60 years of stiff solvers

John Butcher, University of Auckland

SciCADE 2011, Toronto

Jan Verwer Memorial Minisyposium

13 July 2011

Contents

Contents

Contents

Contents

Contents

INTRODUCTION

60 years of stiffness

The seminal paper of Curtiss and Hirschfelder was communicated to the Proceedings of the National Academy of Sciences in 1951 and published in the following year.

The use of the backward Euler method, which can be regarded as the first stiff solver, was advocated as a stable alternative to the forward Euler method.

Techniques for obtaining stable, accurate and efficient solutions to stiff problems have become an important field of research in many institutions and laboratories throughout the world, not least being in the work of Jan Verwer and his colleagues at the CWI.

This talk will not attempt to survey the enormous history of this subject but will give an account of some items from this history of interest to the author.

60 years of stiffness

The seminal paper of Curtiss and Hirschfelder was communicated to the Proceedings of the National Academy of Sciences in 1951 and published in the following year.

The use of the backward Euler method, which can be regarded as the first stiff solver, was advocated as a stable alternative to the forward Euler method.

Techniques for obtaining stable, accurate and efficient solutions to stiff problems have become an important field of research in many institutions and laboratories throughout the world, not least being in the work of Jan Verwer and his colleagues at the CWI.

This talk will not attempt to survey the enormous history of this subject but will give an account of some items from this history of interest to the author.

60 years of stiffness

The seminal paper of Curtiss and Hirschfelder was communicated to the Proceedings of the National Academy of Sciences in 1951 and published in the following year.

The use of the backward Euler method, which can be regarded as the first stiff solver, was advocated as a stable alternative to the forward Euler method.

Techniques for obtaining stable, accurate and efficient solutions to stiff problems have become an important field of research in many institutions and laboratories throughout the world, not least being in the work of Jan Verwer and his colleagues at the CWI.

This talk will not attempt to survey the enormous history of this subject but will give an account of some items from this history of interest to the author.

60 years of stiffness

The seminal paper of Curtiss and Hirschfelder was communicated to the Proceedings of the National Academy of Sciences in 1951 and published in the following year.

The use of the backward Euler method, which can be regarded as the first stiff solver, was advocated as a stable alternative to the forward Euler method.

Techniques for obtaining stable, accurate and efficient solutions to stiff problems have become an important field of research in many institutions and laboratories throughout the world, not least being in the work of Jan Verwer and his colleagues at the CWI.

This talk will not attempt to survey the enormous history of this subject but will give an account of some items from this history of interest to the author.

Questions about stiffness

What are stiff problems?

- I will try to give some sort of answer to this perennial question

What methods are useful?

- Are multivalue methods useful?
- Are multistage methods useful?
- Are multivalue-multistage methods useful?

Stability questions

- A-stability
- G-stability
- L-stability
- XYZ-stability

Implementation questions

- Overcoming the disadvantages of expensive Newton iterations?

Questions about stiffness

What are stiff problems?

- I will try to give some sort of answer to this perennial question

What methods are useful?

- Are multivalue methods useful?
- Are multistage methods useful?
- Are multivalue-multistage methods useful?

Stability questions

- A-stability
- G-stability
- L-stability
- XYZ-stability

Implementation questions

- Overcoming the disadvantages of expensive Newton iterations?

Questions about stiffness

What are stiff problems?

I will try to give some sort of answer to this perennial question

What methods are useful?

Are multivalue methods useful?

Are multistage methods useful?

Are multivalue-multistage methods useful?

Stability questions

A-stability

G-stability

L-stability

XYZ-stability

Implementation questions

Overcoming the disadvantages of expensive Newton iterations?

Questions about stiffness

What are stiff problems?

I will try to give some sort of answer to this perennial question

What methods are useful?

Are multivalue methods useful?

Are multistage methods useful?

Are multivalue-multistage methods useful?

Stability questions

A-stability

G-stability

L-stability

XYZ-stability

Implementation questions

Overcoming the disadvantages of expensive Newton iterations?

WHAT ARE STIFF PROBLEMS?

Learning through definitions

Some mathematical ideas are best understood in terms of formal definitions.

Possible applications and intuitive understanding of meanings then become somewhat secondary.

But for other mathematical ideas, the intuitive notion, perhaps based on observable phenomena, is the starting point and formal definitions come later, if they come at all.

I think that stiffness is like this.

We want to understand this phenomenon and think about how to deal with it.

Definitions can give an insight into some aspects of what stiff problems are but they don't necessarily express every aspect in a comprehensive way.

Learning through definitions

Some mathematical ideas are best understood in terms of formal definitions.

Possible applications and intuitive understanding of meanings then become somewhat secondary.

But for other mathematical ideas, the intuitive notion, perhaps based on observable phenomena, is the starting point and formal definitions come later, if they come at all.

I think that stiffness is like this.

We want to understand this phenomenon and think about how to deal with it.

Definitions can give an insight into some aspects of what stiff problems are but they don't necessarily express every aspect in a comprehensive way.

Learning through definitions

Some mathematical ideas are best understood in terms of formal definitions.

Possible applications and intuitive understanding of meanings then become somewhat secondary.

But for other mathematical ideas, the intuitive notion, perhaps based on observable phenomena, is the starting point and formal definitions come later, if they come at all.

I think that stiffness is like this.

We want to understand this phenomenon and think about how to deal with it.

Definitions can give an insight into some aspects of what stiff problems are but they don't necessarily express every aspect in a comprehensive way.

Learning through definitions

Some mathematical ideas are best understood in terms of formal definitions.

Possible applications and intuitive understanding of meanings then become somewhat secondary.

But for other mathematical ideas, the intuitive notion, perhaps based on observable phenomena, is the starting point and formal definitions come later, if they come at all.

I think that stiffness is like this.

We want to understand this phenomenon and think about how to deal with it.

Definitions can give an insight into some aspects of what stiff problems are but they don't necessarily express every aspect in a comprehensive way.

Learning through definitions

Some mathematical ideas are best understood in terms of formal definitions.

Possible applications and intuitive understanding of meanings then become somewhat secondary.

But for other mathematical ideas, the intuitive notion, perhaps based on observable phenomena, is the starting point and formal definitions come later, if they come at all.

I think that stiffness is like this.

We want to understand this phenomenon and think about how to deal with it.

Definitions can give an insight into some aspects of what stiff problems are but they don't necessarily express every aspect in a comprehensive way.

Learning through examples

Let's look at some initial value problems and see which are stiff

$(a)\quad y' = y, \qquad y(0) = 1,$ $\qquad (b)\quad y' = 100y, \qquad y(0) = 1,$

$(c)\quad y' = -y, \qquad y(0) = 1,$ $\qquad (d)\quad y' = -100y, \qquad y(0) = 1,$

$(e)\quad y' = -y + \sin(x), \quad y(0) = 1,$ $\qquad (f)\quad y' = -100y + \sin(x), \quad y(0) = 1,$

$(g)\quad y' = My + \phi(x),$ $\qquad (h)\quad y' = M(x)y + \phi(x),$

Large negative $\lambda \in \sigma(M)$

$(i)\quad y' = f(y),$ $\qquad (j)\quad y' = f(x, y)$

$z' = (\frac{\partial f}{\partial y})z,$

Learning through examples

Let's look at some initial value problems and see which are stiff

(a) $y' = y, \quad y(0) = 1,$

(b) $y' = 100y, \quad y(0) = 1,$

(c) $y' = -y, \quad y(0) = 1,$

(d) $y' = -100y, \quad y(0) = 1,$

(e) $y' = -y + \sin(x), \quad y(0) = 1,$

(f) $y' = -100y + \sin(x), \quad y(0) = 1,$

(g) $y' = My + \phi(x),$
Large negative $\lambda \in \sigma(M)$

(h) $y' = M(x)y + \phi(x),$

(i) $y' = f(y),$
$z' = (\frac{\partial f}{\partial y})z,$

(j) $y' = f(x, y)$

Learning through examples

Let's look at some initial value problems and see which are stiff

$(a)\ \ y' = y, \quad y(0) = 1,$ $\qquad (b)\ \ y' = 100y, \quad y(0) = 1,$

$(c)\ \ y' = -y, \quad y(0) = 1,$ $\qquad (d)\ \ y' = -100y, \quad y(0) = 1,$

$(e)\ \ y' = -y + \sin(x), \quad y(0) = 1,$ $\qquad (f)\ \ y' = -100y + \sin(x), \quad y(0) = 1,$

$(g)\ \ y' = My + \phi(x),$ Large negative $\lambda \in \sigma(M)$ $\qquad (h)\ \ y' = M(x)y + \phi(x),$

$(i)\ \ y' = f(y), \quad z' = (\frac{\partial f}{\partial y})z,$ $\qquad (j)\ \ y' = f(x, y)$

Learning through examples

Let's look at some initial value problems and see which are stiff

(a) $y' = y, \quad y(0) = 1,$ (b) $y' = 100y, \quad y(0) = 1,$

(c) $y' = -y, \quad y(0) = 1,$ (d) $y' = -100y, \quad y(0) = 1,$

(e) $y' = -y + \sin(x), \quad y(0) = 1,$ (f) $y' = -100y + \sin(x), \quad y(0) = 1,$

(g) $y' = My + \phi(x),$
Large negative $\lambda \in \sigma(M)$

(h) $y' = M(x)y + \phi(x),$

(i) $y' = f(y),$
$z' = (\frac{\partial f}{\partial y})z,$

(j) $y' = f(x, y)$

Learning through examples

Let's look at some initial value problems and see which are stiff

$(a)\;\; y' = y, \quad y(0) = 1,$ $\qquad (b)\;\; y' = 100y, \quad y(0) = 1,$

$(c)\;\; y' = -y, \quad y(0) = 1,$ $\qquad (d)\;\; y' = -100y, \quad y(0) = 1,$

$(e)\;\; y' = -y + \sin(x), \quad y(0) = 1,$ $\qquad (f)\;\; y' = -100y + \sin(x), \quad y(0) = 1,$

$(g)\;\; y' = My + \phi(x),$ Large negative $\lambda \in \sigma(M)$ $\qquad (h)\;\; y' = M(x)y + \phi(x),$

$(i)\;\; y' = f(y), \quad z' = (\frac{\partial f}{\partial y})z,$ $\qquad (j)\;\; y' = f(x, y)$

Learning through examples

Let's look at some initial value problems and see which are stiff

(a) $y' = y, \quad y(0) = 1,$

(b) $y' = 100y, \quad y(0) = 1,$

(c) $y' = -y, \quad y(0) = 1,$

(d) $y' = -100y, \quad y(0) = 1,$

(e) $y' = -y + \sin(x), \quad y(0) = 1,$

(f) $y' = -100y + \sin(x), \quad y(0) = 1,$

(g) $y' = My + \phi(x),$
Large negative $\lambda \in \sigma(M)$

(h) $y' = M(x)y + \phi(x),$

(i) $y' = f(y),$
$z' = (\frac{\partial f}{\partial y})z,$

(j) $y' = f(x, y)$

Learning through examples

Let's look at some initial value problems and see which are stiff

$(a)\ \ y' = y, \qquad y(0) = 1,$

$(b)\ \ y' = 100y, \qquad y(0) = 1,$

$(c)\ \ y' = -y, \qquad y(0) = 1,$

$(d)\ \ y' = -100y, \qquad y(0) = 1,$

$(e)\ \ y' = -y + \sin(x), \qquad y(0) = 1,$

$(f)\ \ y' = -100y + \sin(x), \qquad y(0) = 1,$

$(g)\ \ y' = My + \phi(x),$
Large negative $\lambda \in \sigma(M)$

$(h)\ \ y' = M(x)y + \phi(x),$

$(i)\ \ y' = f(y),$
$z' = (\frac{\partial f}{\partial y})z,$

$(j)\ \ y' = f(x, y)$

Learning through examples

Let's look at some initial value problems and see which are stiff

(a) $y' = y, \quad y(0) = 1,$

(b) $y' = 100y, \quad y(0) = 1,$

(c) $y' = -y, \quad y(0) = 1,$

(d) $y' = -100y, \quad y(0) = 1,$

(e) $y' = -y + \sin(x), \quad y(0) = 1,$

(f) $y' = -100y + \sin(x), \quad y(0) = 1,$

(g) $y' = My + \phi(x),$
Large negative $\lambda \in \sigma(M)$

(h) $y' = M(x)y + \phi(x),$

(i) $y' = f(y),$
$z' = (\frac{\partial f}{\partial y})z,$

(j) $y' = f(x, y)$

Learning through examples

Let's look at some initial value problems and see which are stiff

(a) $y' = y, \quad y(0) = 1,$

(b) $y' = 100y, \quad y(0) = 1,$

(c) $y' = -y, \quad y(0) = 1,$

(d) $y' = -100y, \quad y(0) = 1,$

(e) $y' = -y + \sin(x), \quad y(0) = 1,$

(f) $y' = -100y + \sin(x), \quad y(0) = 1,$

(g) $y' = My + \phi(x),$
Large negative $\lambda \in \sigma(M)$

(h) $y' = M(x)y + \phi(x),$

(i) $y' = f(y),$
$z' = \left(\frac{\partial f}{\partial y}\right) z,$

(j) $y' = f(x, y)$

Learning through examples

Let's look at some initial value problems and see which are stiff

(a) $y' = y, \quad y(0) = 1,$

(b) $y' = 100y, \quad y(0) = 1,$

(c) $y' = -y, \quad y(0) = 1,$

(d) $y' = -100y, \quad y(0) = 1,$

(e) $y' = -y + \sin(x), \quad y(0) = 1,$

(f) $y' = -100y + \sin(x), \quad y(0) = 1,$

(g) $y' = My + \phi(x),$
Large negative $\lambda \in \sigma(M)$

(h) $y' = M(x)y + \phi(x),$

(i) $y' = f(y),$
$z' = (\frac{\partial f}{\partial y})z,$

(j) $y' = f(x, y)$

Learning through examples

Let's look at some initial value problems and see which are stiff

(a) $y' = y,$ $y(0) = 1,$

(b) $y' = 100y,$ $y(0) = 1,$

(c) $y' = -y,$ $y(0) = 1,$

(d) $y' = -100y,$ $y(0) = 1,$

(e) $y' = -y + \sin(x),$ $y(0) = 1,$

(f) $y' = -100y + \sin(x),$ $y(0) = 1,$

(g) $y' = My + \phi(x),$
Large negative $\lambda \in \sigma(M)$

(h) $y' = M(x)y + \phi(x),$

(i) $y' = f(y),$
$z' = (\frac{\partial f}{\partial y})z,$

(j) $y' = f(x, y)$

WHAT METHODS ARE USEFUL FOR STIFF PROBLEMS?

The design of stiff solvers is a search for compromises.

Some of the things we want, if we can get them, are

- High order
- High stage order
- Good stability
- Low cost
- Adaptability

We will talk about which of these features are readily available for known method classes.

The design of stiff solvers is a search for compromises. Some of the things we want, if we can get them, are

- High order
- High stage order
- Good stability
- Low cost
- Adaptability

We will talk about which of these features are readily available for known method classes.

The design of stiff solvers is a search for compromises.
Some of the things we want, if we can get them, are

- High order
- High stage order
- Good stability
- Low cost
- Adaptability

We will talk about which of these features are readily available for known method classes.

The design of stiff solvers is a search for compromises.
Some of the things we want, if we can get them, are

- High order
- High stage order
- Good stability
- Low cost
- Adaptability

We will talk about which of these features are readily available for known method classes.

The design of stiff solvers is a search for compromises.
Some of the things we want, if we can get them, are

- High order
- High stage order
- Good stability
- Low cost
- Adaptability

We will talk about which of these features are readily available for known method classes.

The design of stiff solvers is a search for compromises.
Some of the things we want, if we can get them, are

- High order
- High stage order
- Good stability
- Low cost
- Adaptability

We will talk about which of these features are readily available for known method classes.

The design of stiff solvers is a search for compromises.
Some of the things we want, if we can get them, are

- High order
- High stage order
- Good stability
- Low cost
- Adaptability

We will talk about which of these features are readily available for known method classes.

Are Runge–Kutta methods useful?

A good starting point is explicit methods with a long interval of stability, at the expense of high order.

For a fixed p it seems to be possible to construct methods with a stability interval

$$[-a, 0]$$

where a grows something like s^2.

The usefulness of these methods is restricted to problems for which the eigenvalues are clustered around the real axis.

The next point to explore is methods that are implicit in the most simple way possible.

These have been variously called semi-implicit, semi-explicit, diagonally-implicit and singly-diagonally-implicit.

Are Runge–Kutta methods useful?

A good starting point is explicit methods with a long interval of stability, at the expense of high order.

For a fixed p it seems to be possible to construct methods with a stability interval

$$[-a, 0]$$

where a grows something like s^2.

The usefulness of these methods is restricted to problems for which the eigenvalues are clustered around the real axis.

The next point to explore is methods that are implicit in the most simple way possible.

These have been variously called semi-implicit, semi-explicit, diagonally-implicit and singly-diagonally-implicit.

Are Runge–Kutta methods useful?

A good starting point is explicit methods with a long interval of stability, at the expense of high order.

For a fixed p it seems to be possible to construct methods with a stability interval

$$[-a, 0]$$

where a grows something like s^2.

The usefulness of these methods is restricted to problems for which the eigenvalues are clustered around the real axis.

The next point to explore is methods that are implicit in the most simple way possible.

These have been variously called semi-implicit, semi-explicit, diagonally-implicit and singly-diagonally-implicit.

Are Runge–Kutta methods useful?

A good starting point is explicit methods with a long interval of stability, at the expense of high order.

For a fixed p it seems to be possible to construct methods with a stability interval

$$[-a, 0]$$

where a grows something like s^2.

The usefulness of these methods is restricted to problems for which the eigenvalues are clustered around the real axis.

The next point to explore is methods that are implicit in the most simple way possible.

These have been variously called semi-implicit, semi-explicit, diagonally-implicit and singly-diagonally-implicit.

Are Runge–Kutta methods useful?

A good starting point is explicit methods with a long interval of stability, at the expense of high order.

For a fixed p it seems to be possible to construct methods with a stability interval

$$[-a, 0]$$

where a grows something like s^2.

The usefulness of these methods is restricted to problems for which the eigenvalues are clustered around the real axis.

The next point to explore is methods that are implicit in the most simple way possible.

These have been variously called semi-implicit, semi-explicit, diagonally-implicit and singly-diagonally-implicit.

The tableau for a semi-implicit method looks like this

$$
\begin{array}{c|ccccc}
\gamma & \gamma & 0 & 0 & \cdots & 0 \\
c_2 & a_{21} & \gamma & 0 & \cdots & 0 \\
c_3 & a_{31} & a_{32} & \gamma & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & & \vdots \\
c_s & a_{s1} & a_{s2} & a_{s3} & \cdots & \gamma \\
\hline
 & b_1 & b_2 & b_3 & \cdots & b_s
\end{array}
$$

Notice that

$$\sigma(A) = \{\gamma\}$$

and this is where the "singly (S)" in "SDIRK" comes from. Semi-implicit methods have a limited role in stiff integration because, for many problems they suffer from "order-reduction" because the stage-order is never more than 2.

The tableau for a semi-implicit method looks like this

$$
\begin{array}{c|cccccc}
\gamma & \gamma & 0 & 0 & \cdots & 0 \\
c_2 & a_{21} & \gamma & 0 & \cdots & 0 \\
c_3 & a_{31} & a_{32} & \gamma & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & & \vdots \\
c_s & a_{s1} & a_{s2} & a_{s3} & \cdots & \gamma \\
\hline
 & b_1 & b_2 & b_3 & \cdots & b_s
\end{array}
$$

Notice that

$$\sigma(A) = \{\gamma\}$$

and this is where the "singly (S)" in "SDIRK" comes from.

Semi-implicit methods have a limited role in stiff integration because, for many problems they suffer from "order-reduction" because the stage-order is never more than 2.

The tableau for a semi-implicit method looks like this

$$\begin{array}{c|ccccc} \gamma & \gamma & 0 & 0 & \cdots & 0 \\ c_2 & a_{21} & \gamma & 0 & \cdots & 0 \\ c_3 & a_{31} & a_{32} & \gamma & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ c_s & a_{s1} & a_{s2} & a_{s3} & \cdots & \gamma \\ \hline & b_1 & b_2 & b_3 & \cdots & b_s \end{array}$$

Notice that

$$\sigma(A) = \{\gamma\}$$

and this is where the “singly (S)” in “SDIRK” comes from. Semi-implicit methods have a limited role in stiff integration because, for many problems they suffer from “order-reduction” because the stage-order is never more than 2.

What about "singly-implicit methods" characterised only by the one-point spectrum property?

Because A can now be a full matrix, the methods might seem to have no particular advantage but we will see that they can be implemented for a similar cost to semi-implicit methods.

Furthermore, it is possible to have stage order s and this makes them potentially quite useful for solving stiff problems.

We will see how this can be done by choosing the abscissa vector c as follows:

$$c_i = \gamma \xi_i, \qquad L_s(\xi_i) = 0, \qquad i = 1, 2, \ldots, s,$$

where L_s is the Laguerre polynomial of degree s.

What about “singly-implicit methods” characterised only by the one-point spectrum property?

Because A can now be a full matrix, the methods might seem to have no particular advantage but we will see that they can be implemented for a similar cost to semi-implicit methods.

Furthermore, it is possible to have stage order s and this makes them potentially quite useful for solving stiff problems.

We will see how this can be done by choosing the abscissa vector c as follows:

$$c_i = \gamma \xi_i, \qquad L_s(\xi_i) = 0, \qquad i = 1, 2, \dots, s,$$

where L_s is the Laguerre polynomial of degree s.

What about "singly-implicit methods" characterised only by the one-point spectrum property?

Because A can now be a full matrix, the methods might seem to have no particular advantage but we will see that they can be implemented for a similar cost to semi-implicit methods.

Furthermore, it is possible to have stage order s and this makes them potentially quite useful for solving stiff problems.

We will see how this can be done by choosing the abscissa vector c as follows:

$$c_i = \gamma \xi_i, \qquad L_s(\xi_i) = 0, \qquad i = 1, 2, \ldots, s,$$

where L_s is the Laguerre polynomial of degree s.

What about "singly-implicit methods" characterised only by the one-point spectrum property?

Because A can now be a full matrix, the methods might seem to have no particular advantage but we will see that they can be implemented for a similar cost to semi-implicit methods.

Furthermore, it is possible to have stage order s and this makes them potentially quite useful for solving stiff problems.

We will see how this can be done by choosing the abscissa vector c as follows:

$$c_i = \gamma \xi_i, \qquad L_s(\xi_i) = 0, \qquad i = 1, 2, \ldots, s,$$

where L_s is the Laguerre polynomial of degree s.

What about "singly-implicit methods" characterised only by the one-point spectrum property?

Because A can now be a full matrix, the methods might seem to have no particular advantage but we will see that they can be implemented for a similar cost to semi-implicit methods.

Furthermore, it is possible to have stage order s and this makes them potentially quite useful for solving stiff problems.

We will see how this can be done by choosing the abscissa vector c as follows:

$$c_i = \gamma \xi_i, \qquad L_s(\xi_i) = 0, \qquad i = 1, 2, \ldots, s,$$

where L_s is the Laguerre polynomial of degree s.

If $\mathbf{1}, c, c^2, \dots, c^s$ are the component-wise powers of c, then the stage order conditions are

$$Ac^{k-1} = \tfrac{1}{k}c^k, \qquad k = 1, 2, \dots, s.$$

Substitute $c = \gamma\xi$ and we have

$$A\xi^{k-1} = \gamma\tfrac{1}{k}\xi^k, \qquad k = 1, 2, \dots, s,$$

or, taking linear combinations,

$$A\varphi(\xi) = \gamma\int_0^\xi \varphi(x)dx, \qquad \deg(\varphi) \le s - 1.$$

For example, if we use the Laguerre polynomials,

$$AL_{k-1}(\xi) = L_{k-1}(\xi) - L_k(\xi), \qquad k = 1, 2, \dots, s.$$

If $k = s$, the last term is missing, because $L_k(\xi_i) = 0$, $i = 1, 2, \dots, s$.

If $\mathbf{1}, c, c^2, \ldots, c^s$ are the component-wise powers of c, then the stage order conditions are

$$Ac^{k-1} = \tfrac{1}{k}c^k, \qquad k = 1, 2, \ldots, s.$$

Substitute $c = \gamma\xi$ and we have

$$A\xi^{k-1} = \gamma\tfrac{1}{k}\xi^k, \qquad k = 1, 2, \ldots, s,$$

or, taking linear combinations,

$$A\varphi(\xi) = \gamma \int_0^\xi \varphi(x)dx, \qquad \deg(\varphi) \le s-1.$$

For example, if we use the Laguerre polynomials,

$$AL_{k-1}(\xi) = L_{k-1}(\xi) - L_k(\xi), \qquad k = 1, 2, \ldots, s.$$

If $k = s$, the last term is missing, because $L_k(\xi_i) = 0$, $i = 1, 2, \ldots, s$.

If $\mathbf{1}, c, c^2, \ldots, c^s$ are the component-wise powers of c, then the stage order conditions are

$$Ac^{k-1} = \tfrac{1}{k}c^k, \qquad k = 1, 2, \ldots, s.$$

Substitute $c = \gamma\xi$ and we have

$$A\xi^{k-1} = \gamma\tfrac{1}{k}\xi^k, \qquad k = 1, 2, \ldots, s,$$

or, taking linear combinations,

$$A\varphi(\xi) = \gamma\int_0^{\xi}\varphi(x)dx, \qquad \deg(\varphi) \leq s-1.$$

For example, if we use the Laguerre polynomials,

$$AL_{k-1}(\xi) = L_{k-1}(\xi) - L_k(\xi), \qquad k = 1, 2, \ldots, s.$$

If $k = s$, the last term is missing, because $L_k(\xi_i) = 0$, $i = 1, 2, \ldots, s$.

If $\mathbf{1}, c, c^2, \ldots, c^s$ are the component-wise powers of c, then the stage order conditions are

$$Ac^{k-1} = \tfrac{1}{k}c^k, \qquad k = 1, 2, \ldots, s.$$

Substitute $c = \gamma\xi$ and we have

$$A\xi^{k-1} = \gamma\tfrac{1}{k}\xi^k, \qquad k = 1, 2, \ldots, s,$$

or, taking linear combinations,

$$A\varphi(\xi) = \gamma\int_0^\xi \varphi(x)dx, \qquad \deg(\varphi) \le s-1.$$

For example, if we use the Laguerre polynomials,

$$AL_{k-1}(\xi) = L_{k-1}(\xi) - L_k(\xi), \qquad k = 1, 2, \ldots, s.$$

If $k = s$, the last term is missing, because $L_k(\xi_i) = 0$, $i = 1, 2, \ldots, s$.

If $\mathbf{1}, c, c^2, \dots, c^s$ are the component-wise powers of c, then the stage order conditions are

$$Ac^{k-1} = \tfrac{1}{k}c^k, \qquad k = 1, 2, \dots, s.$$

Substitute $c = \gamma\xi$ and we have

$$A\xi^{k-1} = \gamma\tfrac{1}{k}\xi^k, \qquad k = 1, 2, \dots, s,$$

or, taking linear combinations,

$$A\varphi(\xi) = \gamma \int_0^\xi \varphi(x)dx, \qquad \deg(\varphi) \le s - 1.$$

For example, if we use the Laguerre polynomials,

$$AL_{k-1}(\xi) = L_{k-1}(\xi) - L_k(\xi), \qquad k = 1, 2, \dots, s.$$

If $k = s$, the last term is missing, because $L_k(\xi_i) = 0$, $i = 1, 2, \dots, s$.

Define M by

$$M = \left[\begin{array}{cccc} L_0(\xi) & L_1(\xi) & \cdots & L_{s-1}(\xi) \end{array} \right]$$

then A can be written as

$$A = M \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ -1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & 0 & \cdots & -1 & 1 \end{bmatrix} M^{-1}.$$

It is possible to incorporate the similarity transformation M into the implementation scheme to lower the cost for large problems.

Define M by

$$M = \begin{bmatrix} L_0(\xi) & L_1(\xi) & \cdots & L_{s-1}(\xi) \end{bmatrix}$$

then A can be written as

$$A = M \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ -1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & 0 & \cdots -1 & & 1 \end{bmatrix} M^{-1}.$$

It is possible to incorporate the similarity transformation M into the implementation scheme to lower the cost for large problems.

Define M by

$$M = \begin{bmatrix} L_0(\xi) & L_1(\xi) & \cdots & L_{s-1}(\xi) \end{bmatrix}$$

then A can be written as

$$A = M \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ -1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & 0 & \cdots & -1 & 1 \end{bmatrix} M^{-1}.$$

It is possible to incorporate the similarity transformation M into the implementation scheme to lower the cost for large problems.

Are linear multistep methods useful?

Theorem (Dahlquist Second Barrier)

A linear multistep method with order greater than 2 is not A-stable

Are linear multistep methods useful?

Theorem (Dahlquist Second Barrier)

A linear multistep method with order greater than 2 is not A-stable

How about the following corollary?

Assertion

A linear multistep method with order greater than 2 is not useful

Are linear multistep methods useful?

Theorem (Dahlquist Second Barrier)

A linear multistep method with order greater than 2 is not A-stable

How about the following corollary?

Assertion

A linear multistep method with order greater than 2 is not useful

Nothing could be further from the truth!

Are linear multistep methods useful?

Theorem (Dahlquist Second Barrier)

A linear multistep method with order greater than 2 is not A-stable

How about the following corollary?

~~**Assertion**~~

~~*A linear multistep method with order greater than 2 is not useful*~~

Nothing could be further from the truth!

Are linear multistep methods useful?

Theorem (Dahlquist Second Barrier)

A linear multistep method with order greater than 2 is not A-stable

How about the following corollary?

Assertion

~~*A linear multistep method with order greater than 2 is not useful*~~

Nothing could be further from the truth!
Linear multistep methods (BDF methods) are used every moment of every day to solve stiff problems.

Are linear multistep methods useful?

Theorem (Dahlquist Second Barrier)

A linear multistep method with order greater than 2 is not A-stable

How about the following corollary?

~~**Assertion**~~

~~*A linear multistep method with order greater than 2 is not useful*~~

Nothing could be further from the truth!
Linear multistep methods (BDF methods) are used every moment of every day to solve stiff problems.
What they lack in terms of A-stability and accuracy, they make up for in terms of speed and efficiency.

Are general linear methods useful?

I believe they will emerge as very useful methods, once further work has been done on their implementation.

From a large choice of possible methods, I am selecting the following method with order 3.

The coefficients are presented as a partitioned tableau

$$\left[\begin{array}{cc} A & U \\ B & V \end{array}\right].$$

This means that, with input vector $y^{[0]}$, the stages Y, the stage derivatives F, and the output vector $y^{[1]}$, are given by

$$Y = hAF + Uy^{[0]},$$
$$y^{[1]} = hBF + Vy^{[0]}.$$

Are general linear methods useful?

I believe they will emerge as very useful methods, once further work has been done on their implementation.

From a large choice of possible methods, I am selecting the following method with order 3.

The coefficients are presented as a partitioned tableau

$$\begin{bmatrix} A & U \\ B & V \end{bmatrix}.$$

This means that, with input vector $y^{[0]}$, the stages Y, the stage derivatives F, and the output vector $y^{[1]}$, are given by

$$Y = hAF + Uy^{[0]},$$
$$y^{[1]} = hBF + Vy^{[0]}.$$

Are general linear methods useful?

I believe they will emerge as very useful methods, once further work has been done on their implementation.

From a large choice of possible methods, I am selecting the following method with order 3.

The coefficients are presented as a partitioned tableau

$$\begin{bmatrix} A & U \\ B & V \end{bmatrix}.$$

This means that, with input vector $y^{[0]}$, the stages Y, the stage derivatives F, and the output vector $y^{[1]}$, are given by

$$Y = hAF + Uy^{[0]},$$
$$y^{[1]} = hBF + Vy^{[0]}.$$

Are general linear methods useful?

I believe they will emerge as very useful methods, once further work has been done on their implementation.

From a large choice of possible methods, I am selecting the following method with order 3.

The coefficients are presented as a partitioned tableau

$$\left[\begin{array}{cc} A & U \\ B & V \end{array}\right].$$

This means that, with input vector $y^{[0]}$, the stages Y, the stage derivatives F, and the output vector $y^{[1]}$, are given by

$$Y = hAF + Uy^{[0]},$$
$$y^{[1]} = hBF + Vy^{[0]}.$$

$$\left[\begin{array}{cc} A & U \\ B & V \end{array}\right] = \left[\begin{array}{cccc|cccc}
\frac{1}{4} & 0 & 0 & 0 & 1 & 0 & -\frac{1}{32} & -\frac{1}{192} \\
\frac{11}{2124} & \frac{1}{4} & 0 & 0 & 1 & \frac{130}{531} & -\frac{11}{8496} & -\frac{719}{67968} \\
\frac{117761}{23364} & -\frac{189}{44} & \frac{1}{4} & 0 & 1 & -\frac{130}{531} & \frac{183437}{186912} & \frac{283675}{747648} \\
\frac{312449}{23364} & -\frac{4525}{396} & \frac{1}{36} & \frac{1}{4} & 1 & -\frac{650}{531} & \frac{121459}{46728} & \frac{130127}{124608} \\
\hline
-\frac{58405}{7788} & \frac{4297}{132} & -\frac{475}{12} & 15 & 1 & \frac{125}{236} & \frac{510}{649} & -\frac{733}{20768} \\
-\frac{64}{33} & \frac{746}{33} & -\frac{95}{3} & 12 & 0 & 0 & \frac{85}{44} & \frac{677}{1056} \\
-\frac{8}{3} & \frac{4}{3} & \frac{4}{3} & 0 & 0 & 0 & 0 & \frac{13}{24} \\
-32 & 112 & -128 & 48 & 0 & 0 & 0 & 0
\end{array}\right]$$

- $\longrightarrow$ $c^T = \left[\begin{array}{cccc} \frac{1}{4} & \frac{1}{2} & \frac{3}{4} & 1 \end{array}\right]$
- $\longrightarrow$ A diagonally-implicit
- $\longrightarrow$ Stage order 3
- $\longrightarrow$ Stability function $w^4(1-\frac{1}{4}z)^4 - w^3(1-\frac{1}{8}z^2-\frac{1}{48}z^3)$

$$c.f.\ R(z) = \frac{1-\frac{1}{8}z^2-\frac{1}{48}z^3}{(1-\frac{1}{4}z)^4}$$

- $\longrightarrow$ Simple step-size change mechanism based on Nordsieck representation of data
- $\longrightarrow$ Cheap error estimation and neighbouring method estimation

$$\left[\begin{array}{cc} A & U \\ B & V \end{array}\right] = \left[\begin{array}{cccc|cccc}
\frac{1}{4} & 0 & 0 & 0 & 1 & 0 & -\frac{1}{32} & -\frac{1}{192} \\
\frac{11}{2124} & \frac{1}{4} & 0 & 0 & 1 & \frac{130}{531} & -\frac{11}{8496} & -\frac{719}{67968} \\
\frac{117761}{23364} & -\frac{189}{44} & \frac{1}{4} & 0 & 1 & -\frac{130}{531} & \frac{183437}{186912} & \frac{283675}{747648} \\
\frac{312449}{23364} & -\frac{4525}{396} & \frac{1}{36} & \frac{1}{4} & 1 & -\frac{650}{531} & \frac{121459}{46728} & \frac{130127}{124608} \\
\hline
-\frac{58405}{7788} & \frac{4297}{132} & -\frac{475}{12} & 15 & 1 & \frac{125}{236} & \frac{510}{649} & -\frac{733}{20768} \\
-\frac{64}{33} & \frac{746}{33} & -\frac{95}{3} & 12 & 0 & 0 & \frac{85}{44} & \frac{677}{1056} \\
-\frac{8}{3} & \frac{4}{3} & \frac{4}{3} & 0 & 0 & 0 & 0 & \frac{13}{24} \\
-32 & 112 & -128 & 48 & 0 & 0 & 0 & 0
\end{array}\right]$$

- $\longrightarrow$ $c^T = \left[\begin{array}{cccc} \frac{1}{4} & \frac{1}{2} & \frac{3}{4} & 1 \end{array}\right]$
- $\longrightarrow$ A diagonally-implicit
- $\longrightarrow$ Stage order 3
- $\longrightarrow$ Stability function $w^4(1-\frac{1}{4}z)^4 - w^3(1-\frac{1}{8}z^2-\frac{1}{48}z^3)$

$$c.f.\ R(z) = \frac{1-\frac{1}{8}z^2-\frac{1}{48}z^3}{(1-\frac{1}{4}z)^4}$$

- $\longrightarrow$ Simple step-size change mechanism based on Nordsieck representation of data
- $\longrightarrow$ Cheap error estimation and neighbouring method estimation

$$\left[\begin{array}{cc} A & U \\ B & V \end{array}\right] = \left[\begin{array}{cccc|cccc}
\frac{1}{4} & 0 & 0 & 0 & 1 & 0 & -\frac{1}{32} & -\frac{1}{192} \\
\frac{11}{2124} & \frac{1}{4} & 0 & 0 & 1 & \frac{130}{531} & -\frac{11}{8496} & -\frac{719}{67968} \\
\frac{117761}{23364} & -\frac{189}{44} & \frac{1}{4} & 0 & 1 & -\frac{130}{531} & \frac{183437}{186912} & \frac{283675}{747648} \\
\frac{312449}{23364} & -\frac{4525}{396} & \frac{1}{36} & \frac{1}{4} & 1 & -\frac{650}{531} & \frac{121459}{46728} & \frac{130127}{124608} \\
\hline
-\frac{58405}{7788} & \frac{4297}{132} & -\frac{475}{12} & 15 & 1 & \frac{125}{236} & \frac{510}{649} & -\frac{733}{20768} \\
-\frac{64}{33} & \frac{746}{33} & -\frac{95}{3} & 12 & 0 & 0 & \frac{85}{44} & \frac{677}{1056} \\
-\frac{8}{3} & \frac{4}{3} & \frac{4}{3} & 0 & 0 & 0 & 0 & \frac{13}{24} \\
-32 & 112 & -128 & 48 & 0 & 0 & 0 & 0
\end{array}\right]$$

- $\longrightarrow$ $c^T = \left[\begin{array}{cccc} \frac{1}{4} & \frac{1}{2} & \frac{3}{4} & 1 \end{array}\right]$
- $\longrightarrow$ A diagonally-implicit
- $\longrightarrow$ Stage order 3
- $\longrightarrow$ Stability function $w^4(1-\frac{1}{4}z)^4 - w^3(1-\frac{1}{8}z^2-\frac{1}{48}z^3)$

$$c.f.\ R(z) = \frac{1-\frac{1}{8}z^2-\frac{1}{48}z^3}{(1-\frac{1}{4}z)^4}$$

- $\longrightarrow$ Simple step-size change mechanism based on Nordsieck representation of data
- $\longrightarrow$ Cheap error estimation and neighbouring method estimation

$$\left[\begin{array}{cc} A & U \\ B & V \end{array}\right] = \left[\begin{array}{cccc|cccc}
\frac{1}{4} & 0 & 0 & 0 & 1 & 0 & -\frac{1}{32} & -\frac{1}{192} \\
\frac{11}{2124} & \frac{1}{4} & 0 & 0 & 1 & \frac{130}{531} & -\frac{11}{8496} & -\frac{719}{67968} \\
\frac{117761}{23364} & -\frac{189}{44} & \frac{1}{4} & 0 & 1 & -\frac{130}{531} & \frac{183437}{186912} & \frac{283675}{747648} \\
\frac{312449}{23364} & -\frac{4525}{396} & \frac{1}{36} & \frac{1}{4} & 1 & -\frac{650}{531} & \frac{121459}{46728} & \frac{130127}{124608} \\
\hline
-\frac{58405}{7788} & \frac{4297}{132} & -\frac{475}{12} & 15 & 1 & \frac{125}{236} & \frac{510}{649} & -\frac{733}{20768} \\
-\frac{64}{33} & \frac{746}{33} & -\frac{95}{3} & 12 & 0 & 0 & \frac{85}{44} & \frac{677}{1056} \\
-\frac{8}{3} & \frac{4}{3} & \frac{4}{3} & 0 & 0 & 0 & 0 & \frac{13}{24} \\
-32 & 112 & -128 & 48 & 0 & 0 & 0 & 0
\end{array}\right]$$

- $\longrightarrow$ $c^T = \left[\begin{array}{cccc} \frac{1}{4} & \frac{1}{2} & \frac{3}{4} & 1 \end{array}\right]$
- $\longrightarrow$ A diagonally-implicit
- $\longrightarrow$ Stage order 3
- $\longrightarrow$ Stability function $w^4(1-\frac{1}{4}z)^4 - w^3(1-\frac{1}{8}z^2-\frac{1}{48}z^3)$

$$c.f.\ R(z) = \frac{1-\frac{1}{8}z^2-\frac{1}{48}z^3}{(1-\frac{1}{4}z)^4}$$

- $\longrightarrow$ Simple step-size change mechanism based on Nordsieck representation of data
- $\longrightarrow$ Cheap error estimation and neighbouring method estimation

$$\left[\begin{array}{cc} A & U \\ B & V \end{array}\right] = \left[\begin{array}{cccc|cccc}
\frac{1}{4} & 0 & 0 & 0 & 1 & 0 & -\frac{1}{32} & -\frac{1}{192} \\
\frac{11}{2124} & \frac{1}{4} & 0 & 0 & 1 & \frac{130}{531} & -\frac{11}{8496} & -\frac{719}{67968} \\
\frac{117761}{23364} & -\frac{189}{44} & \frac{1}{4} & 0 & 1 & -\frac{130}{531} & \frac{183437}{186912} & \frac{283675}{747648} \\
\frac{312449}{23364} & -\frac{4525}{396} & \frac{1}{36} & \frac{1}{4} & 1 & -\frac{650}{531} & \frac{121459}{46728} & \frac{130127}{124608} \\
\hline
-\frac{58405}{7788} & \frac{4297}{132} & -\frac{475}{12} & 15 & 1 & \frac{125}{236} & \frac{510}{649} & -\frac{733}{20768} \\
-\frac{64}{33} & \frac{746}{33} & -\frac{95}{3} & 12 & 0 & 0 & \frac{85}{44} & \frac{677}{1056} \\
-\frac{8}{3} & \frac{4}{3} & \frac{4}{3} & 0 & 0 & 0 & 0 & \frac{13}{24} \\
-32 & 112 & -128 & 48 & 0 & 0 & 0 & 0
\end{array}\right]$$

- ⟶ $c^T = \left[\begin{array}{cccc} \frac{1}{4} & \frac{1}{2} & \frac{3}{4} & 1 \end{array}\right]$
- ⟶ A diagonally-implicit
- ⟶ Stage order 3
- ⟶ Stability function $w^4(1-\frac{1}{4}z)^4 - w^3(1-\frac{1}{8}z^2-\frac{1}{48}z^3)$

$$\text{c.f. } R(z) = \frac{1-\frac{1}{8}z^2-\frac{1}{48}z^3}{(1-\frac{1}{4}z)^4}$$

- ⟶ Simple step-size change mechanism based on Nordsieck representation of data
- ⟶ Cheap error estimation and neighbouring method estimation

$$
\left[\begin{array}{cc} A & U \\ B & V \end{array}\right] = \left[\begin{array}{cccc|cccc}
\frac{1}{4} & 0 & 0 & 0 & 1 & 0 & -\frac{1}{32} & -\frac{1}{192} \\
\frac{11}{2124} & \frac{1}{4} & 0 & 0 & 1 & \frac{130}{531} & -\frac{11}{8496} & -\frac{719}{67968} \\
\frac{117761}{23364} & -\frac{189}{44} & \frac{1}{4} & 0 & 1 & -\frac{130}{531} & \frac{183437}{186912} & \frac{283675}{747648} \\
\frac{312449}{23364} & -\frac{4525}{396} & \frac{1}{36} & \frac{1}{4} & 1 & -\frac{650}{531} & \frac{121459}{46728} & \frac{130127}{124608} \\
\hline
-\frac{58405}{7788} & \frac{4297}{132} & -\frac{475}{12} & 15 & 1 & \frac{125}{236} & \frac{510}{649} & -\frac{733}{20768} \\
-\frac{64}{33} & \frac{746}{33} & -\frac{95}{3} & 12 & 0 & 0 & \frac{85}{44} & \frac{677}{1056} \\
-\frac{8}{3} & \frac{4}{3} & \frac{4}{3} & 0 & 0 & 0 & 0 & \frac{13}{24} \\
-32 & 112 & -128 & 48 & 0 & 0 & 0 & 0
\end{array}\right]
$$

- $\longrightarrow$ $c^T = \left[\begin{array}{cccc} \frac{1}{4} & \frac{1}{2} & \frac{3}{4} & 1 \end{array}\right]$
- $\longrightarrow$ A diagonally-implicit
- $\longrightarrow$ Stage order 3
- $\longrightarrow$ Stability function $w^4(1-\frac{1}{4}z)^4 - w^3(1-\frac{1}{8}z^2-\frac{1}{48}z^3)$

$$c.f.\ \ R(z) = \frac{1-\frac{1}{8}z^2-\frac{1}{48}z^3}{(1-\frac{1}{4}z)^4}$$

- $\longrightarrow$ Simple step-size change mechanism based on Nordsieck representation of data
- $\longrightarrow$ Cheap error estimation and neighbouring method estimation

$$\left[\begin{array}{cc} A & U \\ B & V \end{array}\right] = \left[\begin{array}{cccc|cccc}
\frac{1}{4} & 0 & 0 & 0 & 1 & 0 & -\frac{1}{32} & -\frac{1}{192} \\
\frac{11}{2124} & \frac{1}{4} & 0 & 0 & 1 & \frac{130}{531} & -\frac{11}{8496} & -\frac{719}{67968} \\
\frac{117761}{23364} & -\frac{189}{44} & \frac{1}{4} & 0 & 1 & -\frac{130}{531} & \frac{183437}{186912} & \frac{283675}{747648} \\
\frac{312449}{23364} & -\frac{4525}{396} & \frac{1}{36} & \frac{1}{4} & 1 & -\frac{650}{531} & \frac{121459}{46728} & \frac{130127}{124608} \\
\hline
-\frac{58405}{7788} & \frac{4297}{132} & -\frac{475}{12} & 15 & 1 & \frac{125}{236} & \frac{510}{649} & -\frac{733}{20768} \\
-\frac{64}{33} & \frac{746}{33} & -\frac{95}{3} & 12 & 0 & 0 & \frac{85}{44} & \frac{677}{1056} \\
-\frac{8}{3} & \frac{4}{3} & \frac{4}{3} & 0 & 0 & 0 & 0 & \frac{13}{24} \\
-32 & 112 & -128 & 48 & 0 & 0 & 0 & 0
\end{array}\right]$$

- $\longrightarrow$ $c^T = \left[\begin{array}{cccc} \frac{1}{4} & \frac{1}{2} & \frac{3}{4} & 1 \end{array}\right]$
- $\longrightarrow$ A diagonally-implicit
- $\longrightarrow$ Stage order 3
- $\longrightarrow$ Stability function $w^4(1-\frac{1}{4}z)^4 - w^3(1-\frac{1}{8}z^2-\frac{1}{48}z^3)$

$$c.f.\ \ R(z) = \frac{1-\frac{1}{8}z^2-\frac{1}{48}z^3}{(1-\frac{1}{4}z)^4}$$

- $\longrightarrow$ Simple step-size change mechanism based on Nordsieck representation of data
- $\longrightarrow$ Cheap error estimation and neighbouring method estimation

$$\left[\begin{array}{cc} A & U \\ B & V \end{array}\right] = \left[\begin{array}{cccc|cccc}
\frac{1}{4} & 0 & 0 & 0 & 1 & 0 & -\frac{1}{32} & -\frac{1}{192} \\
\frac{11}{2124} & \frac{1}{4} & 0 & 0 & 1 & \frac{130}{531} & -\frac{11}{8496} & -\frac{719}{67968} \\
\frac{117761}{23364} & -\frac{189}{44} & \frac{1}{4} & 0 & 1 & -\frac{130}{531} & \frac{183437}{186912} & \frac{283675}{747648} \\
\frac{312449}{23364} & -\frac{4525}{396} & \frac{1}{36} & \frac{1}{4} & 1 & -\frac{650}{531} & \frac{121459}{46728} & \frac{130127}{124608} \\
\hline
-\frac{58405}{7788} & \frac{4297}{132} & -\frac{475}{12} & 15 & 1 & \frac{125}{236} & \frac{510}{649} & -\frac{733}{20768} \\
-\frac{64}{33} & \frac{746}{33} & -\frac{95}{3} & 12 & 0 & 0 & \frac{85}{44} & \frac{677}{1056} \\
-\frac{8}{3} & \frac{4}{3} & \frac{4}{3} & 0 & 0 & 0 & 0 & \frac{13}{24} \\
-32 & 112 & -128 & 48 & 0 & 0 & 0 & 0
\end{array}\right]$$

⟶ $c^T = \left[\begin{array}{cccc} \frac{1}{4} & \frac{1}{2} & \frac{3}{4} & 1 \end{array}\right]$

⟶ A diagonally-implicit

⟶ Stage order 3

⟶ Stability function $w^4(1-\frac{1}{4}z)^4 - w^3(1-\frac{1}{8}z^2-\frac{1}{48}z^3)$

$$c.f.\ R(z) = \frac{1-\frac{1}{8}z^2-\frac{1}{48}z^3}{(1-\frac{1}{4}z)^4}$$

⟶ Simple step-size change mechanism based on Nordsieck representation of data

⟶ Cheap error estimation and neighbouring method estimation

STABILITY QUESTIONS

The solution to a dynamical system can contain, at least locally, a potent mix of rapidly decaying components, oscillatory components, slowly varying components and rapidly increasing components.

We don't want a numerical process to be controlled by the rapidly decaying components, because we expect these to become negligible after a short time; therefore they will not correspond to anything of physical significance in the long run.

It is a major aim of a stiff solver to look after rapidly decaying components and make sure they don't do any harm.

The solution to a dynamical system can contain, at least locally, a potent mix of rapidly decaying components, oscillatory components, slowly varying components and rapidly increasing components.

We don't want a numerical process to be controlled by the rapidly decaying components, because we expect these to become negligible after a short time; therefore they will not correspond to anything of physical significance in the long run.

It is a major aim of a stiff solver to look after rapidly decaying components and make sure they don't do any harm.

The solution to a dynamical system can contain, at least locally, a potent mix of rapidly decaying components, oscillatory components, slowly varying components and rapidly increasing components.

We don't want a numerical process to be controlled by the rapidly decaying components, because we expect these to become negligible after a short time; therefore they will not correspond to anything of physical significance in the long run.

It is a major aim of a stiff solver to look after rapidly decaying components and make sure they don't do any harm.

A-stability, A(α) stability and L-stability

If we pretend that the various components of a solution combine linearly, or at least close enough to linearly for all practical purposes, then linear stability analysis makes a lot of sense.
We just need to look at how a single component, de-coupled from the rest of the system, behaves under numerical treatment.
This means that we can consider a model linear problem

$$y' = qy,$$

where q is a complex number. If $z = hq$ is such that the approximate solution is stable, we say that z is in the stability region.
If the stability region includes the entire left half-plane then the method is A-stable.

A-stability, A(α) stability and L-stability

If we pretend that the various components of a solution combine linearly, or at least close enough to linearly for all practical purposes, then linear stability analysis makes a lot of sense.
We just need to look at how a single component, de-coupled from the rest of the system, behaves under numerical treatment.
This means that we can consider a model linear problem

$$y' = qy,$$

where q is a complex number. If $z = hq$ is such that the approximate solution is stable, we say that z is in the stability region.
If the stability region includes the entire left half-plane then the method is A-stable.

A-stability, A(α) stability and L-stability

If we pretend that the various components of a solution combine linearly, or at least close enough to linearly for all practical purposes, then linear stability analysis makes a lot of sense. We just need to look at how a single component, de-coupled from the rest of the system, behaves under numerical treatment. This means that we can consider a model linear problem

$$y' = qy,$$

where q is a complex number. If $z = hq$ is such that the approximate solution is stable, we say that z is in the stability region.

If the stability region includes the entire left half-plane then the method is A-stable.

A-stability, A(α) stability and L-stability

If we pretend that the various components of a solution combine linearly, or at least close enough to linearly for all practical purposes, then linear stability analysis makes a lot of sense. We just need to look at how a single component, de-coupled from the rest of the system, behaves under numerical treatment. This means that we can consider a model linear problem

$$y' = qy,$$

where q is a complex number. If $z = hq$ is such that the approximate solution is stable, we say that z is in the stability region.

If the stability region includes the entire left half-plane then the method is A-stable.

A-stability, A(α) stability and L-stability

If we pretend that the various components of a solution combine linearly, or at least close enough to linearly for all practical purposes, then linear stability analysis makes a lot of sense. We just need to look at how a single component, de-coupled from the rest of the system, behaves under numerical treatment. This means that we can consider a model linear problem

$$y' = qy,$$

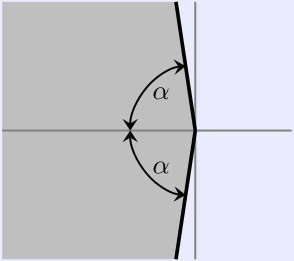
where q is a complex number. If $z = hq$ is such that the approximate solution is stable, we say that z is in the stability region.

If the stability region includes the entire left half-plane then the method is A-stable.

For some problems we want more than A-stability and for some problems we could, and should, be happy with less.

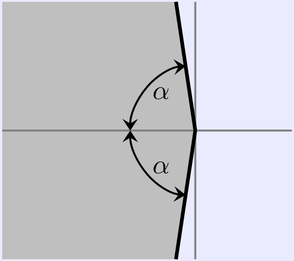
For some problems we want more than A-stability and for some problems we could, and should, be happy with less.

A method is A(α) stable, for a given angle α, if its stability region includes a region like this



For orders greater than 2, linear multistep methods can be A(α) stable for any $\alpha \in [0, \frac{\pi}{2})$.

For some problems we want more than A-stability and for some problems we could, and should, be happy with less.
A method is A(α) stable, for a given angle α, if its stability region includes a region like this



For orders greater than 2, linear multistep methods can be A(α) stable for any $\alpha \in [0, \frac{\pi}{2})$.
As a hope of obtaining high order and accurate linear multistep methods, which are and at the same time "almost A stable", this is an illusion.

Nonlinear stability

In 1975 G. Dahlquist suggested using a test problem satisfying

$$\langle y - z, f(x, y) - f(x, z)\rangle \le 0$$

A differential equation satisfying this condition is dissipative in the sense that

$$\frac{d}{dx}\|y(x) - z(x)\| \le 0.$$

For linear multistep methods GD gave conditions for which the sequence

$$\|y_n - z_n\|$$

followed a similar behaviour.

Shortly afterwards a related theory was developed for Runge–Kutta methods and eventually for general linear methods.

Nonlinear stability

In 1975 G. Dahlquist suggested using a test problem satisfying

$$\langle y - z, f(x, y) - f(x, z)\rangle \le 0$$

A differential equation satisfying this condition is dissipative in the sense that

$$\tfrac{d}{dx}\|y(x) - z(x)\| \le 0.$$

For linear multistep methods GD gave conditions for which the sequence

$$\|y_n - z_n\|$$

followed a similar behaviour.
Shortly afterwards a related theory was developed for Runge–Kutta methods and eventually for general linear methods.

Nonlinear stability

In 1975 G. Dahlquist suggested using a test problem satisfying

$$\langle y - z, f(x, y) - f(x, z) \rangle \leq 0$$

A differential equation satisfying this condition is dissipative in the sense that

$$\frac{d}{dx}\|y(x) - z(x)\| \leq 0.$$

For linear multistep methods GD gave conditions for which the sequence

$$\|y_n - z_n\|$$

followed a similar behaviour.

Shortly afterwards a related theory was developed for Runge–Kutta methods and eventually for general linear methods.

Nonlinear stability

In 1975 G. Dahlquist suggested using a test problem satisfying

$$\langle y - z, f(x, y) - f(x, z) \rangle \leq 0$$

A differential equation satisfying this condition is dissipative in the sense that

$$\frac{d}{dx} \| y(x) - z(x) \| \leq 0.$$

For linear multistep methods GD gave conditions for which the sequence

$$\| y_n - z_n \|$$

followed a similar behaviour.
Shortly afterwards a related theory was developed for Runge–Kutta methods and eventually for general linear methods.

Symplectic condition

For Runge–Kutta methods the requirement for dissipative behaviour was

- $b_i \geq 0$ for all i and
- The matrix M is positive semidefinite where

$$M = \operatorname{diag}(b)A + A^\top \operatorname{diag}(b) - bb^\top$$

The matrix M was later identified by J M Sanz-Serna and others as playing a central role in the study of symplectic integration.

Specifically $M = 0$ is required for a method to be symplectic or canonical.

Symplectic condition

For Runge–Kutta methods the requirement for dissipative behaviour was

- $b_i \geq 0$ for all i and
- The matrix M is positive semidefinite where

$$M = \operatorname{diag}(b)A + A^\top \operatorname{diag}(b) - bb^\top$$

The matrix M was later identified by J M Sanz-Serna and others as playing a central role in the study of symplectic integration.

Specifically $M = 0$ is required for a method to be symplectic or canonical.

Symplectic condition

For Runge–Kutta methods the requirement for dissipative behaviour was

- $b_i \geq 0$ for all i and
- The matrix M is positive semidefinite where

$$M = \operatorname{diag}(b)A + A^\top \operatorname{diag}(b) - bb^\top$$

The matrix M was later identified by J M Sanz-Serna and others as playing a central role in the study of symplectic integration.
Specifically $M = 0$ is required for a method to be symplectic or canonical.

IMPLEMENTATION ISSUES

Solving the non-linear stage equations

For the implicit Euler method

$$y_n = y_{n-1} + hf(y_n)$$

applied to an N dimensional problem, the basic computational unit is the solution of a non-linear system of the form

$$Y - h\gamma f(Y) = \text{something already known} \qquad (*)$$

using an iteration scheme like this:

$$Y^{[k]} = Y^{[k-1]} - (I - h\gamma J)^{-1}(\text{something already known}) \qquad (\dagger)$$

For an s-stage implicit Runge–Kutta method implemented in the natural way, (*) would be replaced by an sN dimensional system and the cost of a single iteration (†) goes up by a large factor Hence there is a special interest in methods which can be transformed to a sequence of steps like (†).

Solving the non-linear stage equations

For the implicit Euler method

$$y_n = y_{n-1} + hf(y_n)$$

applied to an N dimensional problem, the basic computational unit is the solution of a non-linear system of the form

$$Y - h\gamma f(Y) = \text{something already known} \qquad (*)$$

using an iteration scheme like this:

$$Y^{[k]} = Y^{[k-1]} - (I - h\gamma J)^{-1}(\text{something already known}) \qquad (\dagger)$$

For an s-stage implicit Runge–Kutta method implemented in the natural way, (*) would be replaced by an sN dimensional system and the cost of a single iteration (†) goes up by a large factor. Hence there is a special interest in methods which can be transformed to a sequence of steps like (†).

Solving the non-linear stage equations

For the implicit Euler method

$$y_n = y_{n-1} + hf(y_n)$$

applied to an N dimensional problem, the basic computational unit is the solution of a non-linear system of the form

$$Y - h\gamma f(Y) = \text{something already known} \qquad (*)$$

using an iteration scheme like this:

$$Y^{[k]} = Y^{[k-1]} - (I - h\gamma J)^{-1}(\text{something already known}) \qquad (\dagger)$$

For an s-stage implicit Runge–Kutta method implemented in the natural way, (*) would be replaced by an sN dimensional system and the cost of a single iteration (†) goes up by a large factor

Hence there is a special interest in methods which can be transformed to a sequence of steps like (†).

Solving the non-linear stage equations

For the implicit Euler method

$$y_n = y_{n-1} + hf(y_n)$$

applied to an N dimensional problem, the basic computational unit is the solution of a non-linear system of the form

$$Y - h\gamma f(Y) = \text{something already known} \qquad (*)$$

using an iteration scheme like this:

$$Y^{[k]} = Y^{[k-1]} - (I - h\gamma J)^{-1}(\text{something already known}) \qquad (\dagger)$$

For an s-stage implicit Runge–Kutta method implemented in the natural way, (*) would be replaced by an sN dimensional system and the cost of a single iteration (†) goes up by a large factor Hence there is a special interest in methods which can be transformed to a sequence of steps like (†).

Adding transformations

Assume that the Jacobian matrix is slowly varying and that it doesn't need updating very often.

As a consequence, assume that the same Jacobian matrix approximation can be used for each stage.

In the case of a Runge–Kutta method, the non-linear equations are

$$Y = h(A \otimes I)F + \mathbf{1} \otimes y_0, \qquad F_i = f(Y_i),\ \ i = 1, 2, \ldots, s$$

and a single iteration gives an equation system

$$(I - hA \otimes J)(Y^{[k-1]} - Y^{[k]}) = Y^{[k-1]} - h(A \otimes I)F^{[k-1]} - \mathbf{1} \otimes y_0$$

There are two parts to the cost

(a) Occasionally: factorise $M := (I - hA \otimes J)$

(b) Every iteration: solve a system of the form $MD = E$

Adding transformations

Assume that the Jacobian matrix is slowly varying and that it doesn't need updating very often.

As a consequence, assume that the same Jacobian matrix approximation can be used for each stage.

In the case of a Runge–Kutta method, the non-linear equations are

$$Y = h(A \otimes I)F + \mathbf{1} \otimes y_0, \qquad F_i = f(Y_i),\ \ i = 1, 2, \dots, s$$

and a single iteration gives an equation system

$$(I - hA \otimes J)(Y^{[k-1]} - Y^{[k]}) = Y^{[k-1]} - h(A \otimes I)F^{[k-1]} - \mathbf{1} \otimes y_0$$

There are two parts to the cost

(a) Occasionally: factorise $M := (I - hA \otimes J)$

(b) Every iteration: solve a system of the form $MD = E$

Adding transformations

Assume that the Jacobian matrix is slowly varying and that it doesn't need updating very often.
As a consequence, assume that the same Jacobian matrix approximation can be used for each stage.
In the case of a Runge–Kutta method, the non-linear equations are

$$Y = h(A \otimes I)F + \mathbf{1} \otimes y_0, \qquad F_i = f(Y_i), \;\; i = 1, 2, \dots, s$$

and a single iteration gives an equation system

$$(I - hA \otimes J)(Y^{[k-1]} - Y^{[k]}) = Y^{[k-1]} - h(A \otimes I)F^{[k-1]} - \mathbf{1} \otimes y_0$$

There are two parts to the cost
(a) Occasionally: factorise $M := (I - hA \otimes J)$
(b) Every iteration: solve a system of the form $MD = E$

Adding transformations

Assume that the Jacobian matrix is slowly varying and that it doesn't need updating very often.
As a consequence, assume that the same Jacobian matrix approximation can be used for each stage.
In the case of a Runge–Kutta method, the non-linear equations are

$$Y = h(A \otimes I)F + \mathbf{1} \otimes y_0, \qquad F_i = f(Y_i), \;\; i = 1, 2, \dots, s$$

and a single iteration gives an equation system

$$(I - hA \otimes J)(Y^{[k-1]} - Y^{[k]}) = Y^{[k-1]} - h(A \otimes I)F^{[k-1]} - \mathbf{1} \otimes y_0$$

There are two parts to the cost
(a) Occasionally: factorise $M := (I - hA \otimes J)$
(b) Every iteration: solve a system of the form $MD = E$

The costs of parts (a) and (b) are in general

$$\text{(a): } s^3N^3, \qquad \text{(b) } s^2N^2$$

But if A is singly implicit with $\sigma(A) = \{\gamma\}$, we can do better.

Let T be the similarity which transforms A to Jordan canonical form

$$T^{-1}AT = \overline{A},$$

where $\overline{A}$ can be assumed to be lower triangular.
The equation

$$(I - hA \otimes J)D = E,$$

transforms to

$$(I - h\overline{A} \otimes J)T^{-1}D = T^{-1}E$$

and the costs becomes

$$\text{(a): } N^3, \qquad \text{(b) } sN^2$$

There is a small (s^2N) cost for the transformations but for large N this is of no consequence.

The costs of parts (a) and (b) are in general

$$(a):\ s^3N^3, \qquad (b)\ s^2N^2$$

But if A is singly implicit with $\sigma(A) = \{\gamma\}$, we can do better.

Let T be the similarity which transforms A to Jordan canonical form

$$T^{-1}AT = \overline{A},$$

where $\overline{A}$ can be assumed to be lower triangular.
The equation

$$(I - hA \otimes J)D = E,$$

transforms to

$$(I - h\overline{A} \otimes J)T^{-1}D = T^{-1}E$$

and the costs becomes

$$(a):\ N^3, \qquad (b)\ sN^2$$

There is a small (s^2N) cost for the transformations but for large N this is of no consequence.

The costs of parts (a) and (b) are in general

$$\text{(a): } s^3N^3, \qquad \text{(b) } s^2N^2$$

But if A is singly implicit with $\sigma(A) = \{\gamma\}$, we can do better.

Let T be the similarity which transforms A to Jordan canonical form

$$T^{-1}AT = \overline{A},$$

where $\overline{A}$ can be assumed to be lower triangular.

The equation

$$(I - hA \otimes J)D = E,$$

transforms to

$$(I - h\overline{A} \otimes J)T^{-1}D = T^{-1}E$$

and the costs becomes

$$\text{(a): } N^3, \qquad \text{(b) } sN^2$$

There is a small (s^2N) cost for the transformations but for large N this is of no consequence.

The costs of parts (a) and (b) are in general

$$\text{(a): } s^3N^3, \qquad \text{(b) } s^2N^2$$

But if A is singly implicit with $\sigma(A) = \{\gamma\}$, we can do better.

Let T be the similarity which transforms A to Jordan canonical form

$$T^{-1}AT = \overline{A},$$

where $\overline{A}$ can be assumed to be lower triangular.
The equation

$$(I - hA \otimes J)D = E,$$

transforms to

$$(I - h\overline{A} \otimes J)T^{-1}D = T^{-1}E$$

and the costs becomes

$$\text{(a): } N^3, \qquad \text{(b) } sN^2$$

There is a small (s^2N) cost for the transformations but for large N this is of no consequence.

The costs of parts (a) and (b) are in general

$$\text{(a): } s^3N^3, \qquad \text{(b) } s^2N^2$$

But if A is singly implicit with $\sigma(A) = \{\gamma\}$, we can do better.

Let T be the similarity which transforms A to Jordan canonical form

$$T^{-1}AT = \overline{A},$$

where $\overline{A}$ can be assumed to be lower triangular.
The equation

$$(I - hA \otimes J)D = E,$$

transforms to

$$(I - h\overline{A} \otimes J)T^{-1}D = T^{-1}E$$

and the costs becomes

$$\text{(a): } N^3, \qquad \text{(b) } sN^2$$

There is a small (s^2N) cost for the transformations but for large N this is of no consequence.

The costs of parts (a) and (b) are in general

$$\text{(a): } s^3N^3, \qquad \text{(b) } s^2N^2$$

But if A is singly implicit with $\sigma(A) = \{\gamma\}$, we can do better.

Let T be the similarity which transforms A to Jordan canonical form

$$T^{-1}AT = \overline{A},$$

where $\overline{A}$ can be assumed to be lower triangular.
The equation

$$(I - hA \otimes J)D = E,$$

transforms to

$$(I - h\overline{A} \otimes J)T^{-1}D = T^{-1}E$$

and the costs becomes

$$\text{(a): } N^3, \qquad \text{(b) } sN^2$$

There is a small (s^2N) cost for the transformations but for large N this is of no consequence.

The costs of parts (a) and (b) are in general

$$\text{(a): } s^3N^3, \qquad \text{(b) } s^2N^2$$

But if A is singly implicit with $\sigma(A) = \{\gamma\}$, we can do better.

Let T be the similarity which transforms A to Jordan canonical form

$$T^{-1}AT = \overline{A},$$

where $\overline{A}$ can be assumed to be lower triangular.
The equation

$$(I - hA \otimes J)D = E,$$

transforms to

$$(I - h\overline{A} \otimes J)T^{-1}D = T^{-1}E$$

and the costs becomes

$$\text{(a): } N^3, \qquad \text{(b) } sN^2$$

There is a small (s^2N) cost for the transformations but for large N this is of no consequence.

Thank you for your attention