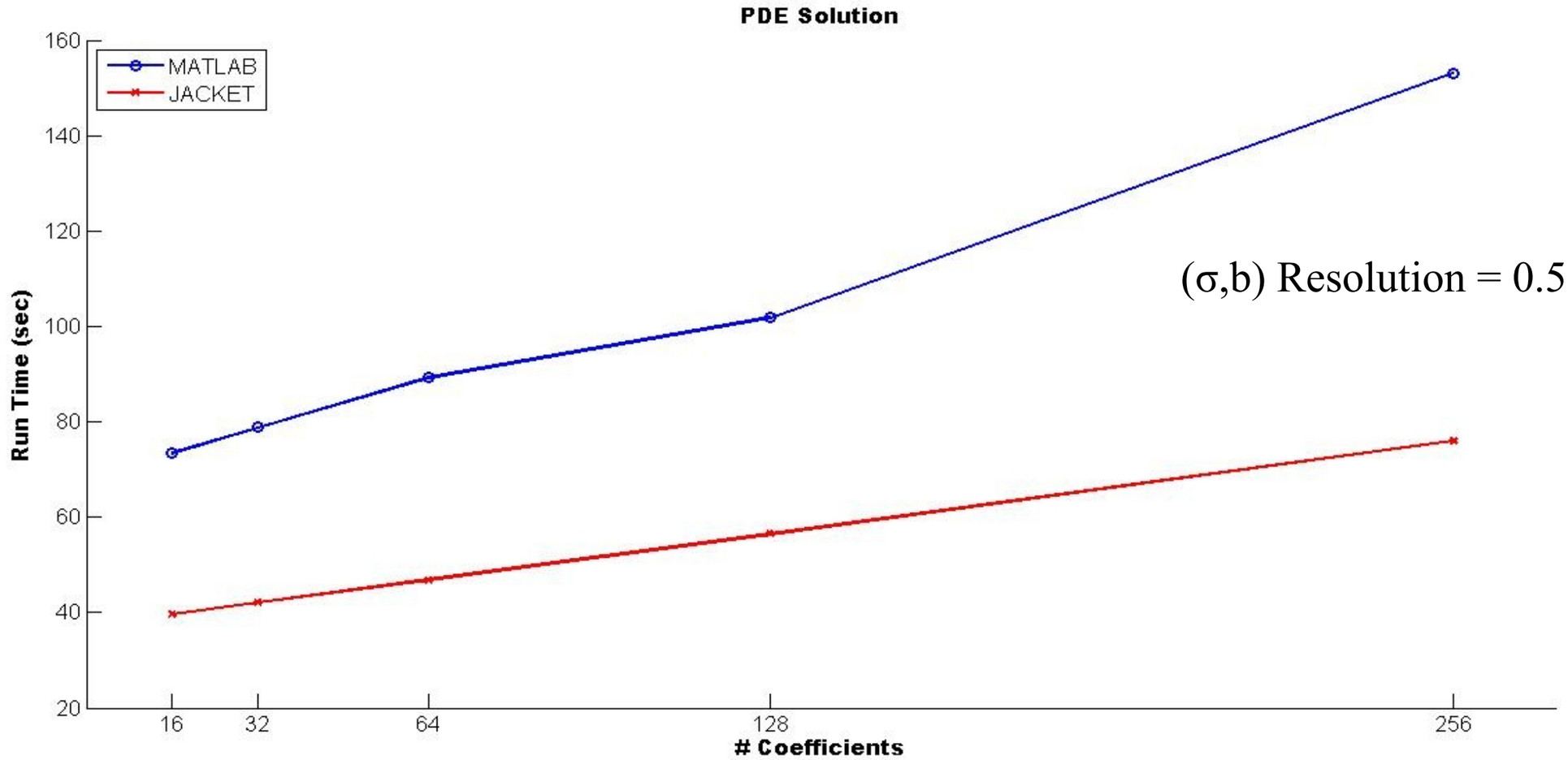
The total PDE simulation run time has the same behavior as the (σ,b) parameter search run time.

256 Coefficients



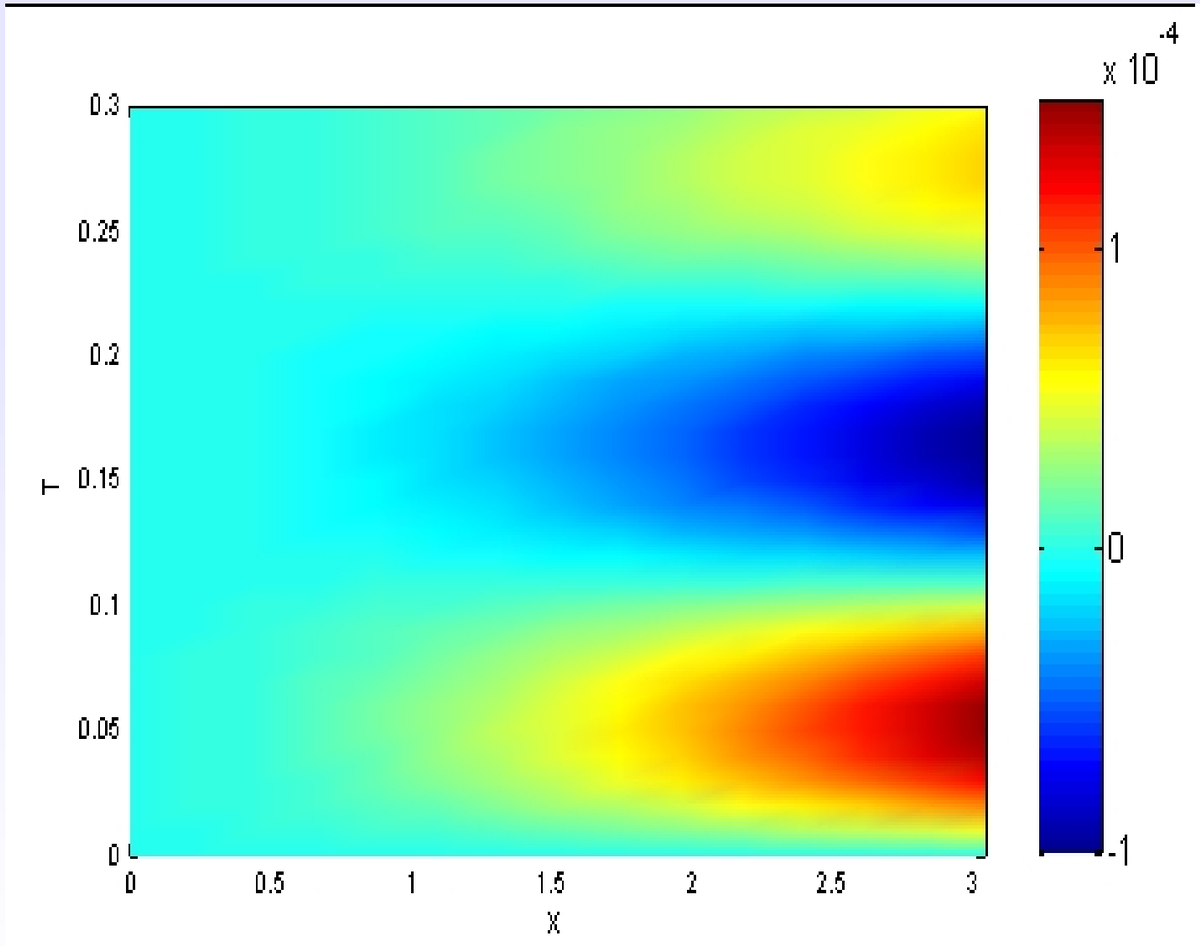
PDE Simulation Timing Tests

The (σ,b) search space resolution has a greater effect on the run time than the number of Laguerre expansion coefficients.

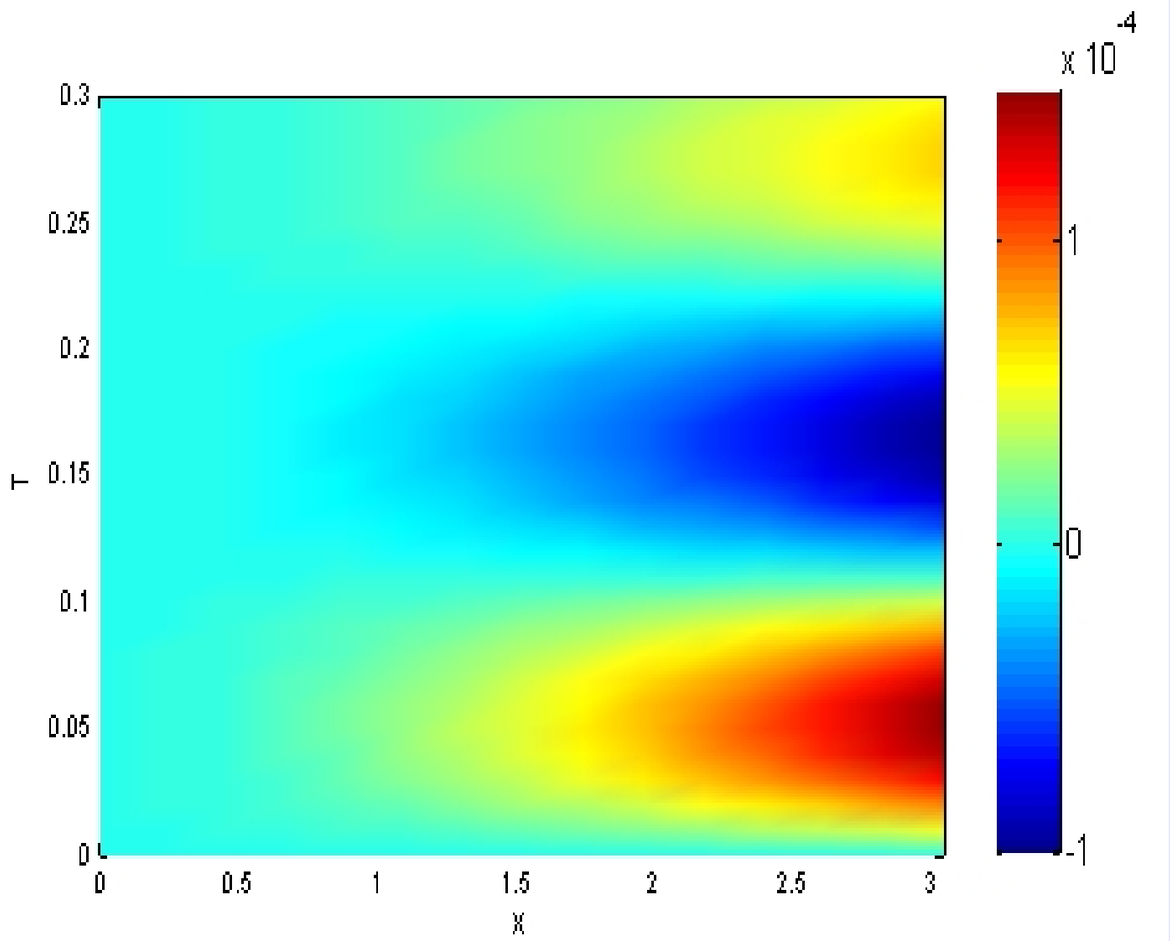


CPU versus GPU Simulations

The CPU and GPU simulation results are very similar.



CPU



GPU

Potential Impact and Future Directions

Software that can provide **reliable** and **fast** numerical Laplace transform inversion can make a considerable impact in a number of fields:

- Viscoelastic beams
- Pulse propagation in highly dispersive materials (nontrival fluids, biomatter)
- Gaussian beam propagation in optical elements
- Hydrology
- Finance and market prediction

- The Weeks method is only one numerical inversion method.
- Other methods, such as Talbot's or Post's, may be more appropriate for certain classes of problems.
- Integrating GPU acceleration with these other Laplace inversion methods is an interesting and worth-while challenge.

BACKUPS

More Complex Mechanical Models

- More complicated mechanical models can be obtained by following rules for parallel and series combinations of elements.

Parallel

$$\epsilon_{total} = \epsilon_j \forall j$$
$$\sigma_{total} = \sum_j \sigma_j$$
$$E_{total} = \sum_j E_j$$

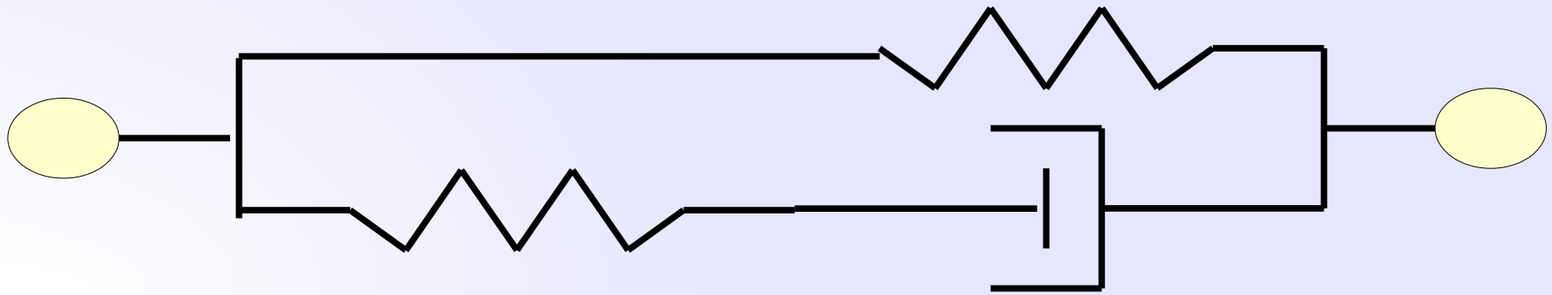
spring	$E \rightarrow E_k$
damper	$E \rightarrow \eta_k \frac{d}{dt}$

Series

$$\sigma_{total} = \sigma_j \forall j$$
$$\epsilon_{total} = \sum_j \epsilon_j$$
$$\frac{1}{E_{total}} = \sum_j \frac{1}{E_j}$$

- Example: **standard linear solid model** involves the parallel combination of the Maxwell & Voight models.

$$\frac{d\epsilon(t)}{dt} = \frac{1}{E_1 + E_2}\left[\frac{d\sigma(t)}{dt} + \frac{E_2}{\eta}\sigma(t) - \frac{E_1 E_2}{\eta}\epsilon(t)\right]$$



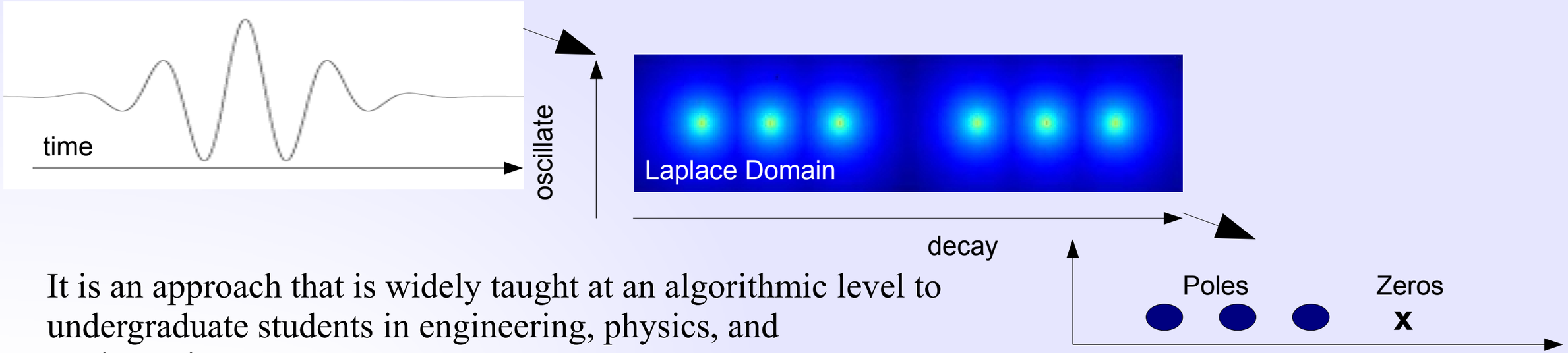
Basic Definitions

The Laplace Transform is tool to convert a difficult problem into a simpler one.

Difficult Time Dependent Problem	Solve Simpler Laplace Space Problem	Invert to a Time Dependent Solution

$$P\{F(t)\} \longrightarrow L \longrightarrow F(s) \longrightarrow L^{-1} \longrightarrow f(t)$$

It transforms a time dependent signal into its oscillating and exponentially decaying components.



It is an approach that is widely taught at an algorithmic level to undergraduate students in engineering, physics, and mathematics.

Selected Numerical Inversion Methods

Of the numerous numerical inversion algorithms, my own research has focused on three of the more well known:

1. Weeks' Method
   - "Application of Weeks method for the numerical inversion of the Laplace transform to the matrix exponential", P. Kano, M. Brio, published 2009
   - "C++/CUDA implementation of the Weeks method for numerical Laplace transform inversion", P. Kano, M. Brio, Acunum white paper 2011
2. Post's Formula
   - "Application of Post's formula to optical pulse propagation in dispersive media", P. Kano, M. Brio, published 2010
3. Talbot's Method
   - "Dempster-Shafer evidential theory for the automated selection of parameters for Talbot's method contours and application to matrix exponentiation", P. Kano, M. Brio, P. Dostert, J. Cain, in review 2011

In the remaining slides, I introduce each of the algorithms and discuss my own applications.

Technology behind Acunum Applications

NVIDIA
Graphics Processing Units

ACUNUM
Matlab/Jacket & C/C++
Numerical
Laplace Transform Inversion
Toolbox

Acunum released a numerical inversion tool to the web for public use.

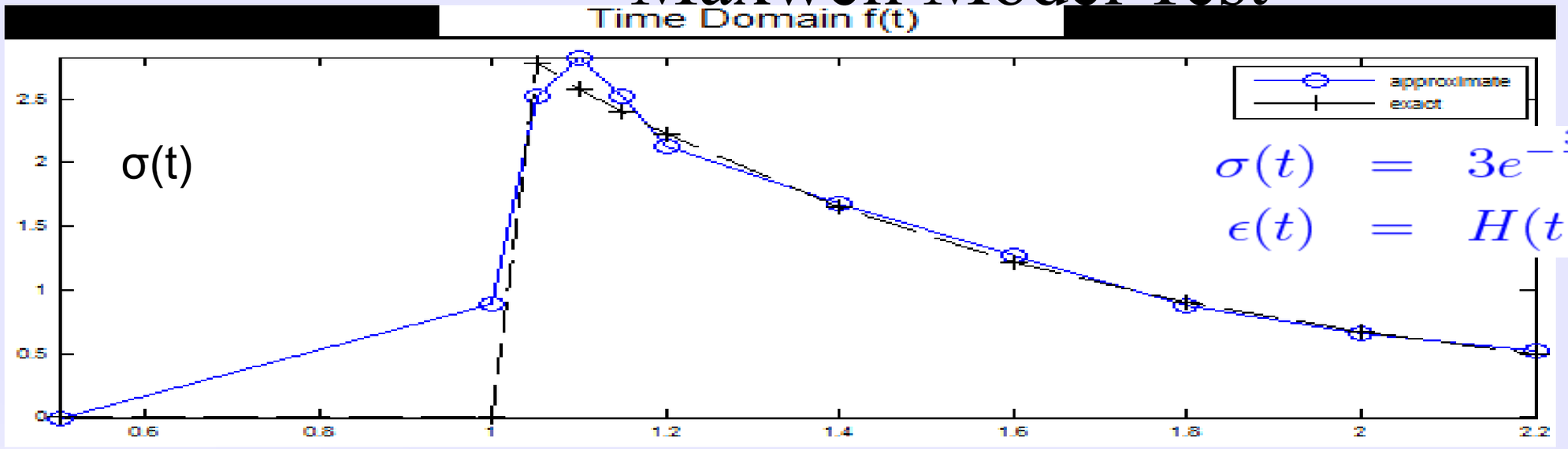
ACUNUM
C/C++
Dempster-Shafer
Data Fusion

Acunum is developing a fast GPU accelerated algorithm for sensor data fusion and object classification.

http://www.accelereyes.com/examples/academia

Our tools uses JACKET from AccelerEyes, Inc. for the MATLAB/GPU interface.

Maxwell Model Test



Stress Relaxation

$$\sigma(t) = 3e^{-\frac{3(t-1)}{2}} H(t-1)$$

$$\epsilon(t) = H(t-1)$$

The Laguerre polynomial basis can not fully capture the instant jump.