Tutorial on

Semi-Lagrangian schemes

Roberto Ferretti

"Recent advances on theory and applications of Semi-Lagrangian methods"
Roma, 05.12.11

Dipartimento di Matematica
Universita' di Roma Tre

- Some history
- Basic ideas and building blocks for SL schemes
- Convergence analysis for the linear problem
- Construction of Semi-Lagrangian schemes for convex HJ equations
- Convergence analysis for the nonlinear problem

Some history

- **Semi–Lagrangian schemes**: introduced as first–order schemes by Courant, Isaacson and Rees (CPAM, '52)

- **Numerical Weather Prediction streamline**: Wiin-Nielsen (Tellus, '59), Robert (Atmosphere-Ocean, '81), Staniforth, Côté, Smolarkiewicz...

- **Plasma physics streamline**: Cheng–Knorr ('76), Bertrand–Izzo, Besse–Mehrenberger,...

In the first developments it had not yet been realized that the possible advantage of SL schemes over conventional difference schemes was to be able to work at large Courant numbers.

This feature has become important in NWP problems, in which an orthogonal grid would have forced a conventional scheme to adopt prohibitively small time steps because of the singularity on the poles.

A further analysis shows that large Courant numbers cause the scheme to be less diffusive.

index

Basic ideas and building blocks for SL schemes

For simplicity, we will discuss SL schemes focusing on the model problem

$$\begin{cases} u_t(x,t) + f(x,t)\cdot Du(x,t) = 0, & (x,t) \in \mathbb{R}^d \times \mathbb{R} \\ u(x,0) = u_0(x) & x \in \mathbb{R}^d. \end{cases}$$

posed on the whole of $\mathbb{R}^d$.

Basic ideas and building blocks for SL schemes

For simplicity, we will discuss SL schemes focusing on the model problem

$$\begin{cases} u_t(x,t) + f(x,t) \cdot Du(x,t) = 0, & (x,t) \in \mathbb{R}^d \times \mathbb{R} \\ u(x,0) = u_0(x) & x \in \mathbb{R}^d. \end{cases}$$

posed on the whole of $\mathbb{R}^d$.

- We avoid the treatment of boundary conditions

Basic ideas and building blocks for SL schemes

For simplicity, we will discuss SL schemes focusing on the model problem

$$\begin{cases} u_t(x,t) + f(x,t) \cdot Du(x,t) = 0, & (x,t) \in \mathbb{R}^d \times \mathbb{R} \\ u(x,0) = u_0(x) & x \in \mathbb{R}^d. \end{cases}$$

posed on the whole of $\mathbb{R}^d$.

- We avoid the treatment of boundary conditions

- We treat separately and more explicitly the case of constant speed

Any large time–step technique (in particular, Semi-Lagrangian approximations) stem from the method of characteristics. Let a system of characteristic curves $y(x, t; s)$ for the model equation be defined by:

$$\begin{cases} \dfrac{d}{ds} y(x, t; s) = f(y(x, t; s), s). \\ y(x, t; t) = x, \end{cases}$$

Then, the solution is constant along such trajectories, which means that the following representation formula

$$u(y(x, t; t + \tau), t + \tau) = u(x, t).$$

holds for the solution u.

Writing the representation formula at a node (x_i, t_{n+1}) and using $\tau = -\Delta t$, we have the time-discrete version

$$u(x_i, t_{n+1}) = u(y(x_i, t_{n+1}; t_n), t_n).$$

Its numerical discretization is obtained by combining:

Writing the representation formula at a node (x_i, t_{n+1}) and using $\tau = -\Delta t$, we have the time-discrete version

$$u(x_i, t_{n+1}) = u(y(x_i, t_{n+1}; t_n), t_n).$$

Its numerical discretization is obtained by combining:

• A numerical technique to integrate backwards the ODE of characteristics

Writing the representation formula at a node (x_i, t_{n+1}) and using $\tau = -\Delta t$, we have the time-discrete version

$$u(x_i, t_{n+1}) = u(y(x_i, t_{n+1}; t_n), t_n).$$

Its numerical discretization is obtained by combining:

• A numerical technique to integrate backwards the ODE of characteristics

• A reconstruction to approximate the value $u(y(x_i, t_{n+1}; t_n), t_n)$, since in general the foot of the characteristic $y(x_i, t_{n+1}; t_n)$ does not coincide with any grid point.

Semi–Lagrangian approximation for the advection equation:

In the SL scheme, the representation formula is discretized as

$$v_i^{n+1} = I[V^n](X^{\Delta}(x_i, t_{n+1}; t_n))$$

Semi–Lagrangian approximation for the advection equation:

In the SL scheme, the representation formula is discretized as

$$v_i^{n+1} = I[V^n](X^\Delta(x_i, t_{n+1}; t_n))$$

- v_i^{n+1} is the numerical solution computed at (x_i, t_{n+1})

Semi–Lagrangian approximation for the advection equation:

In the SL scheme, the representation formula is discretized as

$$v_i^{n+1} = I[V^n](X^{\Delta}(x_i, t_{n+1}; t_n))$$

- v_i^{n+1} is the numerical solution computed at (x_i, t_{n+1})
- $X^{\Delta}(x_i, t_{n+1}; t_n)$ approximates the characteristic through (x_i, t_{n+1})

Semi–Lagrangian approximation for the advection equation:

In the SL scheme, the representation formula is discretized as

$$v_i^{n+1} = I[V^n](X^\Delta(x_i, t_{n+1}; t_n))$$

- v_i^{n+1} is the numerical solution computed at (x_i, t_{n+1})
- $X^\Delta(x_i, t_{n+1}; t_n)$ approximates the characteristic through (x_i, t_{n+1})
- $I[V^n](X^\Delta(x_i, t_{n+1}; t_n)) = \sum_j v_j^n \psi_j(X^\Delta(x_i, t_{n+1}; t_n))$ is the interpolation computed at $(X^\Delta(x_i, t_{n+1}; t_n), t_n)$

Approximation of characteristics $X^{\triangle}$: typically, by one-step or multisptep techniques

Approximation of characteristics $X^{\triangle}$: typically, by one-step or multisptep techniques

- The advection field is known (in relevant problems) only at space-time nodes

Approximation of characteristics $X^{\triangle}$: typically, by one-step or multisptep techniques

- The advection field is known (in relevant problems) only at space-time nodes

- Need to avoid intermediate times, as well as to interpolate the advecting field among space grid nodes

1st example: the explicit Euler scheme needs informations at time step t_{n+1}.

$$y(x_i, t_{n+1}; t_n) \approx X^{\Delta}(x_i, t_{n+1}; t_n) = x_i - \Delta t \; f(x_i, t_{n+1})$$

- This is the classical choice of the Courant–Isaacson–Rees scheme
- In general, it leads to a poor time approximation (1st order)
- If time step t_{n+1} is not available, it could be replaced by step t_n:

$$y(x_i, t_{n+1}; t_n) \approx X^{\Delta}(x_i, t_{n+1}; t_n) = x_i - \Delta t \; f(x_i, t_n)$$

2nd example: a second-order RK scheme only needs the times t and $t - \Delta t$, but if the vector field f is only known at the nodes, it must be interpolated.

$$X^{\Delta}(x_i, t_{n+1}; t_n) = x_i - \frac{\Delta t}{2}\left[f(x_i, t_{n+1}) + \widetilde{f}(x_i - \Delta t f(x_i, t_{n+1}), t_n)\right]$$

2nd example: a second-order RK scheme only needs the times t and $t - \Delta t$, but if the vector field f is only known at the nodes, it must be interpolated.

$$X^{\Delta}(x_i, t_{n+1}; t_n) = x_i - \frac{\Delta t}{2}\left[f(x_i, t_{n+1}) + \widetilde{f}(x_i - \Delta t f(x_i, t_{n+1}), t_n)\right]$$

- $\widetilde{f}(\xi, t_n)$ is an interpolate of the node values of $f(x_j, t_n)$, computed at the point ξ

2nd example: a second-order RK scheme only needs the times t and $t - \Delta t$, but if the vector field f is only known at the nodes, it must be interpolated.

$$X^{\Delta}(x_i, t_{n+1}; t_n) = x_i - \frac{\Delta t}{2}\left[f(x_i, t_{n+1}) + \widetilde{f}(x_i - \Delta t f(x_i, t_{n+1}), t_n)\right]$$

- $\widetilde{f}(\xi, t_n)$ is an interpolate of the node values of $f(x_j, t_n)$, computed at the point ξ

- No interpolation is needed if f is explicitly known

2nd example: a second-order RK scheme only needs the times t and $t - \Delta t$, but if the vector field f is only known at the nodes, it must be interpolated.

$$X^{\Delta}(x_i, t_{n+1}; t_n) = x_i - \frac{\Delta t}{2} \left[f(x_i, t_{n+1}) + \tilde{f}(x_i - \Delta t f(x_i, t_{n+1}), t_n) \right]$$

- $\tilde{f}(\xi, t_n)$ is an interpolate of the node values of $f(x_j, t_n)$, computed at the point ξ

- No interpolation is needed if f is explicitly known

- The approximation is second-order with respect to Δt

Numerical reconstruction of the value $u(y(x_j, t_{n+1}; t_n), t_n)$:

Linear:

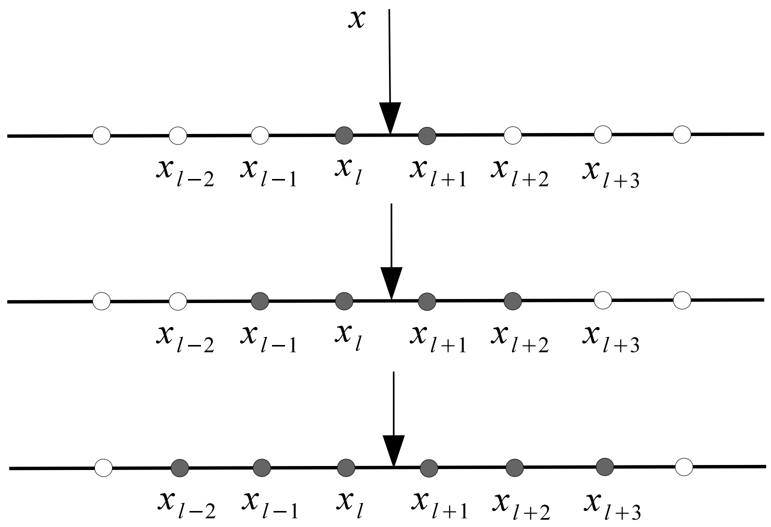
- Symmetric Lagrange interpolation (most common)

- Finite Element interpolation, cubic splines, sparse grids, Chebyshev grids,...

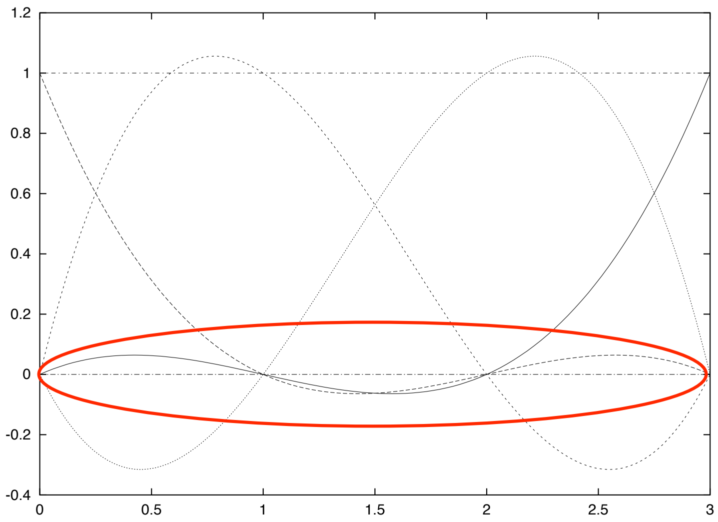
Nonlinear:

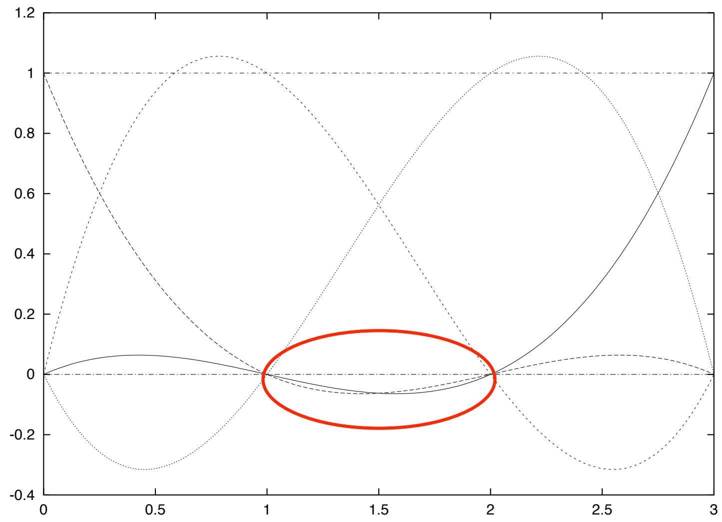
- Non-Oscillatory (ENO/WENO) interpolation, monotone Hermite interpolations,...

Symmetric Lagrange Interpolation is performed using a symmetric stencil of points around x:



stencils of interpolation (linear, cubic and quintic Lagrange)





region of interpolation ($\mathbb{P}_3$ finite elements and cubic Lagrange)

SLI may be expressed in a set of basis functions as

$$I[V](x) = \sum_j v_j \psi_j(x)$$

SLI may be expressed in a set of basis functions as

$$I[V](x) = \sum_j v_j \psi_j(x)$$

• The basis function ψ_j is obtained by interpolating the sequence e_j, i.e., a sequence which is everywhere zero except at the node x_j

SLI may be expressed in a set of basis functions as

$$I[V](x) = \sum_j v_j \psi_j(x)$$

• The basis function ψ_j is obtained by interpolating the sequence e_j, i.e., a sequence which is everywhere zero except at the node x_j

• On a uniform grid, a basis function ψ_j can be written in terms of a reference basis function ψ:

$$\psi_j(\xi) = \psi\left(\frac{\xi}{\Delta x} - j\right)$$

(obtained reconstructing e_0 on a grid with $\Delta x = 1$)

When this procedure is applied to a Lagrange reconstruction of odd order r, the reference basis function has the form:

$$\psi(\xi) = \begin{cases} \displaystyle\prod_{k \neq 0, k=-[r/2]}^{[r/2]+1} \frac{\xi - k}{-k} & \text{if } 0 \leq \xi \leq 1 \\ \vdots & \vdots \\ \displaystyle\prod_{k=1}^{r} \frac{\xi - k}{-k} & \text{if } [r/2] \leq \xi \leq [r/2]+1 \\ 0 & \text{if } \xi > [r/2]+1 \end{cases}$$

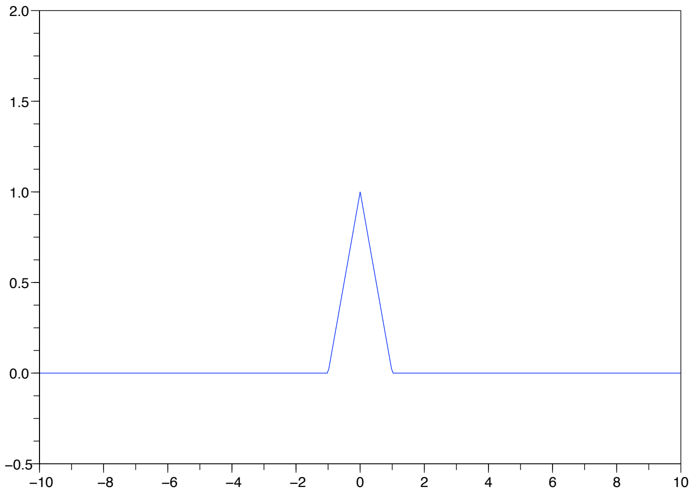
and extended by symmetry for $\xi < 0$.

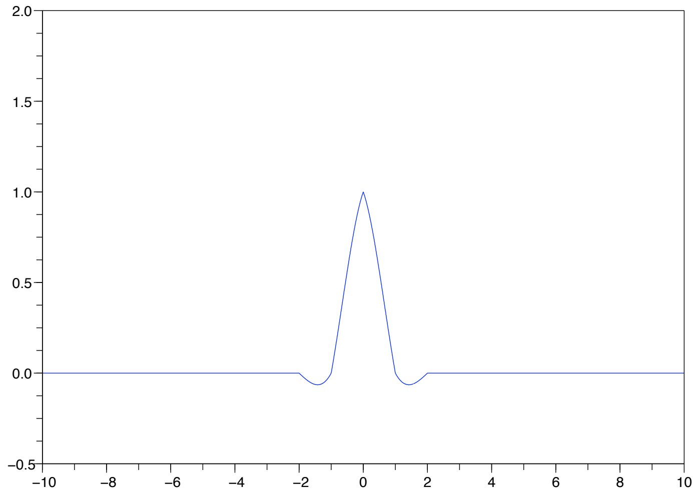
When this procedure is applied to a Lagrange reconstruction of odd order r, the reference basis function has the form:

$$\psi(\xi) = \begin{cases} \displaystyle\prod_{k\neq 0, k=-[r/2]}^{[r/2]+1} \frac{\xi - k}{-k} & \text{if } 0 \leq \xi \leq 1 \\ \vdots & \vdots \\ \displaystyle\prod_{k=1}^{r} \frac{\xi - k}{-k} & \text{if } [r/2] \leq \xi \leq [r/2]+1 \\ 0 & \text{if } \xi > [r/2]+1 \end{cases}$$

and extended by symmetry for $\xi < 0$.

- The interpolation error is $O(\Delta x^{r+1})$ for smooth functions





The reference basis functions ψ for $\mathbb{P}_1$ and cubic interpolation

index

Convergence analysis for the linear problem

To prove **consistency**, we need to compare the scheme:

$$v_i^{n+1} = I[V^n](X^{\Delta}(x_i, t_{n+1}; t_n))$$

with the representation formula:

$$u(x_i, t_{n+1}) = u(y(x_i, t_{n+1}; t_n), t_n).$$

assuming that u is a smooth solution and that $v_j^n = u(x_j, t_n)$.

We also assume to have a general approximation of order p in time and r in space

It turns out that the local truncation error is estimated as:

$$\left| L^{\Delta}(x_i, t_{n+1}) \right| \leq C \left(\Delta t^p + \frac{\Delta x^{r+1}}{\Delta t} \right)$$

It turns out that the local truncation error is estimated as:

$$\left|L^{\Delta}(x_i, t_{n+1})\right| \leq C\left(\Delta t^p + \frac{\Delta x^{r+1}}{\Delta t}\right)$$

• The term Δt^p accounts for the error in the computation of characteristics

It turns out that the local truncation error is estimated as:

$$\left| L^{\Delta}(x_i, t_{n+1}) \right| \leq C \left(\Delta t^p + \frac{\Delta x^{r+1}}{\Delta t} \right)$$

- The term Δt^p accounts for the error in the computation of characteristics

- The term $\frac{\Delta x^{r+1}}{\Delta t}$ accounts for the error generated by the accumulation of interpolation errors

It turns out that the local truncation error is estimated as:

$$\left|L^{\Delta}(x_i, t_{n+1})\right| \leq C\left(\Delta t^p + \frac{\Delta x^{r+1}}{\Delta t}\right)$$

- The term Δt^p accounts for the error in the computation of characteristics

- The term $\frac{\Delta x^{r+1}}{\Delta t}$ accounts for the error generated by the accumulation of interpolation errors

- There exists an optimal $\Delta x/\Delta t$ balance which maximizes the consistency rate

A sharper estimate of the local truncation error might be obtained by taking into account the space dependence of the interpolation error. If $|x - x_i| = O(\Delta t)$, then

$$|u(x) - I[U](x)| \leq C \min\left(\Delta x^{r+1}, \Delta t \Delta x^r\right).$$

As a result, the consistency error takes the form

$$\left|L^{\Delta}(x_i, t_{n+1})\right| \leq C \left(\Delta t^p + \min\left(\frac{\Delta x^{r+1}}{\Delta t}, \Delta x^r\right)\right)$$

• The best estimate uses Δx^r for small Courant numbers, $\Delta x^{r+1}/\Delta t$ for large Courant numbers.

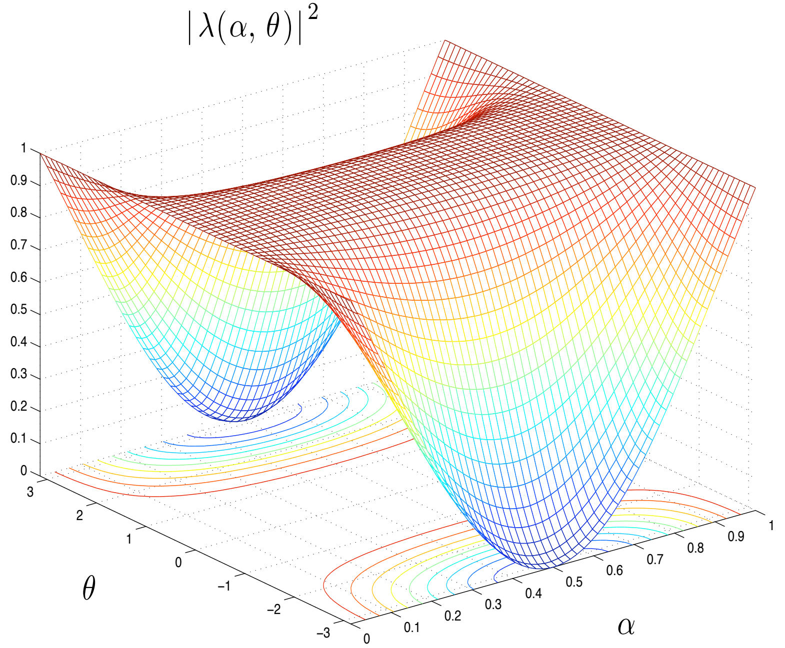
To prove **stability**, we restrict for simplicity to the equation in the constant coefficient form:

$$u_t + cu_x = 0.$$

Here, we have assumed that the advection has constant speed c, so that $X^{\Delta}(x_i, t_{n+1}; t_n) = x_i - c\Delta t$ and the SL scheme has the form

$$v_i^{n+1} = I[V^n](x_i - c\Delta t).$$

We are in the typical framework of Von Neumann analysis, and in fact it is possible to prove by Fourier analysis arguments that the scheme is stable.



Amplitude of the amplification factors λ for cubic interpolation

We will rather follow the line of proving stability by equivalence with a stable scheme, in this case the Lagrange–Galerkin scheme which has the form:

$$\int_{\mathbb{R}} v_{\Delta}^{n+1}(\xi)\phi_i(\xi)d\xi = \int_{\mathbb{R}} v_{\Delta}^{n}(\xi - c\Delta t)\phi_i(\xi)d\xi$$

that is, writing the numerical solution as $v_{\Delta}^k(x) = \sum_j v_j^k \phi_j(x)$,

$$\sum_j v_j^{n+1} \int_{\mathbb{R}} \phi_j(\xi)\phi_i(\xi)d\xi = \sum_j v_j^n \int_{\mathbb{R}} \phi_j(x_i - c\Delta t)\phi_i(\xi)d\xi$$

We will rather follow the line of proving stability by equivalence with a stable scheme, in this case the Lagrange–Galerkin scheme which has the form:

$$\int_{\mathbb{R}} v_\Delta^{n+1}(\xi)\phi_i(\xi)d\xi = \int_{\mathbb{R}} v_\Delta^n(\xi - c\Delta t)\phi_i(\xi)d\xi$$

that is, writing the numerical solution as $v_\Delta^k(x) = \sum_j v_j^k \phi_j(x)$,

$$\sum_j v_j^{n+1} \int_{\mathbb{R}} \phi_j(\xi)\phi_i(\xi)d\xi = \sum_j v_j^n \int_{\mathbb{R}} \phi_j(x_i - c\Delta t)\phi_i(\xi)d\xi$$

• In the LG scheme, interpolation is replaced by Galerkin projection

We will rather follow the line of proving stability by equivalence with a stable scheme, in this case the Lagrange–Galerkin scheme which has the form:

$$\int_{\mathbb{R}} v_{\Delta}^{n+1}(\xi)\phi_i(\xi)d\xi = \int_{\mathbb{R}} v_{\Delta}^{n}(\xi - c\Delta t)\phi_i(\xi)d\xi$$

that is, writing the numerical solution as $v_{\Delta}^{k}(x) = \sum_j v_j^k \phi_j(x)$,

$$\sum_j v_j^{n+1} \int_{\mathbb{R}} \phi_j(\xi)\phi_i(\xi)d\xi = \sum_j v_j^{n} \int_{\mathbb{R}} \phi_j(x_i - c\Delta t)\phi_i(\xi)d\xi$$

- In the LG scheme, interpolation is replaced by Galerkin projection

- As a consequence, $\|v_{\Delta}^{n+1}\|_2 \le \|v_{\Delta}^{n}\|_2$ (i.e., the scheme is stable)

The Galerkin basis is supposed to have a structure similar to the SL basis:

$$\phi_j(\xi) = \frac{1}{\sqrt{\Delta x}}\,\phi\left(\frac{\xi}{\Delta x} - j\right)$$

where ϕ is the reference LG basis function, and the factor $\frac{1}{\sqrt{\Delta x}}$ gives the correct scaling in the integration

The Galerkin basis is supposed to have a structure similar to the SL basis:

$$\phi_j(\xi) = \frac{1}{\sqrt{\Delta x}} \phi\left(\frac{\xi}{\Delta x} - j\right)$$

where ϕ is the reference LG basis function, and the factor $\frac{1}{\sqrt{\Delta x}}$ gives the correct scaling in the integration

• The condition of equivalence between SL and LG schemes relates the reference functions ϕ and ψ with integral equation:

$$\int_{\mathbb{R}} \phi(\eta + t)\phi(\eta)d\eta = \psi(t)$$

that is, ϕ must have ψ as its autocorrelation

This problem has a solution (in general, nonunique) if and only if:

- The function ψ is positive definite, that is

$$\sum_{k=1}^{n} \sum_{j=1}^{n} a_k \psi(t_k - t_j) \bar{a}_j \geq 0$$

for any $t_k \in \mathbb{R}$, $a_k \in \mathbb{C}$ $(k = 1, \ldots, n)$ and for all $n \in \mathbb{N}$

- Equivalently, the function ψ has a real positive Fourier transform $\widehat{\psi}$

This problem has a solution (in general, nonunique) if and only if:

- The function ψ is positive definite, that is

$$\sum_{k=1}^{n} \sum_{j=1}^{n} a_k \psi(t_k - t_j) \bar{a}_j \geq 0$$

for any $t_k \in \mathbb{R}$, $a_k \in \mathbb{C}$ $(k = 1, \ldots, n)$ and for all $n \in \mathbb{N}$

- Equivalently, the function ψ has a real positive Fourier transform $\widehat{\psi}$

- The solution is given by $\phi(t) = \mathcal{F}^{-1}\left\{\widehat{\psi}(\omega)^{1/2}\right\}$

This problem has a solution (in general, nonunique) if and only if:

- The function ψ is positive definite, that is

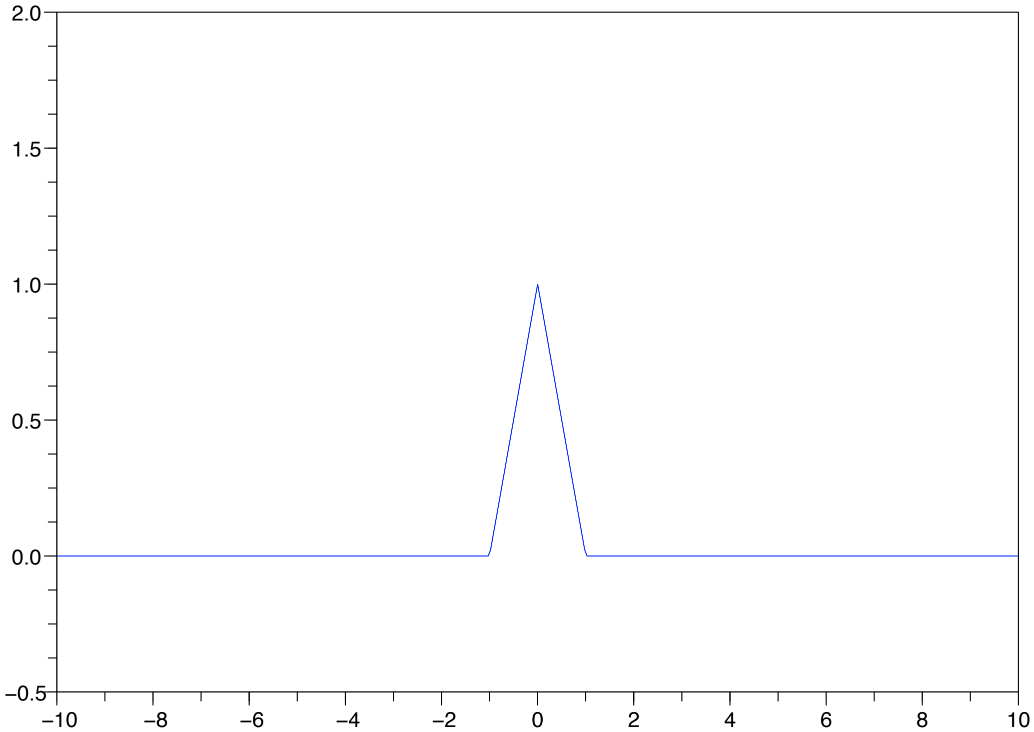
$$\sum_{k=1}^{n} \sum_{j=1}^{n} a_k \psi(t_k - t_j) \bar{a}_j \geq 0$$

for any $t_k \in \mathbb{R}$, $a_k \in \mathbb{C}$ $(k = 1, \ldots, n)$ and for all $n \in \mathbb{N}$

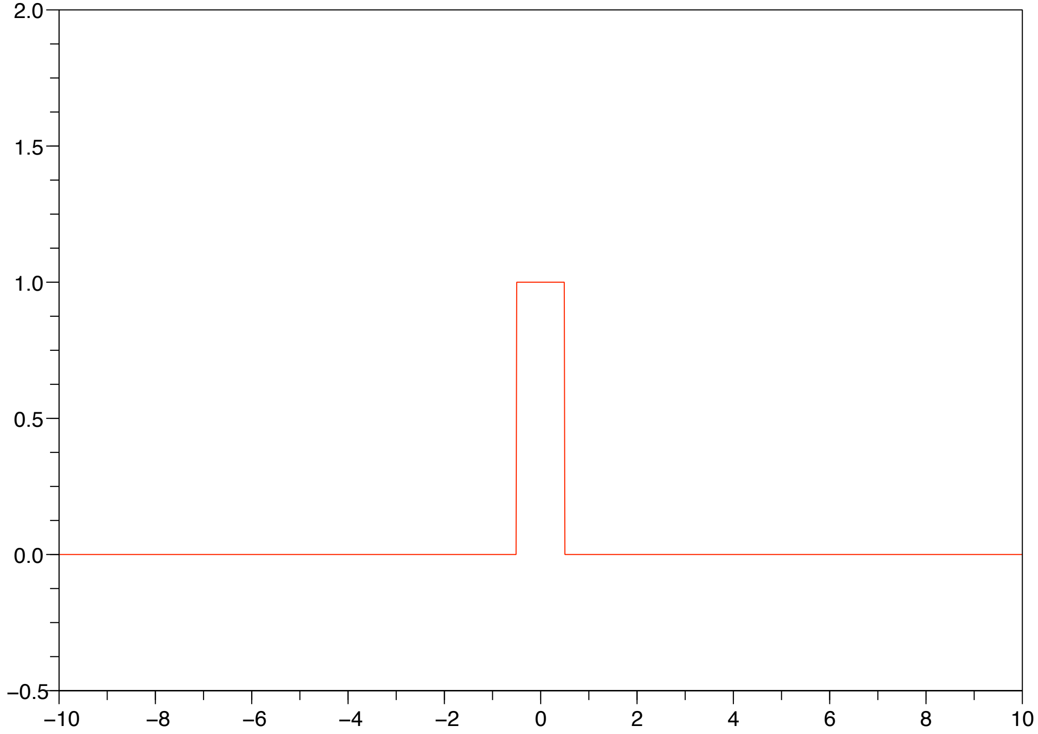
- Equivalently, the function ψ has a real positive Fourier transform $\widehat{\psi}$

- The solution is given by $\phi(t) = \mathcal{F}^{-1}\left\{\widehat{\psi}(\omega)^{1/2}\right\}$

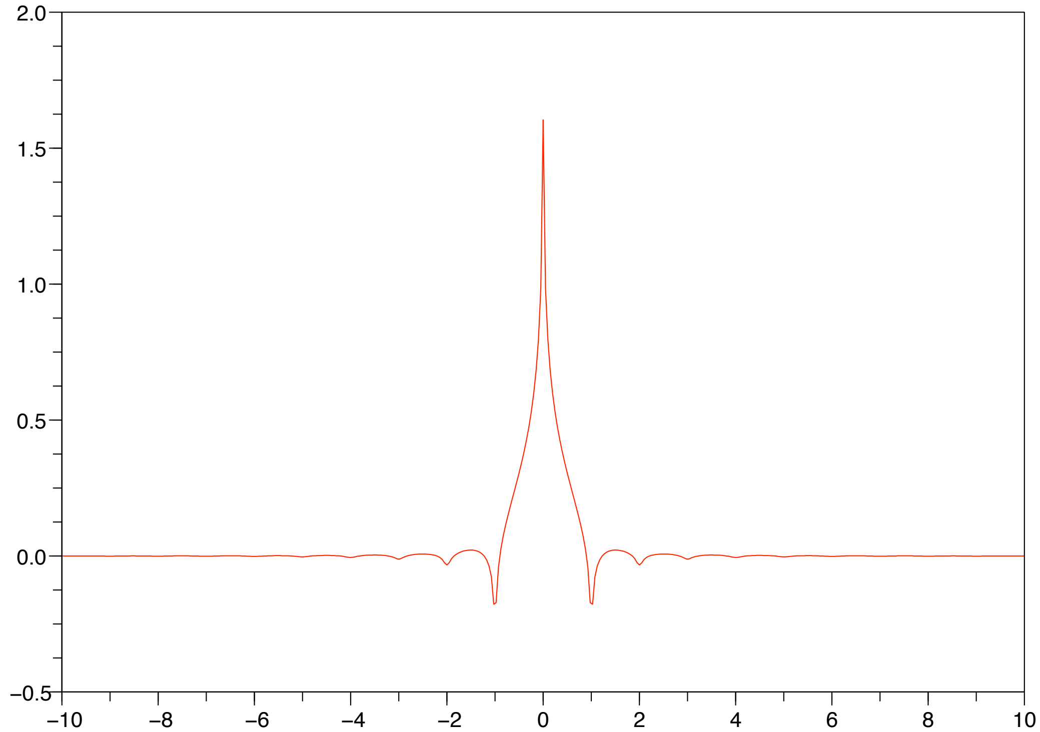
- Existence of a solution implies L^2 stability of SL schemes



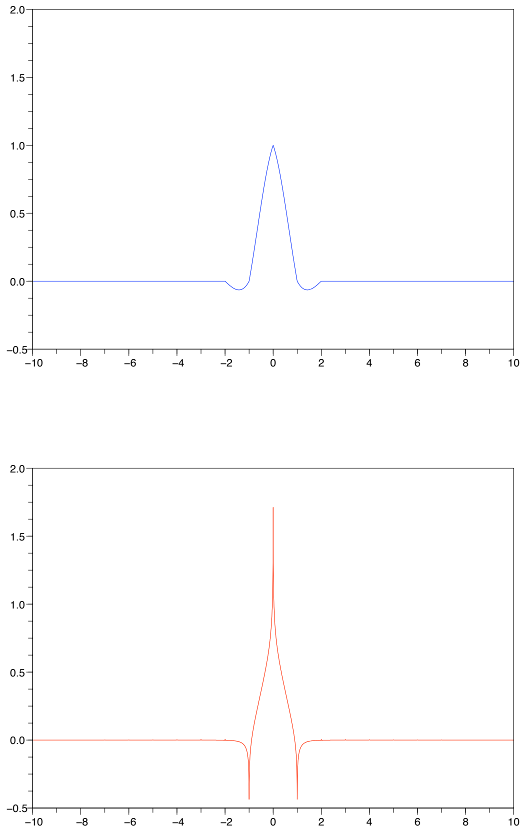
The reference function ψ for $\mathbb{P}_1$ interpolation



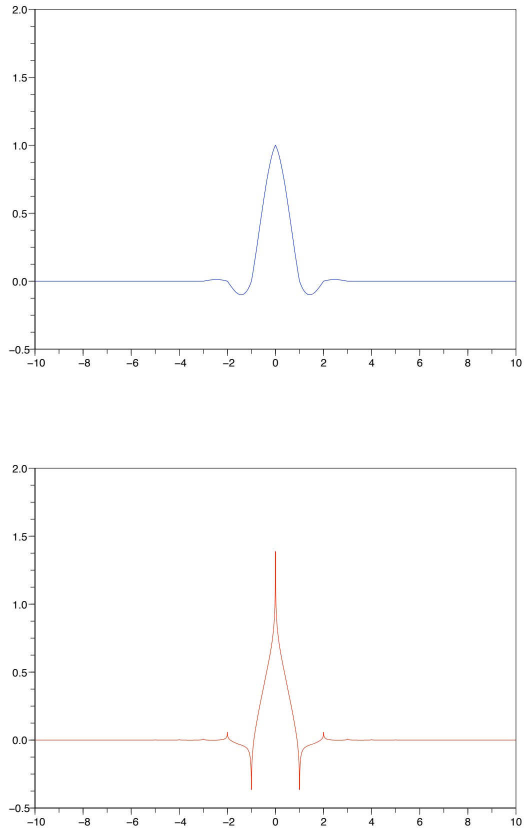
The "obvious" LG counterpart ϕ for $\mathbb{P}_1$ interpolation



The minimal phase LG counterpart ϕ for $\mathbb{P}_1$ interpolation



SL and LG, cubic



SL and LG, quintic

Situations covered by this result:

• High–order Lagrange interpolations which can be shown to have a positive Fourier transform (tested for $n \leq 13$):

$$\widehat{\psi}^{(n)}(\omega) = p(\omega^2)\, \frac{\sin\left(\frac{\omega}{2}\right)^{n+1}}{\left(\frac{\omega}{2}\right)^{n+1}}$$

with $p(\omega^2)$ a polynomial of degree $[n/2]$ with positive coefficients.

• Interpolatory wavelets, usually defined to be positive definite functions (e.g., in the case of the Shannon wavelet, $\hat{\psi}(\omega) = 1_{(-\pi,\pi)}(\omega)$).

• Cubic splines (no rigorous proof)

• In general, (as for the case of the $\mathbb{P}_1$ base) we expect to have multiple solutions to the problem: in fact, the relationship between $\widehat{\phi}$ and $\widehat{\psi}$,

$$|\widehat{\phi}(\omega)|^2 = \widehat{\psi}(\omega)$$

poses no constraint on the phase of $\widehat{\phi}$

• In general, (as for the case of the $\mathbb{P}_1$ base) we expect to have multiple solutions to the problem: in fact, the relationship between $\widehat{\phi}$ and $\widehat{\psi}$,

$$|\widehat{\phi}(\omega)|^2 = \widehat{\psi}(\omega)$$

poses no constraint on the phase of $\widehat{\phi}$

• The possibility to generate solutions with different phase terms is a tool to select a solution with prescribed decay and/or smoothness requirements (a key tool to treat the variable coefficient case)

index

Construction of Semi-Lagrangian schemes for convex HJ equations

Concerning HJ equations, we refer to the model problem:

$$\begin{cases} u_t(x,t) + H(Du(x,t)) = 0, & (x,t) \in \mathbb{R}^d \times [0,T] \\ u(x,0) = u_0(x) & x \in \mathbb{R}^d. \end{cases}$$

- Typical assumptions on $H(p)$: smoothness, convexity, coercivity (e.g., a lower bound on H_{pp})

Construction of Semi-Lagrangian schemes for convex HJ equations

Concerning HJ equations, we refer to the model problem:

$$\begin{cases} u_t(x,t) + H(Du(x,t)) = 0, & (x,t) \in \mathbb{R}^d \times [0,T] \\ u(x,0) = u_0(x) & x \in \mathbb{R}^d. \end{cases}$$

• Typical assumptions on $H(p)$: smoothness, convexity, coercivity (e.g., a lower bound on H_{pp})

• Various extensions (in particular, to Dynamic Programming Equations) are possible

The representation formula which parallels the formula of characteristics for HJ equations, is termed as the *Hopf–Lax formula*:

$$u(x, t + \tau) = \min_{a \in \mathbb{R}^d} [\tau H^*(a) + u(x - a\tau, t)]$$

where

$$H^*(a) = \sup_{p \in \mathbb{R}^d} [a \cdot p - H(p)]$$

is the Legendre transform of the Hamiltonian function H.

The representation formula which parallels the formula of characteristics for HJ equations, is termed as the *Hopf–Lax formula*:

$$u(x, t + \tau) = \min_{a \in \mathbb{R}^d} [\tau H^*(a) + u(x - a\tau, t)]$$

where

$$H^*(a) = \sup_{p \in \mathbb{R}^d} [a \cdot p - H(p)]$$

is the Legendre transform of the Hamiltonian function H.

Via the Hopf–Lax formula it can also be shown that the typical regularity achieved by the solution u is semiconcavity (roughly speaking, a unilateral upper bound on the second incremental ratio).

Semi–Lagrangian approximation for the convex HJ equation:

The Hopf–Lax representation formula is discretized as

$$v_i^{n+1} = \min_{\alpha \in \mathbb{R}^d} [\Delta t H^*(\alpha) + I[V^n](x_i - \alpha \Delta t)].$$

In addition to the reconstruction operator $I[V^n]$, two new ingredients are required:

Semi–Lagrangian approximation for the convex HJ equation:

The Hopf–Lax representation formula is discretized as

$$v_i^{n+1} = \min_{\alpha \in \mathbb{R}^d} [\Delta t H^*(\alpha) + I[V^n](x_i - \alpha \Delta t)].$$

In addition to the reconstruction operator $I[V^n]$, two new ingredients are required:

• A numerical approximation of the Legendre transform $H^*(\alpha)$ (whenever it cannot be explicitly computed)

Semi–Lagrangian approximation for the convex HJ equation:

The Hopf–Lax representation formula is discretized as

$$v_i^{n+1} = \min_{\alpha \in \mathbb{R}^d} [\Delta t H^*(\alpha) + I[V^n](x_i - \alpha \Delta t)].$$

In addition to the reconstruction operator $I[V^n]$, two new ingredients are required:

- A numerical approximation of the Legendre transform $H^*(\alpha)$ (whenever it cannot be explicitly computed)

- A derivative-free minimization procedure

index

Convergence analysis for the nonlinear problem

Beside consistency, two main concepts of stability are available for proving convergence in the nonlinear case:

Convergence analysis for the nonlinear problem

Beside consistency, two main concepts of stability are available for proving convergence in the nonlinear case:

- Barles–Souganidis theorem: the scheme should be invariant for the addition of constants, and *monotone up to a term* $o(\Delta t)$

Convergence analysis for the nonlinear problem

Beside consistency, two main concepts of stability are available for proving convergence in the nonlinear case:

• Barles–Souganidis theorem: the scheme should be invariant for the addition of constants, and *monotone up to a term* $o(\Delta t)$

• Lin–Tadmor theorem: the numerical solutions should be uniformly semiconcave

To prove **consistency**, at least in the sense of Barles–Souganidis, we compare again the scheme with the Hopf–Lax representation formula, assuming that u is a smooth solution and that $v_j^n = u(x_j, t_n)$. It results that the local truncation error has the estimate:

$$\left| L^{\Delta}(x_i, t_{n+1}) \right| \leq C \frac{\Delta x^{r+1}}{\Delta t}$$

To prove **consistency**, at least in the sense of Barles–Souganidis, we compare again the scheme with the Hopf–Lax representation formula, assuming that u is a smooth solution and that $v_j^n = u(x_j, t_n)$. It results that the local truncation error has the estimate:

$$\left| L^{\Delta}(x_i, t_{n+1}) \right| \leq C \frac{\Delta x^{r+1}}{\Delta t}$$

• A time discretization term $O(\Delta t^p)$ appears again as soon as characteristics are no longer straight lines

To prove **consistency**, at least in the sense of Barles–Souganidis, we compare again the scheme with the Hopf–Lax representation formula, assuming that u is a smooth solution and that $v_j^n = u(x_j, t_n)$. It results that the local truncation error has the estimate:

$$\left| L^{\Delta}(x_i, t_{n+1}) \right| \leq C \frac{\Delta x^{r+1}}{\Delta t}$$

• A time discretization term $O(\Delta t^p)$ appears again as soon as characteristics are no longer straight lines

• Consistency analysis is more technical in the Lin–Tadmor theory, although it comes to similar conclusions

Proving **monotonicity** is trivial for the first-order scheme. Monotonicity up to an $o(\Delta t)$ is possible even for high-order reconstructions provided:

Proving **monotonicity** is trivial for the first-order scheme. Monotonicity up to an $o(\Delta t)$ is possible even for high-order reconstructions provided:

- The numerical solutions are Lipschitz stable, so that the reconstruction satisfies monotonicity up to an $O(\Delta x)$

Proving **monotonicity** is trivial for the first-order scheme. Monotonicity up to an $o(\Delta t)$ is possible even for high-order reconstructions provided:

• The numerical solutions are Lipschitz stable, so that the reconstruction satisfies monotonicity up to an $O(\Delta x)$

• The Courant number goes to infinity: $\Delta x = o(\Delta t)$ – here, the SL schemes have some more degrees of freedom in choosing the $\Delta t/\Delta x$ relationship

Lipschitz stability result: Consider the scheme in $\mathbb{R}$

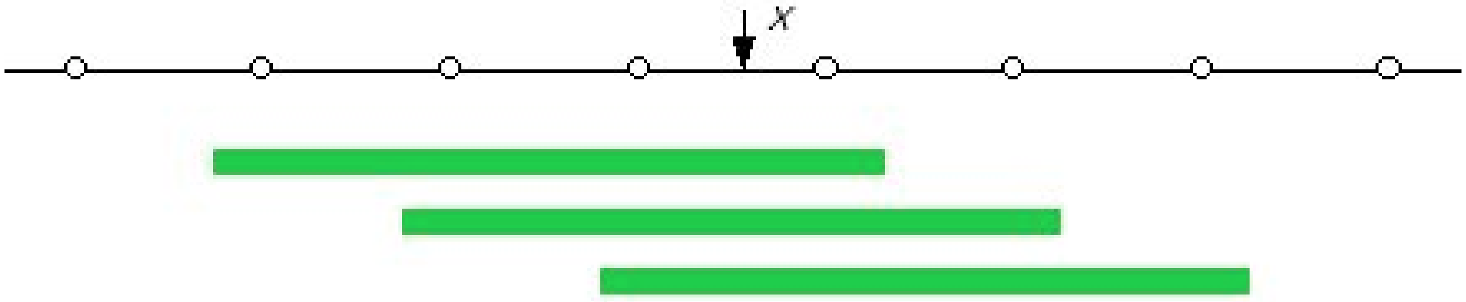
$$v_i^{n+1} = \min_{\alpha \in \mathbb{R}} [\Delta t H^*(a) + I_r[V^n](x_i - \alpha \Delta t)]$$

for a Hamiltonian function $H(p)$ such that $H_{pp} \geq m_H$. Assume that, for some constant $C < 1$:

$$|I_r[V](x) - I_1[V](x)| \leq C \max_{x_{j-1}, x_j, x_{j+1} \in \mathcal{S}(x)} |v_{j+1} - 2v_j + v_{j-1}|$$

(I_1 denoting the $\mathbb{P}_1$ interpolation, and $\mathcal{S}(x)$ denoting the reconstruction stencil at x) and that $\Delta x = O(\Delta t^2)$. Then, the family of numerical solutions V^n is Lipschitz stable.

Admissible reconstructions: the previous condition is satisfied for Lagrange reconstructions up to degree 5, provided the reconstruction stencil overlaps with the cell in which the reconstruction is performed.



admissible stencils

• Consequences: Lipschitz stability holds for ENO and finite element reconstructions up to degree 5

• Consequences: Lipschitz stability holds for ENO and finite element reconstructions up to degree 5

• Furthermore, the case of symmetric Lagrange or WENO reconstructions can be treated by proving that (linear) weights of WENO interpolation are nonnegative. This gives Lipschitz stability up to degree 5/9 for WENO and up to degree 9 for symmetric Lagrange.

• Consequences: Lipschitz stability holds for ENO and finite element reconstructions up to degree 5

• Furthermore, the case of symmetric Lagrange or WENO reconstructions can be treated by proving that (linear) weights of WENO interpolation are nonnegative. This gives Lipschitz stability up to degree 5/9 for WENO and up to degree 9 for symmetric Lagrange.

• In the practical use of the SL scheme, the condition $\Delta x = O(\Delta t^2)$ seems overly restrictive.