General linear methods for ordinary differential equations

John Butcher

The University of Auckland, New Zealand

Acknowledgements:

Will Wright, Zdzisław Jackiewicz, Helmut Podhaisky

Solving ordinary differential equations numerically is, even today, still a great challenge.

This applies especially to stiff differential equations and to closely related problems involving algebraic constraints (DAEs).

Although the problem seems to be solved — there are already highly efficient codes based on Runge–Kutta methods and linear multistep methods — questions concerning methods that lie between the traditional families are still open.

I will talk today about some aspects of these so-called "General linear methods".

Contents

- Introduction to stiff problems
- Why do we want a new numerical method?
- What are General Linear Methods?
- What we want from General Linear Methods
- Propagation of errors
- Algebraic analysis of order
- High order and stage order
- General Linear Methods and Runge–Kutta stability
- Doubly companion matrices
- Inherent Runge–Kutta stability
- Proof of stability result
- Example methods
- Implementation questions

Introduction to stiff problems

We will write a standard initial value problem in the form

$$y'(x) = f(x, y(x)), \quad y(x_0) = y_0.$$

The variable x will be referred to as "time" even if it does not represent physical time.

Consider the problem

$$\begin{bmatrix} y_1'(x) \\ y_2'(x) \\ y_3'(x) \end{bmatrix} = \begin{bmatrix} y_2(x) \\ -y_1(x) \\ -Ly_1(x) + y_2(x) + Ly_3(x) \end{bmatrix},$$

with initial values $y_1(0) = 0, y_2(0) = 1, y_3(0) = \epsilon$.

If $\epsilon = 0$, the solution is

$$y(x) = \begin{bmatrix} \sin(x) \\ \cos(x) \\ \sin(x) \end{bmatrix}.$$

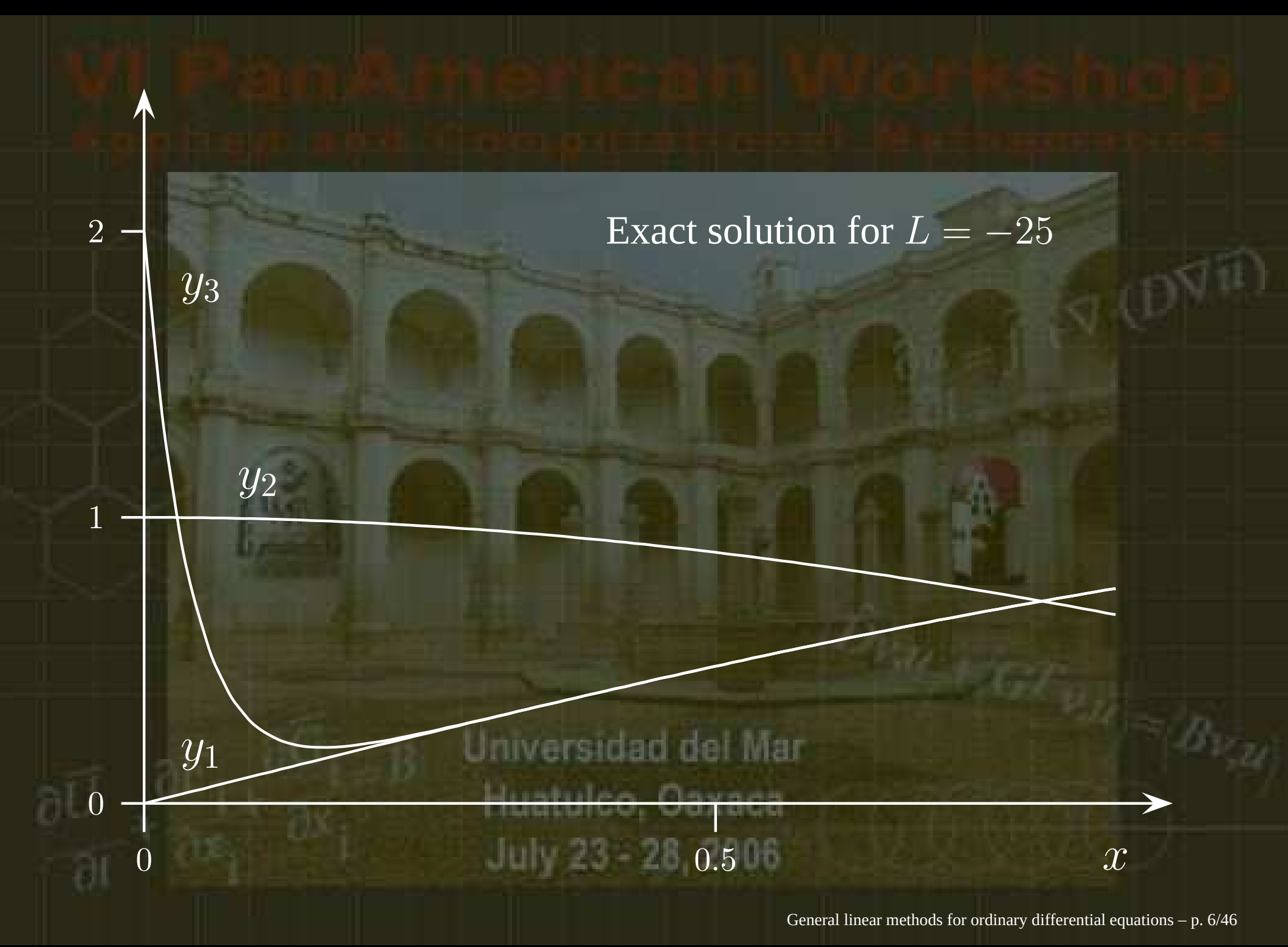
Change the value of ϵ and the third component changes to

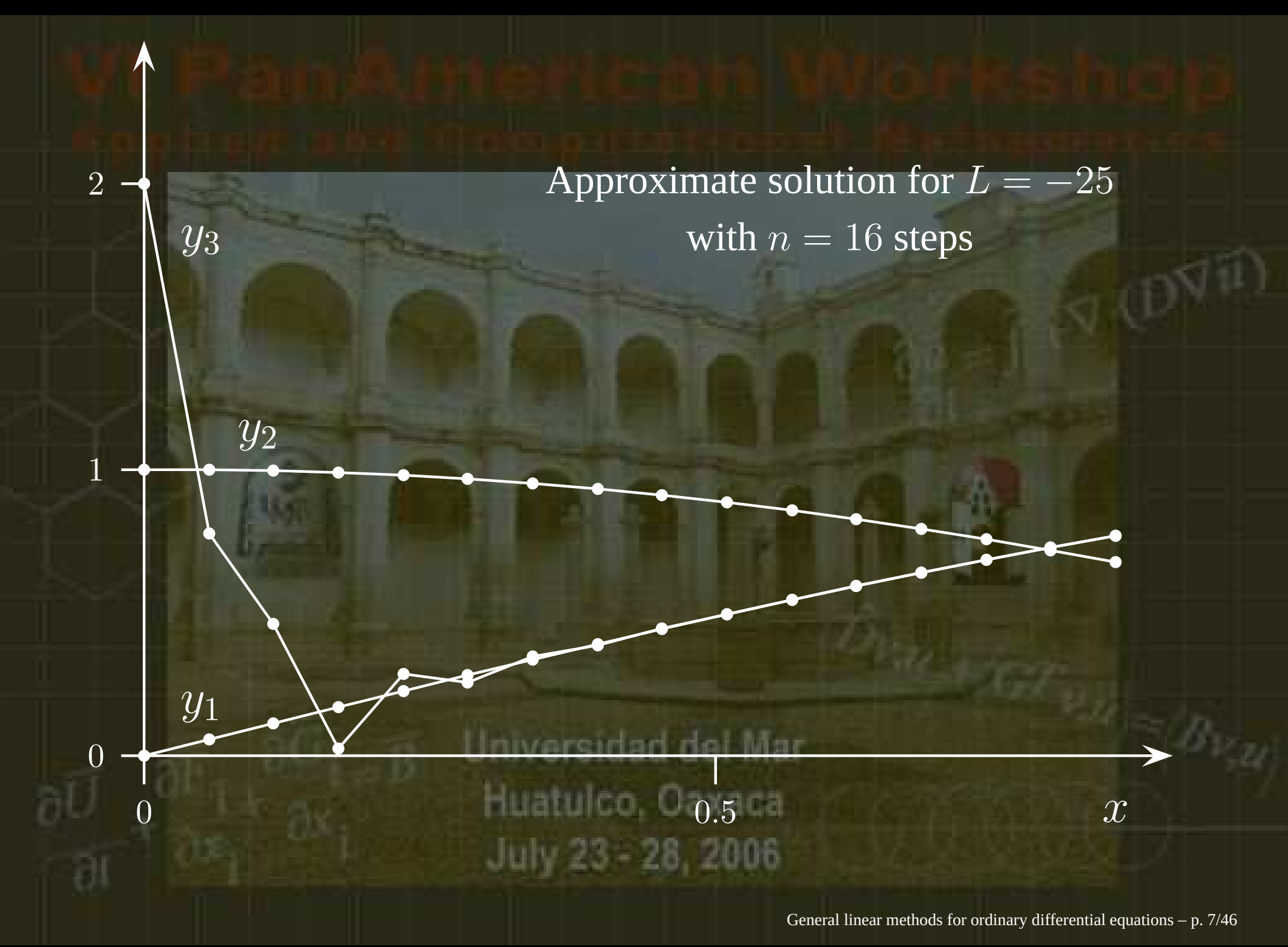
$$\sin(x) + \epsilon \exp(Lx).$$

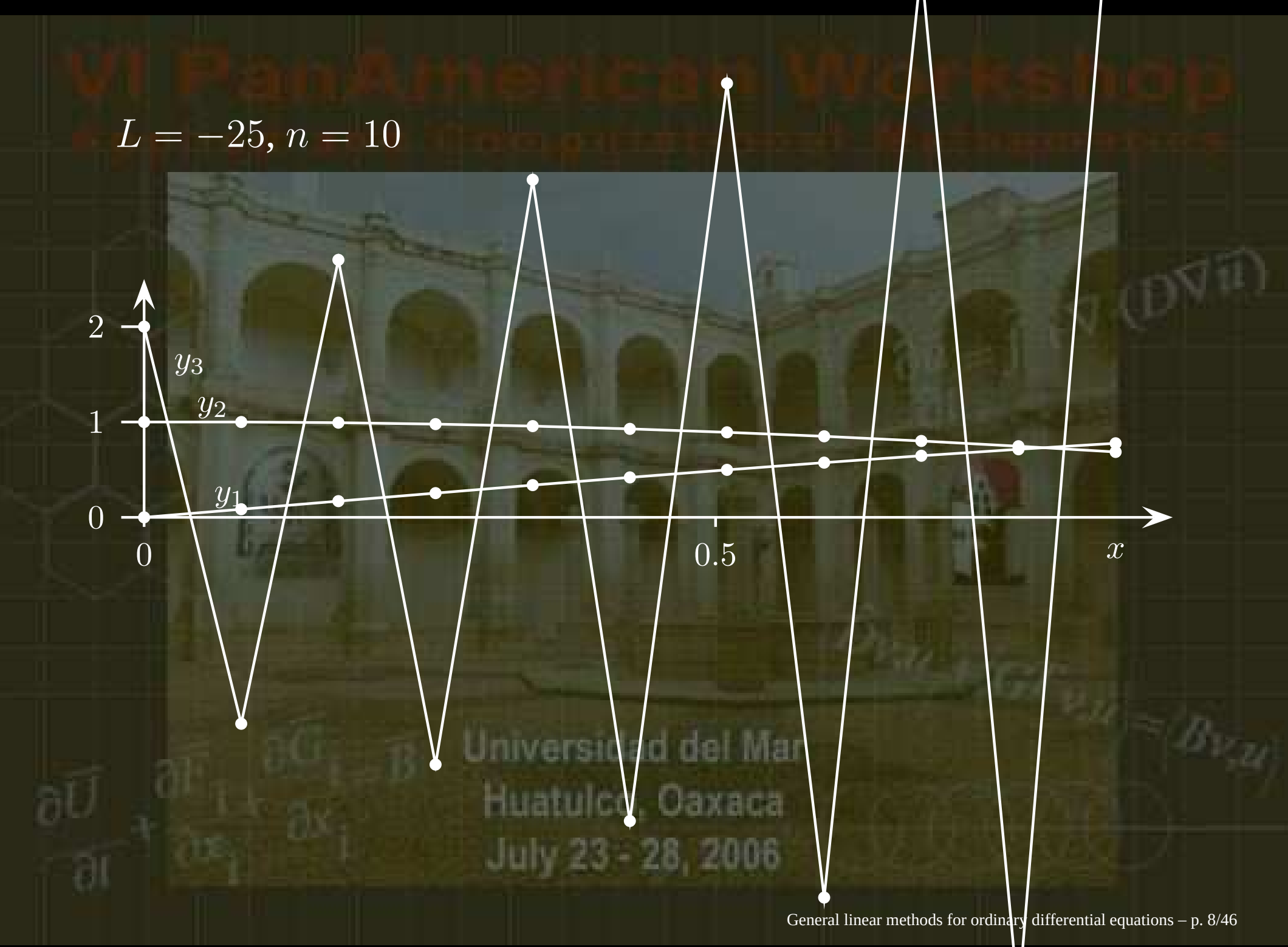
We will see how this works for values of $L = -25$ and $\epsilon = 2$.

The solution is plotted for the interval $[0, 0.85]$.

For the approximate solution computed by the Euler method, we choose $n = 16$ steps and then decrease n to 10.


Exact solution for $L = -25$
y_3
y_2
y_1
2
1
0
0
0.5
x


Approximate solution for $L = -25$
with $n = 16$ steps
y_3
y_2
y_1
2
1
0
0
0.5
x


$L = -25, n = 10$
2
1
0
y_3
y_2
y_1
0
0.5
x

Note that there seems to be no difficulty approximating y_1 and y_2, even for quite large stepsizes.

For stepsizes less than $h = 0.08$, the approximation to y_3 converges to the same value as y_1, but not in precisely the same way as for the exact solution.

For stepsizes greater than $h = 0.08$ the approximations to y_3 are hopeless.

We can understand what happens better by studying the difference $y_3 - y_1$.

The value of $y_3 - y_1$ satisfies the equation

$$y'(x) = Ly(x),$$

with solution const $\cdot \exp(Lx)$.

The exact solution to $y' = Ly$ is multiplied by $\exp(z)$ (where $z = hL$) in each time step but what about the *computed* solution?

According to the formula for the Euler method, $y_n = y_{n-1} + hf(t_{n-1}, y_{n-1})$, the numerical solution is multiplied instead by

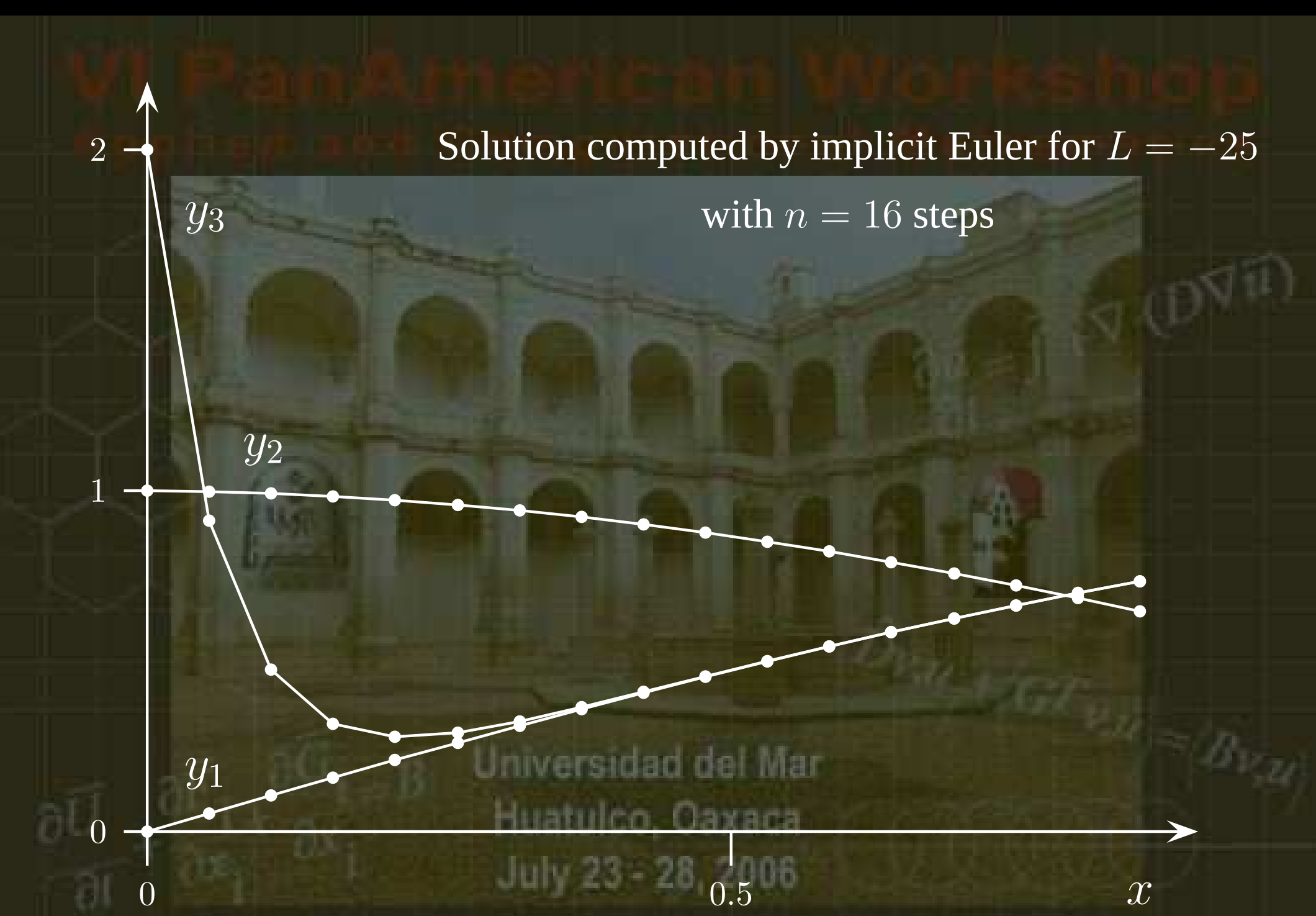
$$1 + z$$

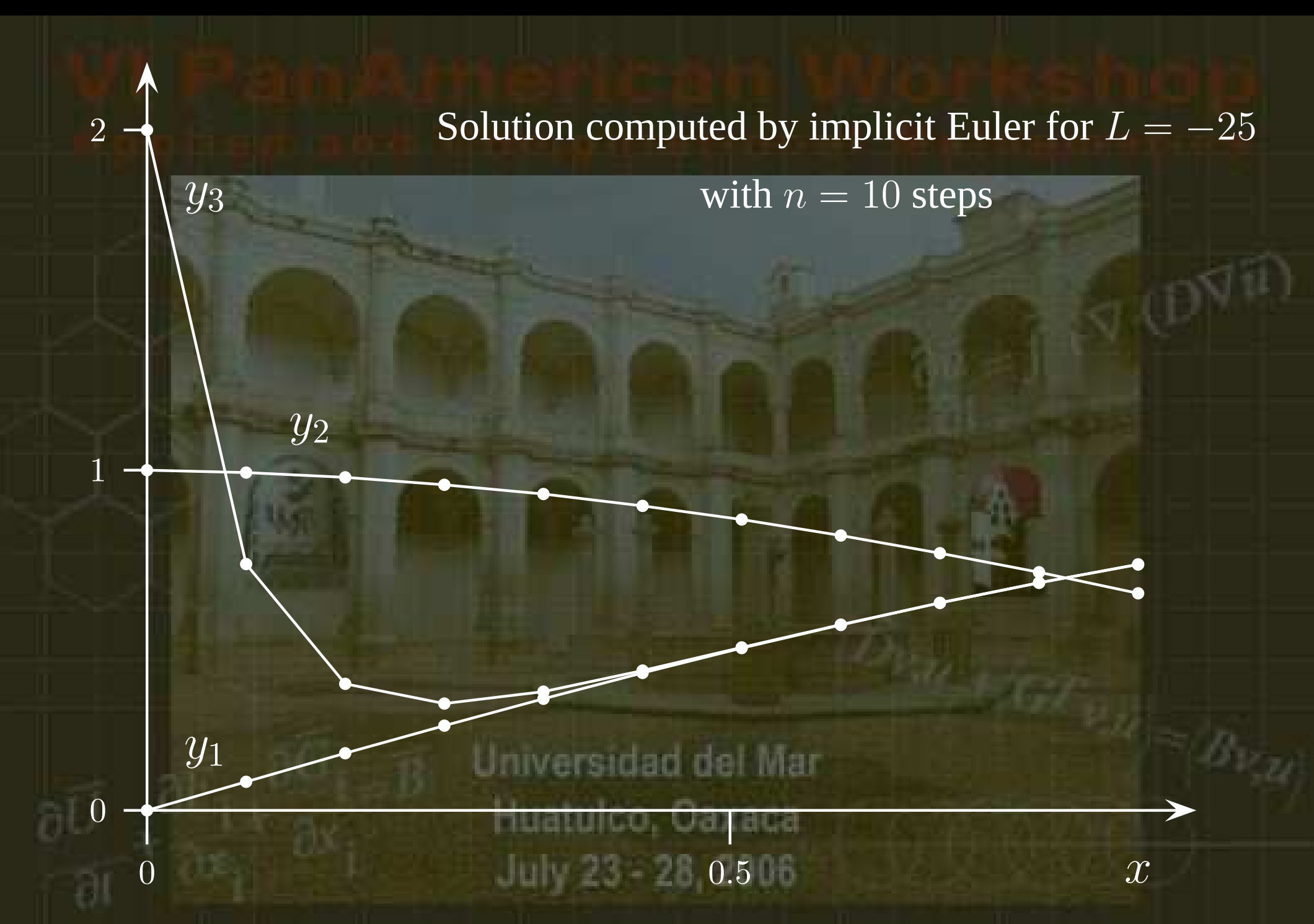
in each time step.

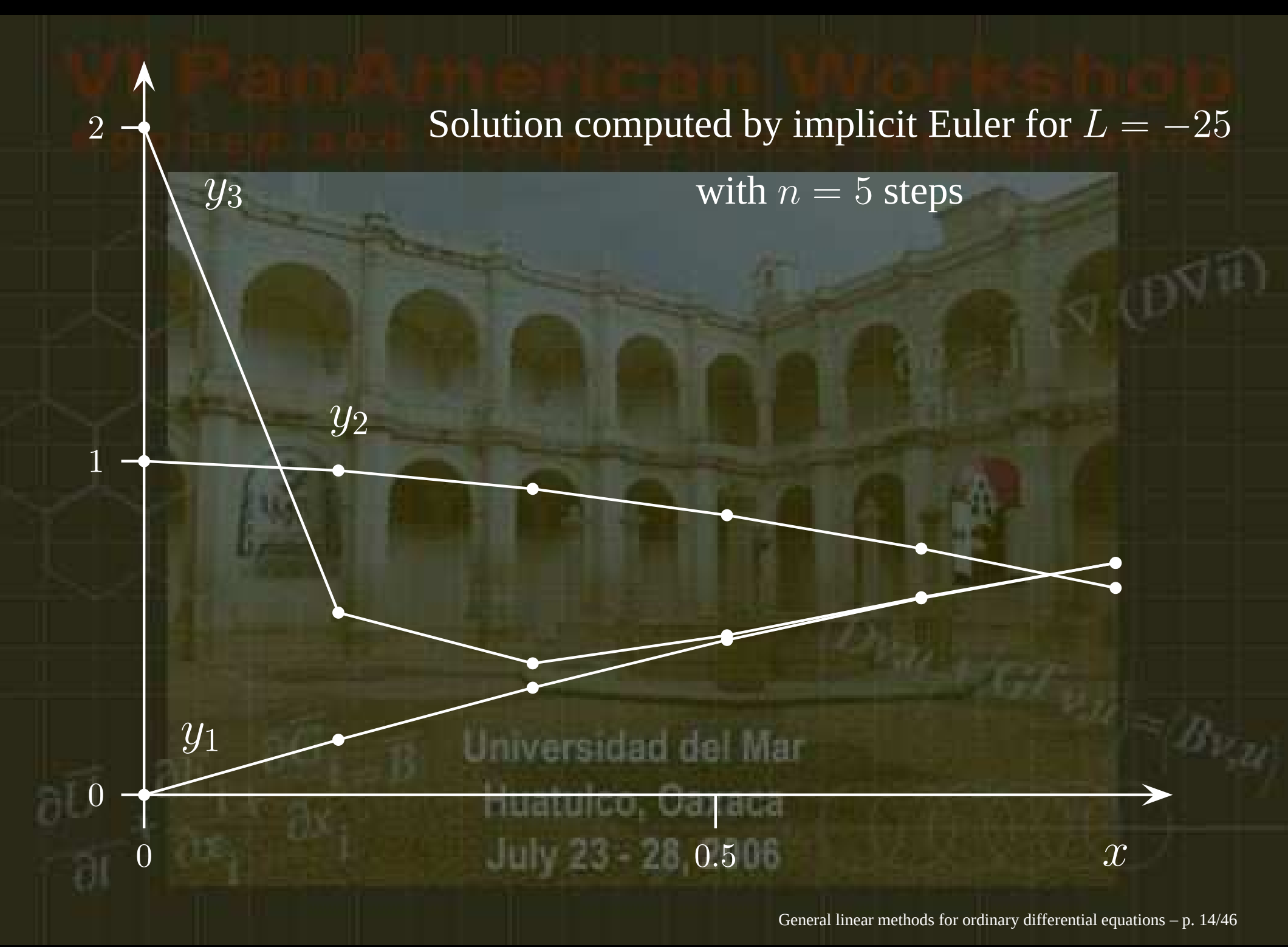
The fact that $1 + z$ is a poor approximation to $\exp(z)$ when z is a very negative number, is at the heart of the difficulty in solving stiff problems.

Sometimes the "implicit Euler method" is used for the solution of stiff problems because in this case the approximation $\exp(z) \approx 1 + z$ is replaced by $\exp(z) \approx (1 - z)^{-1}$.

To see what happens in the case of the contrived problem we have considered, when Euler is replaced by implicit Euler, we present three further figures.


Solution computed by implicit Euler for $L = -25$
with $n = 16$ steps
y_3
y_2
y_1
2
1
0
0
0.5
x


Solution computed by implicit Euler for $L = -25$
with $n = 10$ steps
y_3
y_2
y_1
2
1
0
0
0.5
x


Solution computed by implicit Euler for $L = -25$
with $n = 5$ steps
y_3
y_2
y_1
2
1
0
0
0.5
x

It looks as though, even for n as low as 5, we get completely acceptable performance.

The key to the reason for this is that $|(1-z)^{-1}|$ is bounded by 1 for all z in the left half-plane.

This means that a purely linear problem, with constant coefficients, will have stable numerical behaviour whenever the same is true for the exact solution.

Methods with this property are said to be “A-stable”.

Although by no means a guarantee that it will solve all stiff problems adequately, A-stability is an important attribute and a useful certification.

Why do we want a new numerical method?

We want methods that are efficient for stiff problems

- Hence we cannot use highly implicit Runge-Kutta methods
- Hence we want to avoid non- A-stable BDF methods

We want methods that have high stage order

- Otherwise order reduction can occur for stiff problems
- Otherwise it is difficult to estimate accuracy
- Otherwise it is difficult to interpolate for dense output

We want methods for which asymptotically correct error estimates can be found

- This means having accurate information available in the step to make this possible
- High stage order makes this easier

We want methods for which changes to adjacent orders can be considered and evaluated

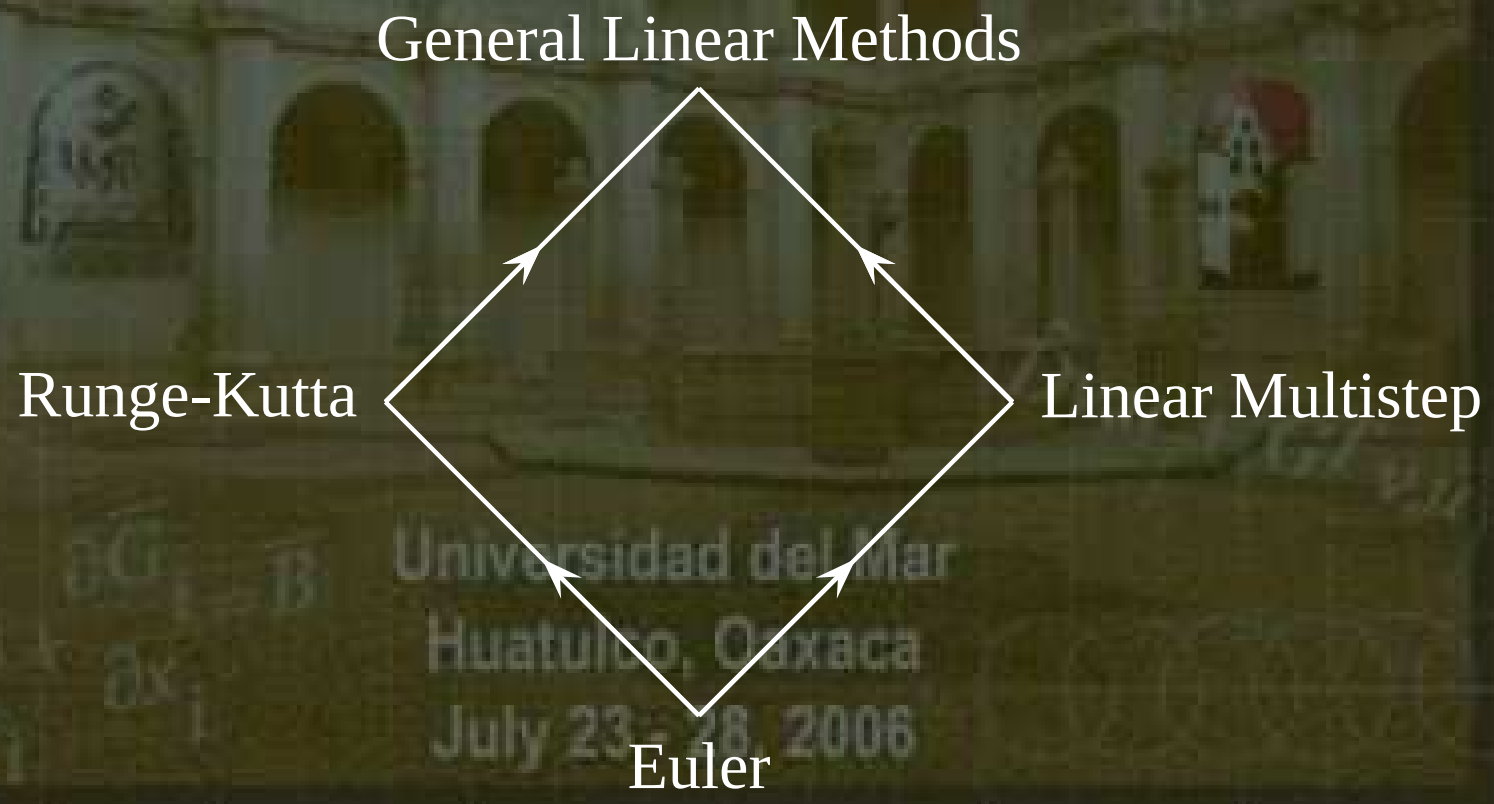
- This means being able to evaluate, for example, $h^{p+2}y^{(p+2)}$ as well as $h^{p+1}y^{(p+1)}$

  (p denotes the order of the method)
- Evaluating the error for lower order alternative methods is usually easy

What are General Linear Methods?

General linear methods are the natural generalization of Runge–Kutta (multi-stage) methods and linear multistep (multivalue) methods.

This can be illustrated in a diagram.



We will consider r-value, s-stage methods, where

$r = 1$ for a Runge–Kutta method

and

$s = 1$ for a linear multistep method.

Each step of the computation takes as input a certain number (r) of items of data and generates for output the same number of items.

The output items correspond to the input items but advanced through one time step (h).

Within the step a certain number (s) of stages of computation are performed.

Denote by p the order of the method and by q the "stage-order".

At the start of step number n, denote the input items by $y_i^{[n-1]}, i = 1, 2, \ldots, r$.

Denote the stages computed in the step and the stage derivatives by Y_i and F_i respectively, $i = 1, 2, \ldots, s$.

For a compact notation, write

$$y^{[n-1]} = \begin{bmatrix} y_1^{[n-1]} \\ y_2^{[n-1]} \\ \vdots \\ y_r^{[n-1]} \end{bmatrix}, \quad y^{[n]} = \begin{bmatrix} y_1^{[n]} \\ y_2^{[n]} \\ \vdots \\ y_r^{[n]} \end{bmatrix}, Y = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_s \end{bmatrix}, F = \begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_s \end{bmatrix}$$

The stages are computed by the formulae

$$Y_i = \sum_{j=1}^{s} a_{ij} h F_j + \sum_{j=1}^{r} u_{ij} y_j^{[n-1]}, \quad i = 1, 2, \ldots, s$$

and the output approximations by the formulae

$$y_i^{[n]} = \sum_{j=1}^{s} b_{ij} h F_j + \sum_{j=1}^{r} v_{ij} y_j^{[n-1]}, \quad i = 1, 2, \ldots, r$$

These can be written together using a partitioned matrix

$$\begin{bmatrix} Y \\ y^{[n]} \end{bmatrix} = \begin{bmatrix} A & U \\ B & V \end{bmatrix} \begin{bmatrix} hF \\ y^{[n-1]} \end{bmatrix}$$

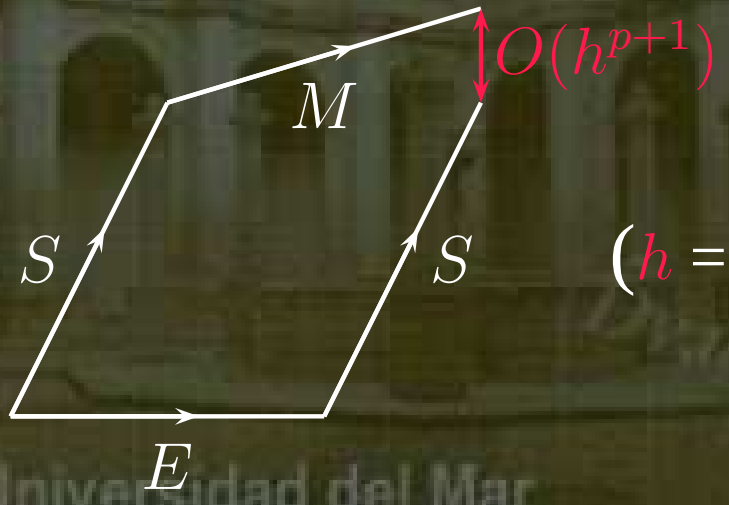
What we want from General Linear Methods

- Convergence (consistency and stability)
- Local accuracy
- Global accuracy (understand propagation of errors)
- A-stability for stiff problems
- RK stability (behaves like Runge-Kutta method)

Propagation of errors

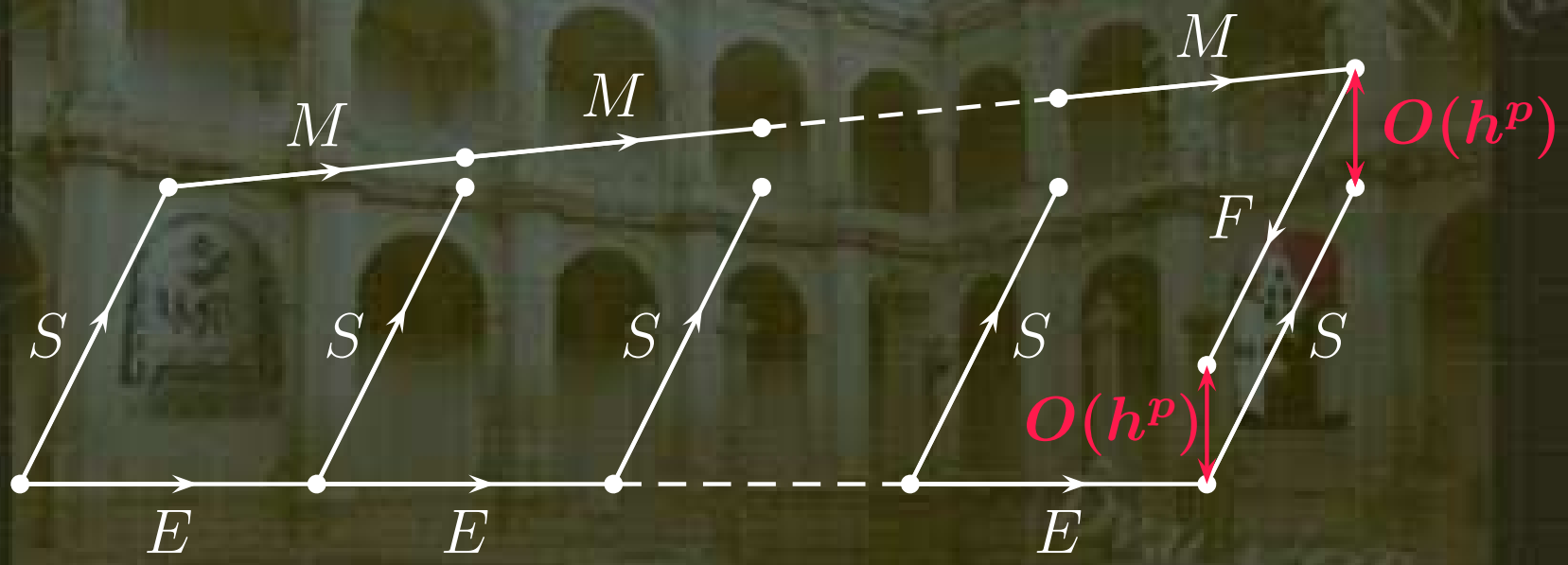
Let S denote a "starting method", that is a mapping from $\mathbb{R}^N$ to $\mathbb{R}^{rN}$ and a corresponding finishing method $F : \mathbb{R}^{rN} \to \mathbb{R}^N$ such that $F \circ S = \mathrm{id}$

The order of accuracy of a multivalue method is defined in terms of the diagram



(h = stepsize)

By duplicating this diagram over many steps, global error estimates may be found



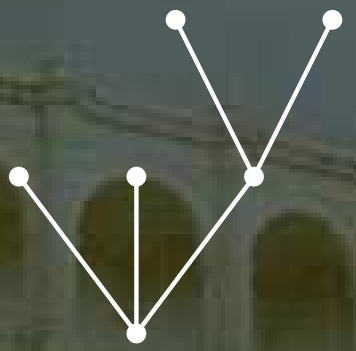
Algebraic analysis of order

We recall the way order is defined for Runge-Kutta methods.
We consider the solution of an autonomous differential equation $y'(x) = f(y(x))$.
Let T denote the set of rooted trees

To each of these is associated an "elementary differential" $F(t)$ defined in terms of the function f.
Also associated with each tree is its symmetry $\sigma(t)$ and its density $\gamma(t)$. In the following table, $r(t)$ is the "order" (number of vertices) of t.

$r(t)$	t	$\sigma(t)$	$\gamma(t)$	$F(t)$	$F(t)^i$
1		1	1	f	f^i
2		1	2	$f'f$	$f^i_j f^j$
3		2	3	$f''(f,f)$	$f^i_{jk} f^j f^k$
3		1	6	$f'f'f$	$f^i_j f^j_k f^k$
4		6	4	$f'''(f,f,f)$	$f^i_{jkl} f^j f^k f^l$
4		1	8	$f''(f,f'f)$	$f^i_{jk} f^j f^k_l f^l$
4		2	12	$f'f''(f,f)$	$f^i_j f^j_{kl} f^k f^l$
4		1	24	$f'f'f'f$	$f^i_j f^j_k f^k_l f^l$

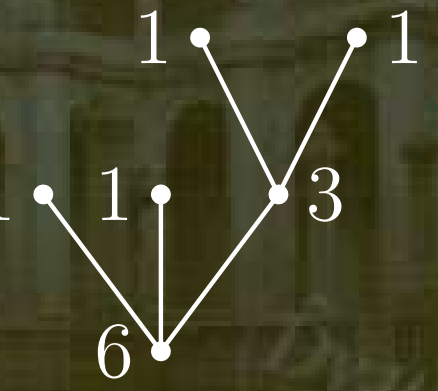
We will look at a single example tree in more detail.

$t =$ 

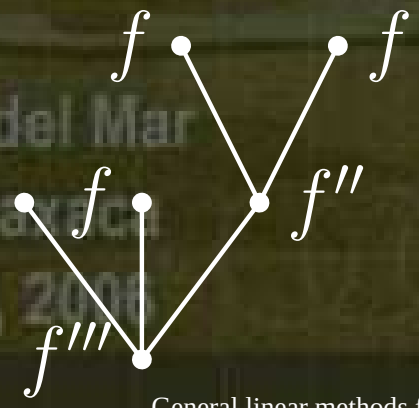
$r(t) = 6$

$\sigma(t) = 4$

$\gamma(t) = 18$



$F(t) = f'''(f,f,f''(f,f))$



The solution to the standard autonomous initial-value problem

$$y'(x) = f(y(x)), \qquad y(x_0) = y_0$$

has the formal Taylor series

$$y(x_0 + h) = y_0 + \sum_{t \in T} \frac{h^{r(t)}}{\sigma(t)\gamma(t)} F(t)(y_0)$$

A Runge-Kutta method has the same series but with $1/\gamma(t)$ replaced by a sequence of "elementary weights" characteristic of the specific Runge-Kutta method.

Expressions of this type are usually referred to as B-series.

Let G denote the set of mappings $T^{\#} \rightarrow \mathbb{R}$ where $T^{\#} = \{\emptyset\} \cup T$ (and $\emptyset$ is the empty tree).

Also write G_0 and G_1 as subsets for which $\emptyset \mapsto 0$ or $\emptyset \mapsto 1$ respectively.

A mapping $G_1 \times G \rightarrow G$ can be defined which represents compositions of various operations.

In particular if this is restricted to $G_1 \times G_1 \rightarrow G_1$ it becomes a group operation.

Let E denote the mapping $t \mapsto 1/\gamma(t)$ and D the mapping which takes every tree to zero except the unique order 1 tree, which maps to one.

A general linear method defined from the matrices $[A, U, B, V]$ has order p if there exists $\xi \in G^r$ and $\eta \in G_1^s$, such that

$$\eta = A\eta D + U\xi,$$
$$E\xi = B\eta D + V\xi, \qquad (*)$$

where pre-multiplication by E and post-multiplication by D are scalar times vector products and (*) is to hold only up to trees of order p.

High order and stage order

If we consider only methods with $q = p$, then simpler criteria can be found.
Let $\exp(cz)$ denote the component-by-component exponential then order p and stage-order $q \geq p - 1$ is equivalent to

$$\exp(cz) = zA \exp(cz) + Uw(z) + O(z^{q+1}),$$

$$\exp(z)w(z) = zB \exp(cz) + Vw(z) + O(z^{p+1}),$$

where the power series

$$w(z) = w_0 + w_1 z + \cdots + w_p z^p$$

represents an input approximation

$$y^{[0]} = \sum_{i=0}^{p} w_i h^i y^{(i)}(x_0).$$

GL Methods and RK stability

A General Linear Method is “Runge–Kutta stable” if its stability matrix

$$M(z) = V + zB(I - zA)^{-1}U$$

has only one non-zero eigenvalue.

Armed with this property we should expect attraction to the manifold $S(\mathbb{R}^N)$, and stable adherence to this manifold.

This means that the method starts acting like a one-step method.

Doubly companion matrices

Matrices like the following are “companion matrices” for the polynomial

$$z^n + \alpha_1 z^{n-1} + \cdots + \alpha_n$$

or

$$z^n + \beta_1 z^{n-1} + \cdots + \beta_n,$$

respectively:

$$\begin{bmatrix} -\alpha_1 & -\alpha_2 & -\alpha_3 & \cdots & -\alpha_{n-1} & -\alpha_n \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{bmatrix}, \quad \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & -\beta_n \\ 1 & 0 & 0 & \cdots & 0 & -\beta_{n-1} \\ 0 & 1 & 0 & \cdots & 0 & -\beta_{n-2} \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & -\beta_2 \\ 0 & 0 & 0 & \cdots & 1 & -\beta_1 \end{bmatrix}$$

Their characteristic polynomials can be found from $\det(I - zA) = \alpha(z)$ or $\beta(z)$, respectively, where,
$\alpha(z) = 1+\alpha_1 z+\cdots+\alpha_n z^n, \quad \beta(z) = 1+\beta_1 z+\cdots+\beta_n z^n.$
A matrix with both α and β terms:

$$X = \begin{bmatrix} -\alpha_1 & -\alpha_2 & -\alpha_3 & \cdots & -\alpha_{n-1} & -\alpha_n - \beta_n \\ 1 & 0 & 0 & \cdots & 0 & -\beta_{n-1} \\ 0 & 1 & 0 & \cdots & 0 & -\beta_{n-2} \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & -\beta_2 \\ 0 & 0 & 0 & \cdots & 1 & -\beta_1 \end{bmatrix},$$

is known as a "doubly companion matrix" and has characteristic polynomial defined by

$$\det(I - zX) = \alpha(z)\beta(z) + O(z^{n+1})$$

Matrices Ψ^{-1} and Ψ transforming X to Jordan canonical form are known.

In the special case of a single Jordan block with n-fold eigenvalue λ, we have

$$\Psi^{-1} = \begin{bmatrix} 1 & \lambda + \alpha_1 & \lambda^2 + \alpha_1\lambda + \alpha_2 & \cdots \\ 0 & 1 & 2\lambda + \alpha_1 & \cdots \\ 0 & 0 & 1 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix},$$

where row number $i + 1$ is formed from row number i by differentiating with respect to λ and dividing by i.

We have a similar expression for Ψ:

$$\Psi = \begin{bmatrix} \ddots & \vdots & \vdots & \vdots \\ \cdots & 1 & 2\lambda+\beta_1 & \lambda^2+\beta_1\lambda+\beta_2 \\ \cdots & 0 & 1 & \lambda+\beta_1 \\ \cdots & 0 & 0 & 1 \end{bmatrix}$$

The Jordan form is $\Psi^{-1}X\Psi = J + \lambda I$, where $J_{ij} = \delta_{i,j+1}$. That is

$$\Psi^{-1}X\Psi = \begin{bmatrix} \lambda & 0 & \cdots & 0 & 0 \\ 1 & \lambda & \cdots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & \lambda & 0 \\ 0 & 0 & \cdots & 1 & \lambda \end{bmatrix}$$

Inherent Runge–Kutta stability

Using doubly companion matrices, it is possible to construct GL methods possessing RK stability using rational operations. The methods constructed in this way are said to possess "Inherent Runge–Kutta Stability". Apart from exceptional cases, (in which certain matrices are singular), we characterize the method with $r = s = p + 1 = q + 1$ by the parameters

- λ single eigenvalue of A
- $c_1, c_2, \ldots, c_s$ stage abscissae
- Error constant
- $\beta_1, \beta_2, \ldots, \beta_p$ elements in last column of $s \times s$ doubly companion matrix X
- Information on the structure of V

Consider only methods for which the step n outputs approximate the "Nordsieck vector":

$$\begin{bmatrix} y_1^{[n]} \\ y_2^{[n]} \\ y_3^{[n]} \\ \vdots \\ y_{p+1}^{[n]} \end{bmatrix} \approx \begin{bmatrix} y(x_n) \\ hy'(x_n) \\ h^2y''(x_n) \\ \vdots \\ h^py^{(p)}(x_n) \end{bmatrix}$$

For such methods, V has the form

$$V = \begin{bmatrix} 1 & v^T \\ 0 & \dot{V} \end{bmatrix}$$

Such a method has the IRKS property if a doubly companion matrix X exists so that for some vector ξ,

$$BA = XB, \quad BU = XV - VX + e_1\xi^T, \quad \rho(\dot{V}) = 0$$

It will be shown that, for such methods, the stability matrix satisfies

$$M(z) \sim V + ze_1\xi^T(I - zX)^{-1}$$

which has all except one of its eigenvalues zero. The non-zero eigenvalue has the role of stability function

$$R(z) = \frac{N(z)}{(1 - \lambda z)^s}$$

Proof of stability result

$$
\begin{aligned}
&(I - zX)M(z)(I - zX)^{-1} \\
&= (V - zXV + z(B - zXB)(I - zA)^{-1}U)(I - zX)^{-1} \\
&= (V - zXV + zBU)(I - zX)^{-1} \\
&= V + ze_1\xi^T(I - zX)^{-1}
\end{aligned}
$$

$XB = BA$

$BU = XV - VX + e_1\xi^T$

This has a single non-zero eigenvalue equal to

$$R(z) = 1 + z\xi^T(I - zX)^{-1}e_1 = \frac{\det(I + z(e_1\xi^T - X))}{\det(I - zX)}$$

Example methods

The following third order method is explicit and suitable for the solution of non-stiff problems

$$\begin{bmatrix} AU \\ BV \end{bmatrix} = \left[\begin{array}{cccc|cccc} 0 & 0 & 0 & 0 & 1 & \frac{1}{4} & \frac{1}{32} & \frac{1}{384} \\ -\frac{176}{1885} & 0 & 0 & 0 & 1 & \frac{2237}{3770} & \frac{2237}{15080} & \frac{2149}{90480} \\ -\frac{335624}{311025} & \frac{29}{55} & 0 & 0 & 1 & \frac{1619591}{1244100} & \frac{260027}{904800} & \frac{1517801}{39811200} \\ -\frac{67843}{6435} & \frac{395}{33} & -5 & 0 & 1 & \frac{29428}{6435} & \frac{527}{585} & \frac{41819}{102960} \\ \hline -\frac{67843}{6435} & \frac{395}{33} & -5 & 0 & 1 & \frac{29428}{6435} & \frac{527}{585} & \frac{41819}{102960} \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ \frac{82}{33} & -\frac{274}{11} & \frac{170}{9} & -\frac{4}{3} & 0 & \frac{482}{99} & 0 & -\frac{161}{264} \\ -8 & -12 & \frac{40}{3} & -2 & 0 & \frac{26}{3} & 0 & 0 \end{array}\right]$$

The following fourth order method is implicit, L-stable, and suitable for the solution of stiff problems

$$\left[\begin{array}{ccccc|ccccc}
\frac{1}{4} & 0 & 0 & 0 & 0 & 1 & \frac{3}{4} & \frac{1}{2} & \frac{1}{4} & 0 \\
-\frac{513}{54272} & \frac{1}{4} & 0 & 0 & 0 & 1 & \frac{27649}{54272} & \frac{5601}{27136} & \frac{1539}{54272} & -\frac{459}{6784} \\
\frac{3706119}{69088256} & -\frac{488}{3819} & \frac{1}{4} & 0 & 0 & 1 & \frac{15366379}{207264768} & \frac{756057}{34544128} & \frac{1620299}{69088256} & -\frac{4854}{454528} \\
\frac{32161061}{197549232} & -\frac{111814}{232959} & \frac{134}{183} & \frac{1}{4} & 0 & 1 & -\frac{32609017}{197549232} & \frac{929753}{32924872} & \frac{4008881}{32924872} & \frac{174981}{3465776} \\
-\frac{135425}{2948496} & -\frac{641}{10431} & \frac{73}{183} & \frac{1}{2} & \frac{1}{4} & 1 & -\frac{367313}{8845488} & -\frac{22727}{1474248} & \frac{40979}{982832} & \frac{323}{25864} \\
\hline
-\frac{135425}{2948496} & -\frac{641}{10431} & \frac{73}{183} & \frac{1}{2} & \frac{1}{4} & 1 & -\frac{367313}{8845488} & -\frac{22727}{1474248} & \frac{40979}{982832} & \frac{323}{25864} \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
\frac{2255}{2318} & -\frac{47125}{20862} & \frac{447}{122} & -\frac{11}{4} & \frac{4}{3} & 0 & -\frac{28745}{20862} & -\frac{1937}{13908} & \frac{351}{18544} & \frac{65}{976} \\
\frac{12620}{10431} & -\frac{96388}{31293} & \frac{3364}{549} & -\frac{10}{3} & \frac{4}{3} & 0 & -\frac{70634}{31293} & -\frac{2050}{10431} & -\frac{187}{2318} & \frac{113}{366} \\
\frac{414}{1159} & -\frac{29954}{31293} & \frac{130}{61} & -1 & \frac{1}{3} & 0 & -\frac{27052}{31293} & -\frac{113}{10431} & -\frac{491}{4636} & \frac{161}{732}
\end{array}\right]$$

Implementation questions

Many implementation questions are similar to those for traditional methods but there are some new challenges.
We want variable order and stepsize and it is even a realistic aim to change between stiff and non-stiff methods automatically.
Because of the variable order and stepsize aims, we wish to be able to do the following:

- Estimate the local truncation error of the current step
- Estimate the local truncation error of an alternative method of higher order
- Change the stepsize with little cost and with little impact on stability

My final comment is that all these things are possible and many of the details are already known and understood.

What is *not* known is how to choose between different choices of the free parameters.

Identifying the best methods is the first step in the construction of a competitive stiff and non-stiff solver.

References

The following recent publications contain references to earlier work:

- *Numerical Methods for Ordinary Differential Equations*, J. C. Butcher, *J. Wiley, Chichester*, (2003)
- *General linear methods*, J. C. Butcher, *Acta Numerica* **15** (2006), 157–256.

Gracias

Thank you

Obrigado