Numerical investigation of movable singularities
@ CASC 2014

Radosław A. Kycia
rkycia@mimuw.edu.pl

The Faculty of Mathematics, Informatics and Mechanics
The Univeristy of Warsaw
Poland

August 9, 2014

Outline

Singularities of ODEs (Ordinary Differential Equations)

Singularities of ODEs

Nonlinear ODEs posses two types of singularities:

- fixed - singularities of the coefficients of ODE
- movable - the singularities of solutions; position depends on initial data; not present in linear ODEs;

Example [Goriely]

The equation

$$\dot{x} = x^3, \quad x(t_0) = x_0$$

has the solutions

$$x(t) = (2(t_0 - t) + x_0^{-2})^{-1/2}.$$

Singularities of ODEs

Nonlinear ODEs posses two types of singularities:

- fixed - singularities of the coefficients of ODE
- movable - the singularities of solutions; position depends on initial data; not present in linear ODEs;

Example [Goriely]

The equation

$$\dot{x} = x^3, \quad x(t_0) = x_0$$

has the solutions

$$x(t) = (2(t_0 - t) + x_0^{-2})^{-1/2}.$$

The Painlevé property/Painlevé test

- Cauchy approach - local existence
- Painlevé approach - global existence, finite form and single valuedness
- Solutions can be globally defined only when we know how to define its Riemann surface, i.e., the only movable singularities are poles.
- Deduce global structure of solution (types of singularities) from the local behaviour around some points in the complex plane. Only sufficient conditions $\rightarrow$ by the contraposition - it gives a result when it fails.
- In application, global existence is sometimes important.

The Painlevé property/Painlevé test

- Cauchy approach - local existence
- Painlevé approach - global existence, finite form and single valuedness
- Solutions can be globally defined only when we know how to define its Riemann surface, i.e., the only movable singularities are poles.
- Deduce global structure of solution (types of singularities) from the local behaviour around some points in the complex plane. Only sufficient conditions $\rightarrow$ by the contraposition - it gives a result when it fails.
- In application, global existence is sometimes important.

The Painlevé property/Painlevé test

- Cauchy approach - local existence
- Painlevé approach - global existence, finite form and single valuedness
- Solutions can be globally defined only when we know how to define its Riemann surface, i.e., the only movable singularities are poles.
- Deduce global structure of solution (types of singularities) from the local behaviour around some points in the complex plane. Only sufficient conditions $\rightarrow$ by the contraposition - it gives a result when it fails.
- In application, global existence is sometimes important.

The Painlevé property/Painlevé test

- Cauchy approach - local existence
- Painlevé approach - global existence, finite form and single valuedness
- Solutions can be globally defined only when we know how to define its Riemann surface, i.e., the only movable singularities are poles.
- Deduce global structure of solution (types of singularities) from the local behaviour around some points in the complex plane. Only sufficient conditions $\rightarrow$ by the contraposition - it gives a result when it fails.
- In application, global existence is sometimes important.

The Painlevé property/Painlevé test

- Cauchy approach - local existence
- Painlevé approach - global existence, finite form and single valuedness
- Solutions can be globally defined only when we know how to define its Riemann surface, i.e., the only movable singularities are poles.
- Deduce global structure of solution (types of singularities) from the local behaviour around some points in the complex plane. Only sufficient conditions $\rightarrow$ by the contraposition - it gives a result when it fails.
- In application, global existence is sometimes important.

Problem Statement

Problem Statement

- ODE as a system of first order DE

$$\frac{d\vec{y}(x)}{dx} = \vec{f}(\vec{y}; x), \quad \vec{y}(x) : x \in \mathbb{C} \to \mathbb{C}^n. \tag{1}$$

- Initial value $\vec{y}(x_0) = \vec{y}_0$.
- Path, e.g., $(t \in \mathbb{R}^+)$
  - Semiline $x(t) = x_0 + (t + shift) \cdot e^{i\phi}$
  - Spiral $x(t) = (x_0 + (at + b)e^{i \cdot dir \cdot t})e^{i\phi}$
- Domain - path connected region (ideally connected by paths along which integration is performed).
- Condition for singularity proximity - the crude estimation $||\vec{y}|| < \mathrm{Large\ \ const}$. Not the state of art, but it can be improved.

Problem Statement

- ODE as a system of first order DE

$$\frac{d\vec{y}(x)}{dx} = \vec{f}(\vec{y};x), \quad \vec{y}(x) : x \in \mathbb{C} \to \mathbb{C}^n. \tag{1}$$

- Initial value $\vec{y}(x_0) = \vec{y}_0$.
- Path, e.g., $(t \in \mathbb{R}^+)$
  - Semiline $x(t) = x_0 + (t + shift) \cdot e^{i\phi}$
  - Spiral $x(t) = (x_0 + (at + b)e^{i \cdot dir \cdot t})e^{i\phi}$
- Domain - path connected region (ideally connected by paths along which integration is performed).
- Condition for singularity proximity - the crude estimation $||\vec{y}|| < \mathrm{Large\ \ const.}$ Not the state of art, but it can be improved.

Problem Statement

- ODE as a system of first order DE

$$\frac{d\vec{y}(x)}{dx} = \vec{f}(\vec{y};x), \quad \vec{y}(x) : x \in \mathbb{C} \to \mathbb{C}^n. \tag{1}$$

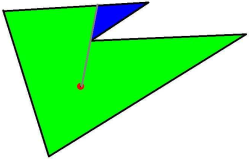
- Initial value $\vec{y}(x_0) = \vec{y}_0$.
- Path, e.g., $(t \in \mathbb{R}^+)$
  - Semiline $x(t) = x_0 + (t + shift) \cdot e^{i\phi}$
  - Spiral $x(t) = (x_0 + (at + b)e^{i \cdot dir \cdot t})e^{i\phi}$
- Domain - path connected region (ideally connected by paths along which integration is performed).
- Condition for singularity proximity - the crude estimation $||\vec{y}|| < \mathrm{Large\_const}$. Not the state of art, but it can be improved.

Problem Statement

- ODE as a system of first order DE

$$\frac{d\vec{y}(x)}{dx} = \vec{f}(\vec{y};x), \quad \vec{y}(x) : x \in \mathbb{C} \to \mathbb{C}^n. \tag{1}$$

- Initial value $\vec{y}(x_0) = \vec{y}_0$.
- Path, e.g., ($t \in \mathbb{R}^+$)
  - Semiline $x(t) = x_0 + (t + shift) \cdot e^{i\phi}$
  - Spiral $x(t) = (x_0 + (at + b)e^{i \cdot dir \cdot t})e^{i\phi}$
- Domain - path connected region (ideally connected by paths along which integration is performed).
- Condition for singularity proximity - the crude estimation $||\vec{y}|| < \text{Large const}$. Not the state of art, but it can be improved.

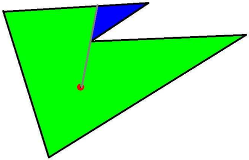


Problem Statement

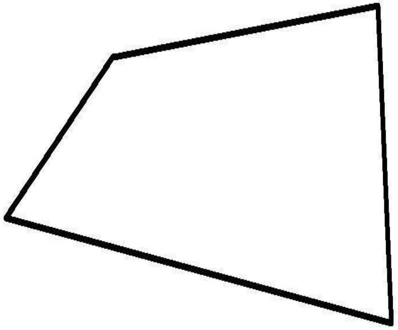
- ODE as a system of first order DE

$$\frac{d\vec{y}(x)}{dx} = \vec{f}(\vec{y};x), \quad \vec{y}(x) : x \in \mathbb{C} \to \mathbb{C}^n. \tag{1}$$

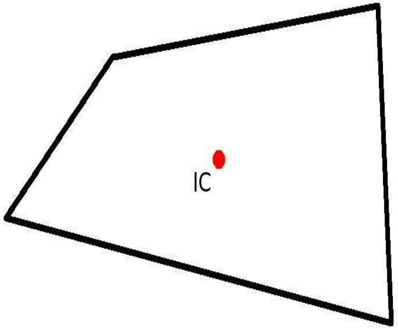
- Initial value $\vec{y}(x_0) = \vec{y}_0$.
- Path, e.g., $(t \in \mathbb{R}^+)$
  - Semiline $x(t) = x_0 + (t + shift) \cdot e^{i\phi}$
  - Spiral $x(t) = (x_0 + (at + b)e^{i \cdot dir \cdot t})e^{i\phi}$
- Domain - path connected region (ideally connected by paths along which integration is performed).
- Condition for singularity proximity - the crude estimation $||\vec{y}|| < \mathrm{Large\ \ const.}$ Not the state of art, but it can be improved.



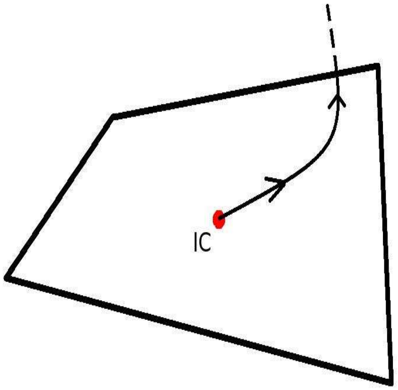
Domain



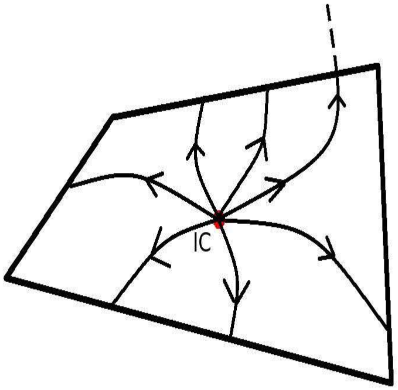
Initial Conditions



Integration along path



Full integration



Implementation

$$\frac{d\vec{y}(x)}{dx} = \vec{f}(\vec{y};x), \quad \vec{y}(x) : x \in \mathbb{C} \to \mathbb{C}^n. \tag{2}$$

If path if prescribed by C^1 curve then the integration can be viewed as an integration on $\mathbb{R}^+$ of the **pullback** of the equation on the path

$$\begin{cases} \frac{d\vec{y}}{dt} = p'(t)\vec{f}(x(t), \vec{y}(t)) \\ x'(t) = p'(t), \end{cases} \tag{3}$$

where $' = \frac{d}{dt}$.

This system of ODEs can be integrated using standard Mathematica algorithms.

In the monitor function the conjunction of x being in the domain **and** proximity of a singularity have to be checked.

$$\frac{d\vec{y}(x)}{dx} = \vec{f}(\vec{y};x), \quad \vec{y}(x) : x \in \mathbb{C} \to \mathbb{C}^n. \tag{2}$$

If path if prescribed by C^1 curve then the integration can be viewed as an integration on $\mathbb{R}^+$ of the **pullback** of the equation on the path

$$\begin{cases} \frac{d\vec{y}}{dt} = p'(t)\vec{f}(x(t),\vec{y}(t)) \\ x'(t) = p'(t), \end{cases} \tag{3}$$

where $' = \frac{d}{dt}$.

This system of ODEs can be integrated using standard Mathematica algorithms.

In the monitor function the conjunction of x being in the domain **and** proximity of a singularity have to be checked.

$$\frac{d\vec{y}(x)}{dx} = \vec{f}(\vec{y};x), \quad \vec{y}(x) : x \in \mathbb{C} \to \mathbb{C}^n. \tag{2}$$

If path if prescribed by C^1 curve then the integration can be viewed as an integration on $\mathbb{R}^+$ of the **pullback** of the equation on the path

$$\begin{cases} \frac{d\vec{y}}{dt} = p'(t)\vec{f}(x(t),\vec{y}(t)) \\ x'(t) = p'(t), \end{cases} \tag{3}$$

where $' = \frac{d}{dt}$.

This system of ODEs can be integrated using standard Mathematica algorithms.

In the monitor function the conjunction of x being in the domain **and** proximity of a singularity have to be checked.

C++

- Implemented using Object Oriented approach...
- ...however, can be used in a functional way.
- Mapper class - maps a path (using ODE, IC, Domain constraints, numerical solver) onto the solution along the path. Some resemblance to the higher-order functions.
- Easy to parallelize as the integrations along curves are independent - parallel producer-consumer problem.

- Implemented using Object Oriented approach...
- ...however, can be used in a functional way.
- Mapper class - maps a path (using ODE, IC, Domain constraints, numerical solver) onto the solution along the path. Some resemblance to the higher-order functions.
- Easy to parallelize as the integrations along curves are independent - parallel producer-consumer problem.

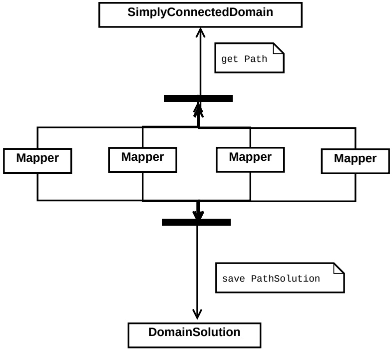
C++

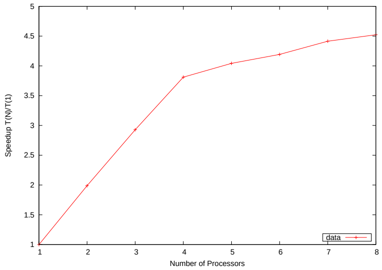
- Implemented using Object Oriented approach...
- ...however, can be used in a functional way.
- Mapper class - maps a path (using ODE, IC, Domain constraints, numerical solver) onto the solution along the path. Some resemblance to the higher-order functions.
- Easy to parallelize as the integrations along curves are independent - parallel producer-consumer problem.

C++

- Implemented using Object Oriented approach...
- ...however, can be used in a functional way.
- Mapper class - maps a path (using ODE, IC, Domain constraints, numerical solver) onto the solution along the path. Some resemblance to the higher-order functions.
- Easy to parallelize as the integrations along curves are independent - parallel producer-consumer problem.

C++ (parallel)





The Amdahl's law is preserved at the beginning up to 4 threads - parallel executed code is about 90%. Then processes start to block each other. New way of parallelization needed.

Examples

Equations [Kycia, Filipuk]

The Emden-Fowler equation

$$\frac{d^2u(x)}{dx^2} + \frac{\alpha}{x}\frac{du(x)}{dx} + x^n u(x)^p = 0 \qquad (4)$$

Generalized Isothermal Sphere equation

$$\frac{d^2u(x)}{dx^2} + \frac{\alpha}{x}\frac{du(x)}{dx} - x^n e^{-u(x)} = 0, \qquad (5)$$

$u(0) = 0$

Location of singularities

A nonzero analytic solutions of the Generlized Emden-Fowler and Isothermal Sphere equations have $n+2$ singularities located symmetrically with respect to the origin on the rays connecting the origin with all $(n+2)$ roots of -1 in the complex plane.

Equations [Kycia, Filipuk]

The Emden-Fowler equation

$$\frac{d^2u(x)}{dx^2} + \frac{\alpha}{x}\frac{du(x)}{dx} + x^n u(x)^p = 0 \tag{4}$$

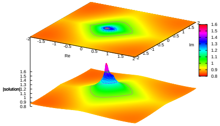
Generalized Isothermal Sphere equation

$$\frac{d^2u(x)}{dx^2} + \frac{\alpha}{x}\frac{du(x)}{dx} - x^n e^{-u(x)} = 0, \tag{5}$$

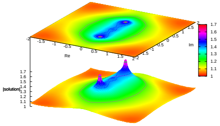
$u(0) = 0$

Location of singularities

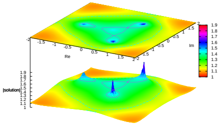
A nonzero analytic solutions of the Generlized Emden-Fowler and Isothermal Sphere equations have $n+2$ singularities located symmetrically with respect to the origin on the rays connecting the origin with all $(n+2)$ roots of -1 in the complex plane.



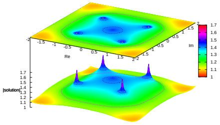
(a) $n = -1$



(b) $n = 0$



(c) $n = 1$



(d) $n = 2$

Figure : $p = 5$ and $u(0) = 1.5$, the Generalized Emden-Fowler solution.

Generalized isothermal sphere equations

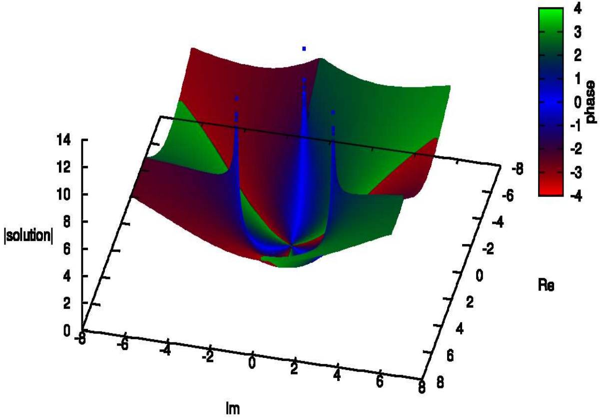


Figure : $u(0) = 0$, $n = 1$

Conclusions

Conclusions

- The method is simple, straightforward, brute force but it works.
- C++ code is good for robust (HPC) computations and it is ready for linking with Mathematica by MathLink to provide 'user-friendly' interface.
- The code is flexible - it can be extended by new methods of integration, types of domains, curves (more effective swapping of domain), methods for determining proximity of singularities, etc.

Conclusions

- The method is simple, straightforward, brute force but it works.
- C++ code is good for robust (HPC) computations and it is ready for linking with Mathematica by MathLink to provide 'user-friendly' interface.
- The code is flexible - it can be extended by new methods of integration, types of domains, curves (more effective swapping of domain), methods for determining proximity of singularities, etc.

Conclusions

- The method is simple, straightforward, brute force but it works.
- C++ code is good for robust (HPC) computations and it is ready for linking with Mathematica by MathLink to provide 'user-friendly' interface.
- The code is flexible - it can be extended by new methods of integration, types of domains, curves (more effective swapping of domain), methods for determining proximity of singularities, etc.

Bibliography

- R. Conte (editor) **The Painlevé property. One century later** CRM Series in Mathematical Physics, Springer 1999
- A. Goriely, **Integrability and Nonintegrability of Dynamical Systems** , World Scientific (2001)
- C. Hunter, **Series solutions for polytropes and the isothermal sphere**, Mon. Not. R. Astron. Soc. 328 (2001), 839–847.
- R. Kycia, G. Filipuk, **On the singularities of the Emden-Fowler type equations**, Proceedings of ISAAC Conference (2014)
- My homepage `http://www.mimuw.edu.pl/~rkycia/`

Thank You for Your Attention

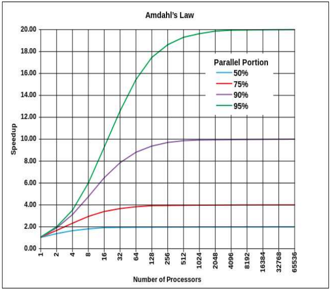
Backup

Amdahl's law

Amdahl's law

$$S(n) = \frac{T(n)}{T(1)} = \frac{1}{B + \frac{1}{n}(1 - B)}$$

n - no. threads of execution;
$B \in [0;1]$ - the fraction of the algorithm that is strictly serial;
$T(n)$ - time of execution of n threads;



see, $http://en.wikipedia.org/wiki/Amdahl\%27s\_law$