

Advanced Topics in High-Order Numerical Methods for Convection-Dominated Problems

BY CHI-WANG SHU

Syllabus:

1. Hamilton-Jacobi
 - a. lectures
 - b. (steady) fast-sweeping
 - c. level-set method
2. Boltzmann type equations
 - a. lectures
 - b. finite difference
 - c. splitting method (+characteristic/Lagrange type methods)
 - d. spectral methods
3. Semiconductor Moment Models in semiconductor device simulations
 - a. lectures
 - b. drift-diffusion
 - c. energy transport models
 - d. hydrodynamic models
 - e. quantum {hydrodynamics, drift-diffusion}

Table of contents

1 Hamilton-Jacobi Equations	2
1.1 Finite-Difference Method	2
1.2 Two-dimensional Extensions	4
1.2.1 Lax-Friedrichs Hamiltonian	4
1.2.2 Gudonov Hamiltonian	5
1.2.3 Osher-Sethian Hamiltonian	5
1.2.4 Unstructured Grids	6
1.3 High-Order Methods	6
1.4 DG for Hamilton-Jacobi	7
1.5 Proving Error Estimates	9
1.6 Level-Set Methods	9
2 Boltzmann-Type Equations	11
3 Semiconductor Device Simulation Models	12
3.1 Drift-diffusion model	12
3.2 High field model	13
3.3 Momentum solvers	13
3.4 Energy Transport Models	14
3.5 Hydrodynamic Model	14
3.6 Kinetic Model	14

3.7 Quantum Effects	15
-------------------------------	----

Typed by Akil Narayan, Andreas Klöckner <{anaray, kloeckner}@dam.brown.edu>.

1 Hamilton-Jacobi Equations

One-dimensional Hamilton-Jacobi equations look like this:

$$\varphi_t + H(\varphi_x) = 0. \quad (1)$$

H is the *Hamiltonian* (nonlinear). For comparison, hyperbolic *conservation laws* look like this:

$$u_t + f(u)_x = 0. \quad (2)$$

Note that formally, if the H (or f) is linear, there's no difference between the two types of equations. Some differences remain however from the choice of function spaces where solutions are sought. For Hamilton-Jacobi Equations, our solutions will be Lipschitz-continuous.

Feature	Hamilton-Jacobi	Conservation Laws
Function Space	Lipschitz	Discontinuous/BV
Generalized solution	Viscosity solution	Entropy solution

Table 1. Comparison of Hamilton-Jacobi and Conservation Laws.

There's extra horridness waiting for people who have time for high-risk research: Equations like

$$\varphi_t + H(\varphi_x) + f(\varphi)_x = 0$$

also occur in applications.

If we set $u = \varphi_x$, then a Hamilton-Jacobi equation (1) becomes $u_t + H(u)_x = 0$.

Definition 1. φ is called a viscosity subsolution of (1) if, for any smooth function ψ at each local maximum point (\bar{x}, \bar{t}) of $\varphi - \psi$, we have

$$\psi_t(\bar{x}, \bar{t}) + H(\psi_x(\bar{x}, \bar{t})) \leq 0.$$

Definition 2. φ is called a viscosity supersolution of (1) if, for any smooth function ψ at each local minimum point (\bar{x}, \bar{t}) of $\varphi - \psi$, we have

$$\psi_t(\bar{x}, \bar{t}) + H(\psi_x(\bar{x}, \bar{t})) \geq 0.$$

Definition 3. φ is called a viscosity solution if it is both a viscosity sub- and supersolution.

Deeper mathematical insight can be obtained from papers by Crandall, Lions, and Souganidis. There is an L^∞ -contraction result that supplies well-posedness of the sort

$$\|\varphi_1 - \varphi_2\|_{L^\infty} \leq \|\varphi_1(\cdot, 0) - \varphi_2(\cdot, 0)\|_{L^\infty}.$$

1.1 Finite-Difference Method

Assume we have a few grid points $\dots < x_0 < x_1 < x_2 < \dots$ and also assume we approximate $\varphi_j \approx \varphi(x_j, t)$. We briefly recall that a correct discretization of a conservation law (2) depended crucially on the knowledge of the present “wind direction”. This would determine the choice between

$$\begin{aligned} f(u)_x|_{x=x_j} &\approx \frac{f(u_j) - f(u_{j-1})}{\Delta x} & f'(u) > 0, \\ f(u)_x|_{x=x_j} &\approx \frac{f(u_{j+1}) - f(u_j)}{\Delta x} & f'(u) < 0. \end{aligned}$$

This led to the introduction of a *numerical flux* to cope with the possibility of $f'(u)$ changing sign.

$$f(u)_x|_{x=x_j} \approx \frac{\hat{f}(u_j, u_{j+1}) - \hat{f}(u_{j-1}, u_j)}{\Delta x},$$

with $\hat{f}(u^-, u^+)$ satisfying:

- $\hat{f}(\cdot, \cdot)$ is Lipschitz-continuous.
- Consistency: $\hat{f}(u, u) = f(u)$.
- Monotonicity: $\hat{f}(\uparrow, \downarrow)$.

So for Hamilton-Jacobi we simply set

$$\begin{aligned} H(\varphi_x|_{x=x_j}) &\approx H\left(\frac{\varphi_j - \varphi_{j-1}}{\Delta x}\right) & H'(u) > 0, \\ H(\varphi_x|_{x=x_j}) &\approx H\left(\frac{\varphi_{j+1} - \varphi_j}{\Delta x}\right) & H'(u) < 0. \end{aligned}$$

The generalization to changing signs of $H'(u)$ becomes

$$\hat{H}\left(\frac{\varphi_j - \varphi_{j-1}}{\Delta x}, \frac{\varphi_{j+1} - \varphi_j}{\Delta x}\right).$$

The scheme we come up with reads

$$(\varphi_j)_t + \hat{H}\left(\frac{\varphi_j - \varphi_{j-1}}{\Delta x}, \frac{\varphi_{j+1} - \varphi_j}{\Delta x}\right) = 0,$$

where we again assume that $\hat{H}(u^-, u^+)$ satisfying:

- $\hat{H}(\cdot, \cdot)$ is Lipschitz-continuous.
- Consistency: $\hat{H}(u, u) = f(u)$.
- Monotonicity: $\hat{H}(\uparrow, \downarrow)$.

Theorem 4. (Crandall, Lions) *Solutions of monotone schemes converge to the viscosity solution.*

$$\|\varphi - \varphi_{\Delta x}\|_{L^\infty} \leq C\sqrt{\Delta x}.$$

[Begin Lecture October 3, 2007]

Recall the scheme

$$(\phi_i)_t + \hat{H}\left(\frac{\phi_i - \phi_{i-1}}{\Delta x}, \frac{\phi_{i+1} - \phi_i}{\Delta x}\right) = 0 \tag{3}$$

where \hat{H} is a monotone numerical Hamiltonian (*Monotonian*):

- \hat{H} is a Lipschitz continuous function with respect to both arguments
- $\hat{H}(\uparrow, \downarrow)$ (increasing in first argument, decreasing in second)
- $\hat{H}(u, u) = H(u)$ (consistency)

Any monotone flux studied in conservation laws can be used for \hat{H} , e.g. a Gudonov-type flux:

$$H(u^-, u^+) = \begin{cases} \min_{u^- \leq u \leq u^+} H(u) & \text{if } u^- \leq u^+ \\ \max_{u^- \leq u \leq u^+} H(u) & \text{if } u^- > u^+ \end{cases}$$

And we can also use Engquist-Osher, or Lax-Friedrichs:

$$H^{\text{LF}}(u^-, u^+) = H\left(\frac{u^- + u^+}{2}\right) - \frac{\alpha}{2}(u^+ - u^-) \tag{4}$$

where $\alpha = \max_u |H'(u)|$, or another ‘Lax-Friedrichs’ flux:

$$H(u^-, u^+) = \frac{1}{2} [H(u^-) + H(u^+) - \alpha(u^+ - u^-)] \quad (5)$$

Theorem 5. *A monotone scheme is convergent to the viscosity solution*

$$\|\phi_{\Delta x} - \phi\|_{L^\infty} \leq C\sqrt{\Delta x} \quad (6)$$

1.2 Two-dimensional Extensions

The two-dimensional Hamilton-Jacobi equation takes the form

$$\phi_t + H(\phi_x, \phi_y) = 0 \quad (7)$$

Note that all of the above has only been developed for equation in one space dimension, equation (1). Recall that one dimension is special: HJ equations in one dimension are easily recast into 1D hyperbolic conservation laws, for which all of the above analysis is already well-known. We can attempt to do a similar thing in 2D: from equation (7), let $u = \phi_x$, $v = \phi_y$. Then equation (7) yields the system

$$\begin{cases} u_t + H(u, v)_x = 0 \\ v_t + H(u, v)_y = 0 \end{cases} \quad (8)$$

Thus our scalar equation has yielded an undesirable system of equations. We can recast (8) as

$$\begin{pmatrix} u \\ v \end{pmatrix}_t + \begin{pmatrix} H(u, v) \\ 0 \end{pmatrix}_x + \begin{pmatrix} 0 \\ H(u, v) \end{pmatrix}_y = 0 \quad (9)$$

We can define $f(u)$ and $g(u)$ so that the second and third terms read $f(u)_x$ and $g(u)_y$, respectively, and write Jacobians as

$$f'(\vec{u}) = \begin{pmatrix} H_u & H_v \\ 0 & 0 \end{pmatrix} \quad g'(\vec{u}) = \begin{pmatrix} 0 & 0 \\ H_u & H_v \end{pmatrix}$$

Now the eigenvalues corresponding to these Jacobians are H_u and 0. If $H_u \neq 0$, then everything is fine, we have a well-posed hyperbolic system. However, if $H_u = 0$, then we have a weakly well-posed system, and thus we must now deal with solving a weakly well-posed problem, which is undesirable.

We avoid this and deal directly with our scalar equation in two dimension, let us introduce a ‘monotone Hamiltonian’, $\hat{H}(u^-, u^+; v^-, v^+)$, which we ask to satisfy

- \hat{H} is Lipschitz in all arguments
- $\hat{H}(\uparrow, \downarrow; \uparrow, \downarrow)$
- $\hat{H}(u, u; v, v) = H(u, v)$

and we introduce the semi-discrete ‘monotone scheme’ for equation (7)

$$(\phi_{ij})_t + \hat{H}\left(\frac{\phi_{i,j} - \phi_{i-1,j}}{\Delta x}, \frac{\phi_{i+1,j} - \phi_{i,j}}{\Delta x}, \frac{\phi_{i,j} - \phi_{i,j-1}}{\Delta x}, \frac{\phi_{i,j+1} - \phi_{i,j}}{\Delta x}\right) = 0 \quad (10)$$

then we have the same result for 2D as is stated in theorem 5. Note that the introduction of the relatively complex restrictions on our monotone Hamiltonians is necessary: because the Hamiltonian is possibly (probably) nonlinear, we cannot simply take $\hat{H} = \hat{H}_1 + \hat{H}_2$ where \hat{H}_1 and \hat{H}_2 are appropriate monotone Hamiltonians for 1D problems.

1.2.1 Lax-Friedrichs Hamiltonian

So what are the possible monotone Hamiltonians at our disposal? As usual, we have a Lax-Friedrichs choice:

$$\hat{H}^{\text{LF}}(u^-, u^+; v^-, v^+) = H\left(\frac{u^- + u^+}{2}, \frac{v^- + v^+}{2}\right) - \frac{1}{2}\alpha^x(u^+ - u^-) - \frac{1}{2}\alpha^y(v^+ - v^-) \quad (11)$$

where $\alpha^x = \max_{u,v} |H_u(u, v)|$ and $\alpha^y = \max_{u,v} |H_v(u, v)|$. Note that the range over which the maximums are taken define whether we have a local or a global Lax-Friedrichs flux (cf. 1D hyperbolic conservation laws). If we have global numbers such that $A \leq u \leq B$ and $C \leq v \leq D$ for all space, then taking maximums over this is a ‘global’ flux. However, if we consider the ranges in the definitions of α^x and α^y as

$$\alpha^x = \max_{\substack{u^- \leq u \leq u^+ \\ v^- \leq v \leq v^+}} |H_u(u, v)|$$

and similarly for α^y . One can show (cf. Osher and Shu) that this flux is not monotone. Can we define a proper local Lax-Friedrichs flux? Sort of. If we define

$$\alpha^x = \max_{\substack{u^- \leq u \leq u^+ \\ C \leq v \leq D}} |H_u(u, v)| \quad \alpha^y = \max_{\substack{A \leq u \leq B \\ v^- \leq v \leq v^+}} |H_v(u, v)|$$

where A, B, C , and D are the appropriate global bounds on u and v , then this in conjunction with equation (11) gives a true monotone Hamiltonian.

1.2.2 Gudonov Hamiltonian

We also have a Gudonov flux for the 2D scheme (10):

$$\hat{H}^G(u^-, u^+; v^-, v^+) = \text{ext}_{u^- \leq u \leq u^+} \text{ext}_{v^- \leq v \leq v^+} H(u, v) \quad (12)$$

where we define the ext function as

$$\text{ext}_{u^- \leq u \leq u^+} = \begin{cases} \min_{u^- \leq u \leq u^+} & \text{if } u^- \leq u^+ \\ \max_{u^+ \leq u \leq u^-} & \text{if } u^- > u^+ \end{cases} \quad (13)$$

Note that the Gudonov flux definition (12) is *not* symmetric, and it is not the same if we reverse the order in which we apply the ext functions. I.e.,

$$\min_{u^- \leq u \leq u^+} \max_{v^+ \leq v \leq v^-} H(u, v) \begin{matrix} \geq \\ \neq \\ \leq \end{matrix} \max_{v^+ \leq v \leq v^-} \min_{u^- \leq u \leq u^+} H(u, v)$$

However, one can use the Gudonov Hamiltonian as shown in equation (12), or one can switch the direction of the application of the ext operators, and it can be shown that either Gudonov flux is a valid monotone Hamiltonian.

1.2.3 Osher-Sethian Hamiltonian

Another type of monotone Hamiltonian which we may implement is the Osher-Sethian Hamiltonian: **If** $H(u, v) = f(u^2, v^2)$ and f is monotone with respect to each argument, one can use the Hamiltonian

$$\hat{H}^{\text{OS}}(u^-, u^+; v^-, v^+) = f(\bar{u}^2, \bar{v}^2) \quad (14)$$

where

$$\bar{u}^2 = \begin{cases} (\min(\bar{u}, 0))^2 + (\max(u^+, 0))^2 & \text{if } f(\downarrow, \cdot) \\ (\min(u^+, 0))^2 + (\max(\bar{u}, 0))^2 & \text{if } f(\uparrow, \cdot) \end{cases} \quad (15)$$

and \bar{v} is defined similarly. The great advantage of this monotone Hamiltonian is its ease of implementation and low computational cost. However, the restriction on the required form of the analytic Hamiltonian limits its applicability. NB this monotone Hamiltonian has the flavor of the Engquist-Osher monotone flux for hyperbolic conservation laws.

A typical equation on which can use the Osher-Sethian Hamiltonian is the *Eikonal Equation*:

$$\phi_t + \sqrt{\phi_x^2 + \phi_y^2} = 0 \quad (16)$$

In this equation $H(u, v) = \sqrt{u^2 + v^2}$, which satisfies the hypotheses of the Osher-Sethian Hamiltonian formulation, so one can use (14) and (15). This equation is solved very often in the context of level-set methods.

1.2.4 Unstructured Grids

The only known monotone flux on unstructured grids is a Lax-Friedrichs flux introduced by Abgrall. See Shu's paper for details and picture. The general form looks like

$$\hat{H}^{\text{LF}} = H\left(\frac{\sum_i \theta_i (\nabla \phi)_i}{2\pi}\right) - \frac{\alpha}{\pi} \sum_i \beta_{i+1/2} \left(\frac{(\nabla \phi)_i + (\nabla \phi)_{i+1}}{2}\right) \cdot \vec{n}_{i+1/2}$$

where the sums are taken over all the adjacent faces.

1.3 High-Order Methods

We have so far described first-order monotone schemes for Hamilton-Jacobi equations. We can now handle equations in one space dimension very well, and we have methods for solving two-dimensional equations either on Cartesian grids, or on unstructured grids. Now we shall attempt to formulate some high-order methods to solve equations (1) and (7).

We again consider the one-dimensional Hamilton-Jacobi equation (1) with the scheme (3) where we require monotonicity of the flux: $\hat{H}(\uparrow, \downarrow)$. Recall from the hyperbolic conservation law case that when formulating high-order methods, we had many options for scheme type (FD vs FV) and there were regularity restrictions on the mesh in 1D. However, for H-J equations in 1D, the idea is much simpler: we restrict ourselves to

- finite-difference schemes
- any mesh

We define a high-order scheme as a semi-discrete ODE of the form

$$\begin{aligned} (\phi_j)_t + \hat{H}(u_j^-, u_j^+) &= 0 \\ u_j^\pm &= \phi_x|_{x=x_j} + O(\Delta x^r) \end{aligned} \tag{17}$$

We impose the following stencils (cf hyperbolic conservation laws):

- The stencil for u^- is biased to the left
- The stencil for u^+ is biased to the right

The modus operandi for this method is very familiar from conservation laws: we identify a stencil over which to interpolate a polynomial ϕ of degree r . When then differentiate this polynomial, and evaluate the result at x_j , and assign that as the value of u_j^\pm .

One main setback for using this method for conservation laws is the following: for conservation laws, the underlying function which the interpolant represents is discontinuous, so our reconstruction will be oscillatory near shocks. However, for H-J equations, the underlying function defining the interpolant is continuous. Thus, one may conjecture that perhaps our direct interpolant reconstruction will be more well-behaved. However, it turns out that even with our increased regularity in ϕ , this straightforward method still does not work except for rather simple problems. (cf Y. Cheng paper)

We can, however, borrow many of the tools we introduced for conservation laws (e.g. ENO). However, we shall need a bigger stencil to achieve the same order since we are differentiating our interpolant and thus require one more degree of freedom.

Let us recall the ENO procedure to compute u^- :

1. Start with the stencil $S^{(1)} = \{\phi_{j-1}, \phi_j\}$. Build the linear function $p^{(1)}(x) = \phi_j - \frac{\phi_j - \phi_{j-1}}{\Delta x}(x - x_j)$
2. Add either (x_{j-2}, ϕ_{j-2}) or (x_{j+1}, ϕ_{j+1}) to the stencil depending on the magnitude of the updating Newton divided difference, which is a measure of the oscillatory nature of the polynomial. E.g. for this step, the divided differences to compare are $\left(\frac{\phi_j - \phi_{j-1}}{\Delta x_{j-1/2}} - \frac{\phi_{j-1} - \phi_{j-2}}{\Delta x_{j-3/2}}\right) \frac{1}{(x_j - x_{j-2})}$ and $\left(\frac{\phi_{j+1} - \phi_j}{\Delta x_{j+1/2}} - \frac{\phi_j - \phi_{j-1}}{\Delta x_{j-1/2}}\right) \frac{1}{(x_{j+1} - x_{j-1})}$. Upon choosing the new stencil, build a new polynomial.

3. Continue in this fashion by enlargening the stencil and updating the polynomial $p^{(r)}$ until the desired accuracy is required. $u^- = p^{(r)}(x_{j-1})$

And this is the abridged WENO procedure:

1. Start with one stencil
2. Consider the all the possible r th-order stencils to add, construct all the possible polynomials, and linearly combine all these polynomials into via nonlinear weights of the form:

$$w_i \sim \int \left[(p'_i(x))^2 + c(p''_i(x))^2 \right] dx.$$

To convert this into an appropriate H-J formulation, we can do the same thing, except for WENO, we use nonlinear weights of the form

$$w_i \sim \int \left[c(p''_i(x))^2 \right] dx.$$

We make this modification because the reconstructed interpolant has more smoothness than in the conversation law case, and it is now the second derivative norm which is a more appropriate measure of regularity. (Cf. Jiang & Peng paper, 5th-order, conforming Cartesian meshes)

If one wishes to do either nonconforming Cartesian meshes or unstructured meshes, the above procedure does not work out-of-the-box. Instead, others have worked about the WENO procedure for triangular meshes (cf Yongtao Zhang thesis).

1.4 DG for Hamilton-Jacobi

Recall FD WENO (Jiang & Peng):

For HCL, transforamtion from global to uniform mesh must be smooth \Rightarrow nonuniform Cartesian meshes do not work. For HJ, said transformation need not be smoot , can do this on nonuniform Cartesian meshes (and smooth meshes). (Thus WENO here is more popular for HJ equation sthan for HCL equations.)

DG for HJ is not quite so popular yet because geometries are regular for many applications, because FD WENO is easily h -adaptable and because FD is more mature.

Applying DG:

1. *Convert HJ to HCL:*

$$\varphi_t + H(\varphi_x) = 0,$$

then set $u = \varphi_x$ and get

$$\int_{I_i} u_t v - \int_{I_i} H(u) v_x dx + (\hat{H}v)|_{i-1/2}^{i+1/2},$$

where $H_{i+1/2} := \hat{H}(u_{i+1/2}^-, u_{i+1/2}^+)$ and $v|_{i+1/2} = v_{i+1/2}^-$ and $v|_{i-1/2} = v_{i-1/2}^+$. Altogether, we get

$$\int_{I_i} (\varphi_x)_t v - \int_{I_i} H(\varphi_x) v_x dx + (\hat{H}v)_{i-1/2}^{i+1/2},$$

where $\hat{H}_{i+1/2} = \hat{H}((\varphi_x)_{i+1/2}^-, (\varphi_x)_{i+1/2}^+)$.

2. *Integrate HJ directly:* In 1D:

$$\int_{I_i} \varphi_t + \int_{I_i} H(\varphi_x) dx = 0.$$

In 2D: $\varphi_t + H(\varphi_x, \varphi_y) = 0$. Setting $u = \varphi_x$, $v = \varphi_y$ gives the nasty system

$$\begin{cases} u_t + H(u, v)_x = 0 \\ v_t + H(u, v)_y = 0 \end{cases} \rightarrow \text{HCL system.}$$

Finte, but the system may be weakly hyperbolic if $(H_u = 0, H_v \neq 0)$ or $(H_v = 0, H_u \neq 0)$.

Despite the complications, we can still write down a DG scheme:

$$(*) \quad \begin{aligned} \int_K u_t w - \int H_t(u, v) w_x + \sum_{e \in \partial K} \int_e \hat{H}_{1,e,K} w d\Gamma &= 0, \\ \int_K v_t z - \int H_t(u, v) z_y + \sum_{e \in \partial K} \int_e \hat{H}_{2,e,K} z d\Gamma &= 0, \end{aligned}$$

where

$$\hat{H}_i := \begin{cases} \frac{\partial H}{\partial u} & i=1, \\ \frac{\partial H}{\partial v} & i=2. \end{cases}$$

If we do what we did in 1D and let $u \rightarrow \varphi_x$, $v \rightarrow \varphi_y$ and we count equations and unknowns, we find an overdetermined system—BUT:

$$\int_K f_t dx dy + \int_K H(u, v) dx dy = 0$$

is OK.

a. Start with $\varphi^n \in P^k$:

$$u^n = \varphi_x^n, \quad v^n = \varphi_y^n.$$

b. Euler forward of (*) to get u^{n+1} , v^{n+1} .

We can advance (*) using a time stepper for $\varphi^n \in P^k$, there may not exist a polynomial of degree k such that $\varphi_x = u^{n+1}$, $\varphi_y = v^{n+1}$. “Do your best”: Find a least-squares φ solution so that

$$\left\| \varphi_x^{n+1} - u^{n+1} \right\|_{L^2}^2 + \left\| \varphi_y^{n+1} - v^{n+1} \right\|_{L^2}^2 = \min_{\psi \in P^k} \left\| \psi_x^{n+1} - u^{n+1} \right\|_{L^2}^2 + \left\| \psi_y^{n+1} - v^{n+1} \right\|_{L^2}^2.$$

(This method is from Hu & Shu in SISC '99.)

The nasty and unmotivated least-squares approach method can however be recast in a more beautiful manner: (Li & Shu) Instead of all of $P^k \times P^k$, we consider only a subspace of it, namely a space $W \subset P^k \times P^k$ whose members satisfy the condition

$$\varphi_{xy} - \varphi_{yx} = u_y - v_x = 0.$$

It turns out that the two procedures yield the same results, as proven in Li & Shu, Appl. Math. Let. 2005.

3. Yet another different method: (Yingda Cheng & Shu) Let's start with something simple:

$$\varphi_t + a(x)\varphi_x = 0$$

or

$$\varphi_t + (a(x)\varphi)_x = a'(x)\varphi,$$

i.e. a hyperbolic conservation law with a source term. Further, assume $a(x) > 0$. Construct a DG method for this

$$\begin{aligned} \int_{I_j} \varphi_t v dx - \int_{I_j} a(x)\varphi v_x dx + a(x_{j+1/2})\varphi_{j+1/2}^- v_{j+1/2}^- - a(x_{j-1/2})\varphi_{j-1/2}^- v_{j-1/2}^+ &= \int_{I_j} a'(x)\varphi v dx \\ &\quad \downarrow \text{(I. by parts)} \\ \int_{I_j} \varphi_t v dx + \int_{I_j} (a(x)\varphi)_x v dx - a(x_{j+1/2})\varphi_{j+1/2}^- v_{j+1/2}^- + a(x_{j-1/2})\varphi_{j-1/2}^+ v_{j-1/2}^+ \\ &\quad + a(x_{j+1/2})\varphi_{j+1/2}^- v_{j+1/2}^- - a(x_{j-1/2})\varphi_{j-1/2}^- v_{j-1/2}^+ = \int_{I_j} a'(x)\varphi v dx \\ &\quad \left[\int_{I_j} (a(x)\varphi)_x v = \int_{I_j} a'(x)\varphi v + \int_{I_j} a(x)\varphi_x v \right] \\ \int_{I_j} (\varphi_t + a(x)\varphi_x) v dx + a(x_{j-1/2})[\varphi]_{j-1/2} v_{j-1/2}^+ &= 0. \end{aligned}$$

We take this as inspiration for the general case $\varphi_t + H(\varphi_x) = 0$, and from the sky falls the following monstrosity (ouch):

$$\begin{aligned} & \int_{I_j} (\varphi_t + H(\varphi_x)) v dx \\ & + \frac{1}{2} \left(\min_{x \in I_{j+1/2}} H'(\varphi_x) - \left| \min_{x \in I_{j+1/2}} H'(\varphi_x) \right| \right) [\varphi]_{j+1/2} v_{j+1/2}^- \\ & + \frac{1}{2} \left(\max_{x \in I_{j-1/2}} H'(\varphi_x) - \left| \max_{x \in I_{j-1/2}} H'(\varphi_x) \right| \right) [\varphi]_{j-1/2} v_{j-1/2}^+ = 0. \end{aligned}$$

Yingda came up with this by experimentally figuring out what perturbation to the coefficient in the above “inspiring” example are possible. This scheme is a pure upwinding scheme, effectively of Roe type. This scheme work satisfactorily except for cells where $H'(\varphi_x) = 0$, where a quick fix is to go back to Hu & Shu.

1.5 Proving Error Estimates

For HCLs, a typical convergence proof goes like this:

- u_h is bounded in some strong (semi-)norm, e.g. BV-norm

$$\|u_h\|_{\text{TV}} = \sum_j |u_{j+1} - u_j|.$$

$\text{TV}(u^{n+1}) \leq \text{TV}(u^n)$ or $\text{TV}(u^n) \leq C(n\Delta t) \Rightarrow$ a subsequence of u_h converges to *something* when $h \rightarrow 0$.

- “Something” is a weak solution if a conservative scheme is used (Lax-Wendroff theorem).
- If a discrete entropy condition (cell entropy inequality/wavewise entropy inequality) is satisfied, then the “something” will be an entropy solution. (*)
- Use the uniqueness of the entropy solution to conclude that the original sequence of the numerical solutions (not just a subsequence) converges to this unique entropy solution. (Every subsequence has a subsequence that converges to something unique.)

Also, there is a “brute-force” approach estimating $\|u - u_h\|_{L^1} \leq C\sqrt{h}$.

For HJ equations, things are generally expected to be better, since the solutions are smoother. In fact, it is not hard to prove $\text{TV}(\varphi) \leq C$ for the Hu & Shu DG scheme. However, there is no equivalent for the step (*) in the HJ case for the first approach. There is however some “brute-force” work done in the literature.

[Lecture 10/3/07: Hyongsu Baek: Level Set Methods]

1.6 Level-Set Methods

Level set methods can be described as methods to capture the evolution of an interface. There are two ways in general to capture an interface:

- Lagrangian: mark locations of the interface and evolve the locations in time
- Eulerian: define a function over space x as $F(x)$ and define the level set as $\{x: F(x) = 0\}$.

In the Lagrangian framework, suppose that we have one marker denoted as \vec{x} which denotes the location of one point on the interface. If we can calculate the temporal derivative of the location of this point and call it \vec{v} , the velocity, then solving

$$\frac{d\vec{x}}{dt} = \vec{v} \tag{18}$$

will generate the evolution of one point on the interface. Now suppose $\vec{x} \in \mathbb{R}^2$ and we want to evolve an interface which is a one-dimensional subset C of \mathbb{R}^2 . We parameterize the points on C via e.g. normalized arclength $p \in [0, 1)$ and we can then solve for

$$\vec{x}(p, t) = (x(p, t), y(p, t))$$

via some evolution method similar to (18). Analogously, if $\vec{x} \in \mathbb{R}^3$, then our unknown is a two-dimensional subspace, and is thus parameterized by two variables:

$$\vec{x}(p, t) = (x(p, q, t), y(p, q, t), z(p, q, t))$$

Now consider the Eulerian framework: we consider a function $\phi: \mathbb{R}^2 \times \mathbb{R}^+ \rightarrow \mathbb{R}$. We let $D \in \mathbb{R}^2$ be a subset of \mathbb{R}^2 . Define

$$\begin{aligned} \Omega^- &= \{(x, y): \phi(x, y, t) < 0\} \\ \Omega^+ &= \{(x, y): \phi(x, y, t) > 0\} \\ \partial\Omega &= \{(x, y): \phi(x, y, t) = 0\} \end{aligned}$$

Then $\partial\Omega \subset \mathbb{R}^2$ is the interface we are looking for, and solving for (x, y) satisfying $\phi(x, y, t) = 0$ defines the interface $\partial\Omega$ as a function of time. Note that this framework is easily generalized to higher dimensions.

Consider a very special form of ϕ as $\phi(x, y, t) = \sqrt{x^2 + y^2} - 1$. This function represents the signed distance function between a point in \mathbb{R}^2 and the unit circle. I.e.,

$$\phi = \begin{cases} d(x, S^1) & x \in \Omega^+ \\ -d(x, S^1) & x \in \Omega^- \end{cases}$$

where S^1 is the unit circle in \mathbb{R}^2 and $d(\cdot, \cdot)$ is the natural Euclidean function on \mathbb{R}^2 . It turns out that any function satisfying $|\nabla d| = 1$ is a distance function.

Consider the Lagrangian framework (again). Suppose that our marker \vec{x} evolves with the law (18). We can decompose the velocity of the marker into a portion tangential to the interface (\vec{T}) and a component normal to the interface (\vec{N}). The portion of the velocity \vec{v} in the direction \vec{T} does not change the shape of the object, it simply rotates the markers around the interface. Thus, it does not contribute to the change in shape and we shall not consider it in our evolution. With this, our new evolution equation simplifies to

$$\frac{d\vec{x}}{dt} = (\vec{v} \cdot \vec{N}) \vec{N}$$

Now consider the Eulerian framework. We wish to solve for $x(t), y(t)$ satisfying $\phi(x(t), y(t), t) = 0$. Taking a total derivative of ϕ yields

$$\begin{aligned} \frac{d\phi}{dt} &= \frac{\partial\phi}{\partial t} + \frac{\partial\phi}{\partial x} \frac{dx}{dt} + \frac{\partial\phi}{\partial y} \frac{dy}{dt} \\ &= \frac{\partial\phi}{\partial t} + \left(\frac{dx}{dt}, \frac{dy}{dt} \right) \cdot \nabla\phi \\ &= \frac{\partial\phi}{\partial t} + F \vec{N} \cdot \nabla\phi, \end{aligned}$$

where

$$\vec{N} = \frac{\nabla\phi}{|\nabla\phi|}.$$

Then the evolution equation simplifies to

$$\frac{\partial\phi}{\partial t} + F |\nabla\phi| = 0. \tag{19}$$

But a distance function satisfies $|\nabla\phi| = 1$, so we recover

$$\frac{\partial\phi}{\partial t} + F = 0$$

for distance functions ϕ . Equation (19) is the starting point for our mathematical discussion.
[Lecture 10/10/07: Chi-Wang Shu]

2 Boltzmann-Type Equations

The function that we are seeking as a solution of Boltzmann-type equations is a density $f(x, u, t)$ over space (1D to 3D), velocity (1D to 3D), and time (1D). Total dimensionality is 6 (+1 for time). The equations that we're dealing with read

$$\begin{aligned} f_t + u f_x - \underbrace{\frac{e}{m} E f_u}_{\text{outside force}} &= \frac{1}{\tau} (n M(u) - f), \\ E &= -\varphi_x, \\ (\varepsilon \varphi_x)_x &= e(n - n_d), \quad \varphi(0) = 0, \quad \varphi(1) = v_{\text{bias}}, \\ n(x, t) &:= \int_{-\infty}^{\infty} f(x, u, t) du, \\ M(u) &= \frac{1}{\sqrt{2\pi\theta}} e^{-2u^2/2\theta}, \end{aligned}$$

where we have given constants $e, m, \tau, v_{\text{bias}}$, and ε, n_d are given functions of x . The coordinates observe the bounds: $0 < x < 1, -\infty < u < \infty, t > 0$. This equation has the following features:

- ⊕ . It's a scalar equation.
- ⊕ . LHS is a simple convection in conservation form. Upwinding is simple to implement.
- ⊕ . RHS is "just" a source term
- ⊖ . Dimension is too high.
 - 1D or at most 2D in space.
 - 1D in space \Rightarrow 1D or 2D in velocity space. (collision dependent)
 - 2D in space \Rightarrow 2D or 3D in velocity space.
- ⊖ . Often the characteristic speed u convection is unbounded. That, of course, kills explicit timestepping in its reliance on CFL-type conditions

$$\max |u| \frac{\Delta t}{\Delta x} + \max \left(\left| -\frac{e}{m} E \right| \right) \frac{\Delta t}{\Delta u} \leq 1.$$

Possible resolutions include:

- Just bound it $-u_{\text{max}} \leq u \leq u_{\text{max}}$.
- u in the Boltzmann equation is replaced by a bounded function $h(u)$.
- ⊖ . More complicated collision models exist:

$$\int K(x, u, u') f(u') du'.$$

x, u : compute a numerical integral in the dimension of u .

Treating the LHS:

- spectral methods.
 - In space:
 - Chebyshev
 - Legendre (no good FFT)
 - In velocity space: $-\infty < u < \infty$ in multiple dimensions
 - Infinite domains: Hermite and Laguerre polynomials have been tried...
 - ...with mixed success

- f can sometimes be pretty nonsmooth
- Lagrange-type methods (follow characteristics)
 - increase Δt tremendously
- Usual “high-resolution” solvers
 - treat f_x and f_u by TVD, ENO, **WENO**, DG
 - drastic restrictions on Δt
 - save time by freezing weights in WENO (high-order linear approximation)

Treating the RHS:

- simple relaxation type collision can be handled explicitly as a source term.
- handled explicitly as a source term
 - difficulty arises when $\tau \ll \Delta x$ or Δu . (“stiff” source term) $\rightarrow \Delta t$ has to be reduced
 - use an implicit solver for the source term
 - could choose an implicit solver to overcome this difficulty
 - or go really high-tech and use *implicit-explicit methods*:

$$f^{n+1} = f^n + \Delta t(C^n + S^{n+1}),$$

where C stands for the convection term and S stands for the source term.

In general, the collision term is a real integral.

- Fatemi & Odeh: use linear interpolation to actually evaluate the integral.
- Possibly use WENO interpolation to avoid oscillation.
- A. Majorana (2001) et al. use a clever transform to avoid interpolation and have the integral evaluation always use values at exact grid points.
- Possibly use Fast Multipole Methods to reduce computational cost.
- Or use FFT.

[Begin lecture October 17, 2007: Chi-Wang Shu]

3 Semiconductor Device Simulation Models

[Name-dropping in Shu’s circle of friends: Joseph Jerome (Northwestern University), Odeh (Bell labs?), and Fatemi (NU as well? Graduated)]

3.1 Drift-diffusion model

$$n_t + J_x = 0, \tag{20}$$

where $n(x, t)$ is the number density of electrons, and J is the current density. Physical properties of the medium include the doping density n_d . We can decompose J into a hyperbolic and a viscous factor as follows:

$$J = J_{\text{hyp}} + J_{\text{vis}}. \tag{21}$$

We define

$$J_{\text{hyp}} = -\mu n E,$$

where μ is the *electron mobility*. A typical value for μ is $0.75 [\mu m^2/V]$. One of the usual functions values for μ is

$$\mu = \begin{cases} 0.75 & \text{where } n_d = 10^6 (\mu m)^{-3} \\ 4.0 & \text{where } n_d = 2 \times 10^3 (\mu m)^{-3} \end{cases}$$

or we can also make μ a function of E (the electric field) as

$$\mu = \mu(E) = \frac{2\mu_0}{1 + \sqrt{(1 + 4 \left(\frac{\mu_0 |E|}{v_d}\right)^2)},}$$

for $\mu_0 = 4.0$, $v_d = 2.0 [\mu m/ps]$. Recall that E is the electric field given by

$$E = -\phi_x,$$

where ϕ is the *electric potential* and is solved for via

$$\left. \begin{aligned} (\varepsilon \phi_x)_x &= e(n - n_d) \\ \phi(0) &= 0, \quad \phi(L) = v_{\text{bias}} \end{aligned} \right\}$$

ε is the *electric permittivity* and v_{bias} is some bias potential. Now we can define the diffusive term as

$$J_{\text{vis}} = -\sigma(n\theta)_x,$$

where we have introduced the temperature θ

$$\theta = \frac{k_b}{m} T_b.$$

Note that in this model, the diffusive term $(J_{\text{vis}})_x$ is linear in n , so this is a relativey ‘easy’ model. However, this model is difficult to compute because the doping n_d is very often discontinuous, and leads to very sharp gradients (layers) in the number density n .

A very popular method (e.g. commercial softwares) for solving drift-diffusion systems like this one is the *Schurfetter-Gummel* method: suppose we have steady state. Then the equation basically reads (20)

$$(an)_x - (bn_x)_x = 0$$

Integrating as

$$\frac{a}{b}n - n_x = \text{const.}$$

then the solution can be written as

$$n = c e^{\frac{ax}{b}}.$$

However, one can still just hit (20) with ENO/WENO/DG solvers, and it will work quite well.

3.2 High field model

(See Cercignani and Gamba and Levermore)

Idea: use Maxwell’s equations as the driving force. Start with same equations (20) and (21). But now take

$$\begin{aligned} J_{\text{hyp}} &= -\mu n E - \tau \mu \frac{e}{\varepsilon} n (-\mu n E + \omega) \\ J_{\text{vis}} &= -\tau (n(\theta + 2\mu^2 E^2))_x + \tau \mu E (\mu n E)_x \end{aligned}$$

Computationally, is not that much different than drift-diffusion: just add some more terms to solver.

3.3 Momentum solvers

So far we have only conserved ‘mass’, i.e. the mass density $n(x, t)$. However, it also makes sense to conserve ‘momentum’:

$$\text{nu} = \int f v dv,$$

where f is the phase-space density function (unknown in the Boltzmann eqn).

3.4 Energy Transport Models

Idea: use mass and ‘energy’ conservation:

$$W = \int f (v - nu)^2 dv$$

Then we model momentum explicitly so that we don’t have to conserve it. The equations are then conservation of mass (20) and some conservation law for W . There are many, many types of energy transport models. A particular one is

$$u_t + f(u)_x = g(u)_{xx} + h(u),$$

with

$$u = \begin{pmatrix} \frac{e n}{n E} \\ \frac{n E}{m} \end{pmatrix}, \quad f(u) = \phi' n \begin{pmatrix} e \mu(E) \\ \mu^E(E) + D(E) \end{pmatrix}, \quad g(n) = \begin{pmatrix} n D(E) \\ n D^E(E) \end{pmatrix},$$

and

$$h(u) = \begin{pmatrix} 0 \\ e n \mu(E) (\phi')^2 + \frac{e}{\epsilon} (n - n_a) n D(E) - n \left\langle \frac{\partial E}{\partial t} \Big|_{\text{coll}} \right\rangle \end{pmatrix}$$

Advantage of this model: only two unknowns (as opposed to three for momentum conservation).

3.5 Hydrodynamic Model

$$\begin{cases} n_t + (n v)_x = 0 \\ p_t + (p v + n k_b T)_x = -e n E - \frac{p}{\tau_p} \\ W_t + (n w + n v k_b T)_x = -\sigma n v E - \frac{W - W_0}{\tau_W} + (\kappa n T_x)_x \end{cases} \quad (22)$$

p is the momentum, T is temperature, E is energy. The LHS of the system above exactly the Euler equations of gas dynamics. This is great: we can just apply the usual hyperbolic conservation law techniques. However, it’s costly: in 2D, we’ve got 4 unknowns.

3.6 Kinetic Model

$$f_t + u f_x - \frac{e}{m} E f_u = \frac{n M - f}{\tau}$$

f is the phase-space density function. M is the Maxwellian (Gaussian) profile in velocity-space. This model is more general than drift-diffusion, high field, and (maybe) energy transport. However, one cannot derive the hydrodynamic model from this one. One could also replace the right-hand side by

$$Q(f) = \int_{\mathbb{R}^3} [s(k', k) f(t, x, k') - s(k, k') f(t, x, k)] dk',$$

where the kernel s is defined as

$$s(k, k') = k_0(k, k') \delta(z(k') - z(k)) +$$

$$k(k, k') [(n_q + 1) \delta(z(k') - z(k) + \hbar\omega) + n_1 (\delta(z(k') - z(k) - \hbar\omega))],$$

where

$$z(k) = \frac{1}{1 + \sqrt{1 + z \frac{\tilde{\sigma}}{m} \hbar^2 |k|^2}} \frac{\hbar^2}{m} |k|^2.$$

3.7 Quantum Effects

Earliest attempts at incorporating quantum effects into these models is attributed to C. Gardner, who developed a quantum hydrodynamic model. Unfortunately, that first model looks like it might have some problems with it. Quantum models for simpler models (drift-diffusion) have been done, and are relatively well-accepted.

Nobody has really done anything solid with quantum effects in the kinetic model.

[Begin Lecture 10/25/07: Alan Schiemenz]

4 Fast Sweeping

We're solving the 2D Eikonal equation

$$|\nabla\varphi(x, y)| = f(x, y) \quad \text{for } x \in \Omega \setminus \Gamma$$

with the condition $\varphi(x) = 0$ for x on some given curve Γ .

4.1 Godunov Hamiltonian

Let $\Omega_h = \{(x_i, y_j)\}_{i,j=1}^{I,J}$, where $h = \Delta x = \Delta y$. and then set

$$\hat{H} : \left[\left(\frac{\varphi_{i,j} - \varphi_{x,\min}^{i,j}}{h} \right)^+ \right]^2 + \left[\left(\frac{\varphi_{i,j} - \varphi_{y,\min}^{i,j}}{h} \right)^+ \right]^2 = f_{i,j}^2, \quad (23)$$

where

$$\begin{aligned} \varphi_{x,\min}^{i,j} &:= \min(\varphi_{i-1,j}, \varphi_{i+1,j}), \\ \varphi_{y,\min}^{i,j} &:= \min(\varphi_{i,j-1}, \varphi_{i,j+1}). \end{aligned}$$

Steps in the algorithm:

1. Initialize $\varphi = \infty$
2. Fix φ on Γ , never let it change.
3. Until convergence, do
 - a. Pick the next **sweep_order** out of
 - $i = 1 \dots I, j = 1 \dots J$
 - $i = I \dots 1, j = 1 \dots J$
 - $i = I \dots 1, j = J \dots 1$
 - $i = 1 \dots I, j = J \dots 1$

and for each point in the sweep according to **sweep_order**, do:

- i. solve (23) for a quantity $\bar{\varphi}_{i,j}$
- ii. set the new $\varphi_{i,j} := \min(\bar{\varphi}_{i,j}, \varphi_{i,j}^{\text{old}})$

in a sweeping motion, moving according to **sweep_order**.

The solution to (23) can be written down explicitly:

$$\bar{\varphi}_{i,j} = \begin{cases} \min(a, b) + f_{i,j}h & |a - b| \geq f_{i,j}h, \\ \frac{a + b + \sqrt{2f_{i,j}h^2 - (a - b)^2}}{2} & \text{else.} \end{cases}$$

Remarks:

- Easily generalizable to multiple dimensions
- The entire thing is essentially a Gauß-Seidel update. Since Gauß-Seidel is order-dependent, we try each ordering in turn.

- A 1D problem is done after 2 sweeps.
- In $O(1)$ sweeps, you can bring down the error to order h .

Thinking about the 1D situation is really helpful: Solve $|\partial_x f| = 1$, keeping $f(0) = f(1)$. Sweeping right just gives $f(x) = x$, and sweeping left puts the mandatory kink at $x = 0.5$ if starting out to the left with slope 1 from $f(1) = 0$.

4.2 Lax-Friedrichs Hamiltonian

Next, we're trying to solve a more horrible, even more nonlinear equation:

$$\psi_t + \gamma \left(\frac{\nabla \psi}{|\nabla \psi|} \right) |\nabla \psi| = 0,$$

where again we're given $\psi = 0$ on the curve Γ , and γ is sort of a "normal speed" or a "surface tension". Focusing on the zero level set gives us the time-independent equation

$$\underbrace{\gamma \left(\frac{\nabla \psi}{|\nabla \psi|} \right) |\nabla \psi|}_H = 1$$

on all of \mathbb{R}^d . The solution of this problem takes a shape called *Wulff crystals*.

To tackle this nonlinearity, we use the Lax-Friedrichs Hamiltonian, which in 1D reads

$$\hat{H}_{i,j}(\rho^-, \rho^+) = H \left(\frac{\rho^+ + \rho^-}{2} \right) - \sigma_x \frac{\rho^+ - \rho^-}{2},$$

where

$$\sigma_x \geq \max_{\rho \in [\rho^-, \rho^+]} \left| \frac{\partial H}{\partial p} \right|.$$

To repeat, our equation reads $H(\nabla \varphi) = R(x)$, with $\varphi = g$ on Γ given. Then our L-F update can be written as

$$\bar{\varphi}_i = \frac{\Delta x}{\sigma_x} \left(R(x_i) - H \left(\frac{\varphi_{i+1} - \varphi_{i-1}}{2\Delta x} \right) \right) + \frac{\varphi_{i+1} - \varphi_{i-1}}{2}, \quad (24)$$

and again we'll have $\varphi_i^{\text{new}} = \min(\varphi_i^{\text{old}}, \bar{\varphi}_i)$.

The computational boundary is handled by means of ghost cells, setting

$$\begin{aligned} \varphi_{0,j} &= \varphi_{i,0}, \\ \varphi_{I+1,j} &= \varphi_{i,J+1}. \end{aligned}$$

Remarks:

- We're no longer unwinding—(24) takes information from *both* sides of the point currently being updated.
- The flux is monotone. \oplus
- It's easier to do in higher dimensions. \oplus
- No nonlinear inversion necessary. \oplus
- Takes many more Gauß-Seidel iterations to converge. \ominus

4.3 High-Order Methods

In order to achieve high-order, we will look at a WENO approximation $\nabla \varphi$. Recall:

$$\begin{aligned} (\varphi_x)_{i,j}^{\text{Godunov}} &= \frac{\varphi_{i,j} - \varphi_{i,j}^{x,\min}}{\Delta x}, \\ (\varphi_x)_{i,j}^{\text{LF}} &= \frac{\varphi_{i+1,j} - \varphi_{i-1,j}}{2\Delta x}. \end{aligned}$$

Now we're going to have

$$\begin{aligned}(\varphi_x)_{i,j}^- &= (1 - w_-) \frac{\varphi_{i+1,j} - \varphi_{i-1,j}}{2h} + w_- \frac{3\varphi_{i,j} - 4\varphi_{i-1,j} + \varphi_{i-2,j}}{2h}, \\(\varphi_x)_{i,j}^+ &= (1 + w_+) \frac{\varphi_{i+1,j} - \varphi_{i-1,j}}{2h} + w_+ \frac{-\varphi_{i+2,j} + 4\varphi_{i+1,j} - 3\varphi_{i,j}}{2h},\end{aligned}$$

where our weights are

$$\begin{aligned}w_- &= \frac{1}{1 + 2\Gamma_-^2}, & \Gamma_- &= \frac{\varepsilon + [\varphi_{i,j} - 2\varphi_{i-1,j} + \varphi_{i-2,j}]^2}{\varepsilon + [\varphi_{i+1,j} - 2\varphi_{i,j} + \varphi_{i-1,j}]^2}, \\w_+ &= \frac{1}{1 + 2\Gamma_+^2}, & \Gamma_+ &= \frac{\varepsilon + [\varphi_{i+2,j} - 2\varphi_{i+1,j} + \varphi_{i,j}]^2}{\varepsilon + [\varphi_{i+1,j} - 2\varphi_{i,j} + \varphi_{i-1,j}]^2}.\end{aligned}$$

$(\varphi_y)^\pm$ are analogous. We introduce the Godunov Hamiltonian just as above:

$$\bar{\varphi}_{i,j} = \begin{cases} \min(a, b) + f_{i,j}h & |a - b| \geq f_{i,j}h, \\ \frac{a + b + \sqrt{2f_{i,j}h^2 - (a - b)^2}}{2} & \text{else.} \end{cases}$$

But this time, we are going to set

$$a = u_{x,\min} = \min\left[(\varphi_{i,j}^{\text{old}} - \Delta x(\varphi_x)_{i,j}^-), (\varphi_{i,j}^{\text{old}} - \Delta x(\varphi_x)_{i,j}^+)\right],$$

and b likewise.

using our WENO approximations.

Remarks:

- Neither monotone nor upwinded.
- We do not take $\varphi_{i,j}^{\text{new}} = \min(\varphi_{i,j}^{\text{old}}, \bar{\varphi}_{i,j})$, because we might accidentally generate values *below* the viscosity solution, which would kill our nice high-order accuracy. Instead, we take $\varphi_{i,j}^{\text{new}} = \bar{\varphi}_{i,j}$.

WENO may also be introduced for the Lax-Friedrichs Hamiltonian. Recall our previous result

$$\bar{\varphi}_{i,j} = \frac{1}{\frac{\sigma_x}{\Delta x} + \frac{\sigma_y}{\Delta y}} \left[R_{i,j} - H\left(\frac{\varphi_{i+1,j} - \varphi_{i-1,j}}{2\Delta x}, \frac{\varphi_{i,j+1} - \varphi_{i,j-1}}{2\Delta y}\right) + \sigma_x \frac{\varphi_{i+1,j} - \varphi_{i-1,j}}{2\Delta x} + \sigma_y \frac{\varphi_{i,j+1} - \varphi_{i,j-1}}{2\Delta y} \right].$$

To introduce WENO into this, we do the following:

1. Replace

$$H(\cdot, \cdot) \rightarrow H\left(\frac{(\varphi_x)_{i,j}^- + (\varphi_x)_{i,j}^+}{2}, \frac{(\varphi_y)_{i,j}^- + (\varphi_y)_{i,j}^+}{2}\right)$$

2. Replace

$$\frac{\varphi_{i+1,j} + \varphi_{i-1,j}}{2} \rightarrow \frac{2\varphi_{i,j}^{\text{old}} + \Delta x[(\varphi_x)_{i,j}^+ - (\varphi_x)_{i,j}^-]}{2\Delta x}.$$

Remarks:

- LF-WENO is more robust than Godunov-WENO.
- The high-order methods require a good starting guess to actually converge. One possibility is to have a low-order method to produce a starting guess, and then use high-order.

[Begin Lecture 10/31/07: Ishani Roy]

5 Boltzmann Equations for Semiconductor Devices

The Boltzmann equation models the evolution of (multiple) species of particles in phase (physical-velocity) space. The collisionless Boltzmann equation is called the *Vlasov Equation*. What kinds of collisions can we have? In semiconductor devices, electron-electron and electron-hole collisions are negligible (hole = positively charged ion). The only external applied field we consider is the electric field.

Consider the number density $f = f(t, x, v)$. The rate of change due to convection is zero (neglecting collision):

$$\left(\frac{df}{dt}\right)_{\text{conv}} = 0$$

It's reasonable to say the rate of change of collision and convection are the same:

$$\left(\frac{df}{dt}\right)_{\text{conv}} = \left(\frac{df}{dt}\right)_{\text{coll}}$$

5.1 Convection

The convection can be carried out by following characteristics:

$$\dot{x} = u$$

$$\dot{u} = -\frac{q}{m}E(x, t)$$

5.2 Collision

The collision term $P(t, x, u' \rightarrow u)$ is the rate that a particle with position x at time t changes its velocity from u' to u due to a scattering (collision). Via probabilities and Pauli exclusion principle, the scattering term has the form

slide advanced (d'oh!)

5.3 Semiclassical Boltzmann Eqn

This formulation employed to incorporate quantum effects in semiconductor crystal lattice. Semiclassical Boltzmann eqn:

$$\partial_t f + u(k) \cdot \nabla_x f - \frac{e}{\hbar} E \cdot \nabla_k f = Q(f),$$

where

$$Q(f) = \int_{(\mathbb{R}^3)} [s(x, k', k) f(t, x, k') (1 - f(t, x, k)) - s(x, k, k') f(t, x, k) (1 - f(t, x, k'))] dk'$$

Types of collisions:

- conservation of particles
- Relaxation to thermal equilibrium
- (something else)
- Low-density approximation: quadratic term ignored

$$\int_{\mathbb{R}^3} Q(f) dk = 0$$

We will look at relaxation to equilibrium: the Maxwellian (equilibrium distribution) is

$$M(k) = N^* e^{\left(-\frac{\varepsilon(k)}{k_B T_L}\right)}$$

where N^* is a normalizing term.

We also will look at Polar optical scattering:

$$Q(f) = \int_{\mathbb{R}^3} [S(k', k) f(t, x, k') - S(k, k') f(t, x, k)] dk',$$

where

$$S(k, k') = \sum_{i=1}^n G_i(k, k') [(n_i + 1) \delta(\varepsilon(k') - \varepsilon(k) + \hbar\omega_i) + \text{something else}]$$

The $\delta((\varepsilon(k') - \varepsilon(k) + \hbar\omega_i))$ function is used as an approximation. Usually $\varepsilon(k)$ can be taken as some simple smooth function to make the calculation of $Q(f)$ easier. We can take

- $\varepsilon(k) =$ parabolic
- non-parabolic
- Kane dispersion

A common application of this equation is for Gallium-arsenide semiconductors. Direct band-gap semiconductors like GaAs can be used for LEDs and for semiconductor lasers.

5.4 Kinetic model for a GaAs diode

$$\frac{\partial f}{\partial t} + u \frac{\partial f}{\partial x} - \frac{e}{m} E(x, t) = \text{blah blah}$$

In order to determine the electric field, we need to solve a Poisson problem

$$\left. \begin{aligned} E(x, t) &= -\phi_x \\ (\varepsilon\phi_x)_x &= e(n(x, t) - n_d) \\ \phi(0, t) &= 0 \\ \phi(1, t) &= v_{\text{bias}} \end{aligned} \right\}$$

Usually the collision term is

$$\frac{n(x, t)M(u) - f(t, x, u)}{\tau}$$

All the moments (except the zeroth order one) relax to the equilibrium at the same rate, but momentum and energy realistically don't do this, so this is a disadvantage of this method.

This equation has stuff like

$$f_t + u f_x = 0,$$

but u is technically unbounded. This is bad for CFL conditions. The way to get around this is to discretize so that you cut off u at say $|u| \leq a$, and you increase a until you see that the answer doesn't change. E.g. WENO is used to solve these things.

5.5 WENO solver for transients of the Boltzmann-Poisson system

$$\frac{\partial f}{\partial t} + \frac{1}{\hbar} \frac{\partial \varepsilon}{\partial k} \cdot \frac{\partial f}{\partial x} - \frac{e}{\hbar} E \cdot \frac{\partial f}{\partial k} = Q(f)$$

is the model we shall use. \hbar and e are constants. In this case, we take $\varepsilon(k)$ in $Q(f)$ as

$$\varepsilon(k) = \frac{1}{1 + \sqrt{1 + 2 \frac{\tilde{\alpha}}{m} \hbar^2 |k|^2}} \frac{\hbar^2}{m} |k|^2, \quad (25)$$

where $\tilde{\alpha}$ is the non-parabolicity factor (parabolic band $\Rightarrow \tilde{\alpha} = 0$). we use $\tilde{\alpha} = 0.5$. The collision operator is

$$Q(f) = \int [S(k', k) f(t, x, k') - S(k, k') f(t, x, k)] dk',$$

and S is some horrible function of $\varepsilon(k)$. Again the E -field is obtained via a Poisson system.

5.5.1 Solving

We've got a 6-D in space, 1-D in time for a 3D device. This huge physical space with the collision integral is motivation for using DSMC. DSMC is simple, and relatively cheap to compute. The disadvantage is that this method introduces some noise, so we need lots and lots of particles (Monte-Carlo trials) in order to get a reasonable approximation back. This is why we need deterministic solvers. However, this is expensive. Ways that have been explored to do this are

- Use spherical coordinates and energy as an unknown \Rightarrow can get down to 2 dimensions
- Use parabolic approximation, solve HCL with upwinding and a predictor-corrector in time. However, there is singularity in the transformation, so you've got to do some fancy footwork to get around this.
- Mazorana and Pidatella use a new transformation of variables which is good even in the non-parabolic case, and changes the BP system into an HCL, and this is great.

The HCL is linear if we neglect nonlinearities in E . A good way to avoid oscillations inherent with high-order oscillations is WENO. Usually these applications are 1D, but they 'can be generalized' to as much as 2D (5D phase space) problem.

5.5.2 Transformation

Crazy transformation in k :

$$k = \sqrt{2} \frac{\sqrt{m k_B T_L w}}{h} \sqrt{w} \sqrt{1 + \alpha_k w} \left(\sqrt{1 - \mu^2} \cos \phi, \sqrt{1 - \mu^2} \sin \phi, \mu \right).$$

In addition we use (25). Need to find the Jacobian J also, which can be done. The equation is nondimensionalized by introducing appropriate variables for k, t, x . the new unknown is

$$\begin{aligned} \Phi(t, z, w, \mu) &= s(w) F(t, z, w, \mu) \\ s(w) &= \sqrt{w} \sqrt{1 + \alpha_k w} (1 + 2 \alpha_k w) \end{aligned}$$

where $F(t, z, w, \mu)$ is the direct transformation of $f(t, x, v)$ at some energy. ($z \sim x, w \sim |v|, \mu \sim \cos(\text{angle between } z \text{ and } k)$).

$$f \cong \frac{1}{2} \iiint \Phi \, dw \, d\mu \, dz$$

We now write the collision term as $C(\Phi)$, which is a one-dimensional integral over μ . Now $z \in [0, L], w \in [0, w_{\max}], \mu \in [-1, 1]$. We let $w_{\max} = N h w$ be the maximum value of the energy. We can also introduce some 'cut' in the collision kernel so that the number of electrons with velocity less than w_{\max} is constant in time. The new eqn is

$$\frac{\partial \Phi}{\partial t} + \frac{\partial(a_1 \Phi)}{\partial z} + \frac{\partial(a_2 \Phi)}{\partial w} + \frac{\partial(a_3 \Phi)}{\partial \mu} = s(\mu) C(\Phi)$$

This is a form for which we can do successive one-dimensional differentiations, in contrast to the non-transformed equation, for which we must do a three-dimensional differentiation. In addition, $C(\Phi)$ is a one-dimensional integral (as opposed to the original three-dimensional integral without the δ -function approximation).

5.5.3 Numerical Scheme

The numerical scheme uses a 5th order FD WENO method with a 3rd order TVD RK in time. w_{\max} is chosen to be an integer multiple of α , where α is some translation in w in evaluating Φ needed to compute the collision operator. w_{\max} is thus chosen so that $\Phi(t, z, w \pm \alpha, \mu)$ lands exactly on a grid point. Thus, Δw is also chosen so that $w_j + \alpha$ is an integral multiple of Δw . This saves time so that we don't have to do any high-order interpolation to evaluate $\Phi(t, z, w \pm \alpha, \mu)$. The scheme is to integrate the BTE on some $w \times \mu$ rectangle. We use a tensor-product like approximation so that we can do constant-coefficient one-dimensional derivative approximations for a multi-dimensional non-constant coefficient problem.

[Begin Lecture 11/07/07: Yanlai Chen]

6 Adaptive High-Order DG Method w/Error Control for H-J Equations

6.1 Error Indicator Methods

(cf. R. Nochetto)

The general adaptive method includes the following (looped) steps:

1. Solve a problem on a triangulation \mathcal{T}_h
2. Estimate the error in solving on this mesh
3. Mark elements for which the error is too large
4. Refine/Coarsen the grid according to the marked elements

The ‘estimate’ step usually involves estimates of the error e as follows:

- $\|e\| \leq C \sum_{T \in \mathcal{T}_h} \eta(T)$ (upper bound)
- Lower bound: $C' \sum_{T \in \mathcal{T}_h} \eta(T) \leq \|e\|$

Often times we don’t really use the lower bound because we don’t do much coarsening. The ‘refine’ step usually involves refined meshes $\hat{\mathcal{T}}_h$ and to determine:

$$\operatorname{argmin}_{\hat{\mathcal{T}}_h} \left\{ \sum_{T \in \hat{\mathcal{T}}_h} \eta(T) \geq \theta \eta(\Omega) \right\}$$

for some $\theta \in (0, 1)$. If you do this, with the estimation loop, then one can prove convergence on loop k :

$$\|u - u_h\| \leq C \beta^k$$

for some $\beta < 1$.

6.2 Moving Mesh Methods

(cf. T. Tang)

The steps here are

1. Evolve n steps
2. Redistribute the mesh
3. Update

One keeps the number of elements $|\mathcal{T}_h|$ the same, but redistributes the vertices (and thus nodes) of the elements according to a ‘monitor function’, of which

$$w = \sqrt{1 + |\nabla u|^2}$$

is an example. Let $x(\xi)$ be the physical domain (global) quantities while ξ is the computational domain (standard) coordinates. One evolves the element vertices as

$$(w x_\xi)_\xi = 0 \Rightarrow w x_\xi = 0$$

6.3 Adaptivity with Error Control

The procedure here is

1. Solve
2. Estimate

3. Generate a new mesh

Utilize a computable function Φ which satisfies $\|u - u_h\| < \Phi(u_h)$.

The equation we're trying to solve is

$$u + H(\nabla u) = f. \quad (26)$$

Why no time-evolution? (Unofficially: can't prove error bound.) However, suppose ϕ solves the Eikonal equation, e.g. (16). Then $u = e^\phi$ satisfies (26).

Another way: first obtain $\phi = u_x$, which solves

$$\phi + H(\phi)_x = f_x$$

and compute u via

$$u(x) = f(0) - H(\phi(0)) + \int_0^x \phi(s) ds$$

Note: u is also a steady-state solution to

$$w_t + w + H(w_x) = 0.$$

So another way for obtaining ϕ is to solve

$$\psi_t + \psi + H(\psi)_x - f_x = 0.$$

Newton's method is used to solve the weak formulation of (26). However, Newton's method only converges for sufficiently close initial guesses. So a dual Runge-Kutta-type method is used to supplement Newton-Raphson far away from the point of convergence.

6.3.1 A Posteriori Error Estimate

Several estimates have been attempted:

- 1995: Cockburn & Gau developed a method for conservation laws \Rightarrow H-J equations. However, the effectivity indicator is $O(h^{-1})$.
- Crandall & Lions: Continuous dependence \Rightarrow error estimate, the effectivity index is $O(1)$ in smooth, monotone regions, but $O(h^{-1})$ for non-smooth solutions
- Albert, Cockburn, French, & Paterson developed an $O(1)$, or $O(\log h)$ effectivity index for monotone solutions (in smooth or non-smooth regions). However, it's $O(h^{-1})$ for high-order DG schemes. However, one can postprocess the solution to get an $O(1)$ effectivity index, but only when the mesh is uniform.
- Chen and Cockburn: the effectivity index is $O(1)$, $O(\log h)$ for *any* scheme, but only for one-dimensional problems.

6.3.2 Continuous Dependence

One can show

$$\begin{cases} u + H(\nabla u) = f \\ v + H(\nabla v) = g \end{cases} \Rightarrow \|u - v\|_{L^\infty} \leq \|f - g\|_{L^\infty}$$

Then we can form

$$\begin{cases} u + H(\nabla u) = f \\ u_h + H(\nabla u_h) = g \end{cases}$$

where we define g as

$$g = \begin{cases} \sup \{u_h + H(p) : p \in D^+\} & \text{if } D^+ \neq \emptyset \\ \inf \{u_h + H(p) : p \in D^-\} & \text{if } D^- \neq \emptyset \end{cases}$$

in accordance with the theory of viscosity solutions. If we do this then

$$\|u - u_h\|_{L^\infty} \leq \sup \{|R(u_h)|\},$$

where R is the residual.

[Missed Lecture 11/29/07: Wei Wang and Dan Paulsen]

[Begin Lecture 12/05/07: Chi-Wang Shu]

7 Other Applications

7.1 Hierarchical Size-Structured Model

Much of this section is taken from A.S. Acnleh, et al, *Applied Math Optimization*, v51 (2005), pp 35-59.
Notation:

- x : size
- t : time
- $u(x, t)$: density of individuals of size x at time t

We take $(x, t) \in (0, L] \times (0, T]$, where L and T are some upper bounds. The PDE is

$$u_t + (g(x, Q(x, t)) u)_x + m(x, Q(x, t)) u = 0 \quad (27)$$

We can call this a conservation law with a ‘source term’. g represents some sort of growth. Boundary conditions can be prescribed as

$$g(0, Q(0, t)) u(0, t) = C(t) + \int_0^L \beta(x, Q(x, t)) u(x, t) dx, \quad t \in (0, T].$$

C can be viewed as some source (e.g. growth from some external source). $Q(x, t)$ is called the ‘environment’. It is defined as

$$Q(x, t) = \alpha \int_0^x w(\xi) u(\xi, t) d\xi + \int_x^L w(\xi) u(\xi, t) d\xi.$$

Thus the ‘environment’ depends globally on the density everywhere. $0 \leq \alpha < 1$ is a constant which takes into account the notion that individuals with size smaller than x have limited (or reduced) influence on those of size x . Some physical interpretations of certain quantities are

- g : growth rate
- m : mortality rate
- β : reproduction rate
- $C(\theta)$: inflow rate of size-zero individuals from some external source

We assume some hypotheses on properties and regularity of the system:

1. $g(x, Q) \in C^2$, $g(x, Q) > 0$ for $x \in (0, L)$, $g(L, Q) = 0$, $g_Q(x, Q) \leq 0$
2. $m(x, Q) \geq 0$, $m \in C^1$
3. $\beta(x, Q) \geq 0$, $\beta \in C^1$, $\beta(x, Q) \leq \omega_1$
4. $w(x) \geq 0$, $w \in C^1$
5. $C(t) \geq 0$, $C \in C^1$
6. $u(x, 0) \in \text{BV}(0, L)$, $u(x, 0) \geq 0$

Acnleh uses a first-order backward-Euler scheme with upwinding:

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + \frac{g(x_j, Q(x_j, t^{n+1})) u_j^{n+1} - g(x_{j-1}, Q(x_{j-1}, t^{n+1})) u_{j-1}^{n+1}}{\Delta x} + m(x_j, Q(x_j, t^{n+1})) u_j^{n+1} = 0.$$

The quadrature used to compute the integrals in the definition of Q is Riemann integration using right-hand values. This is to avoid using the value $u(0, t)$, which depends in a complicated fashion on the global values of u .

The above scheme is weighed toward theory more than practice: it's hard to implement but one can prove existence and uniqueness for solutions to the PDE and convergence of the scheme. If one instead uses Euler forward, then the implementation becomes much easier (using the right-hand Riemann quadrature). However, proving convergence is much harder, but it can be done (Jun Shen). However, both schemes are first-order.

Suppose we're looking for a second-order TVD scheme. One ingredient would be second-order quadrature, which requires $u(0, t)$, and this makes implementation much harder. But this can also be done.

[long, long discussion on quadrature]

One can also do e.g. 5th order WENO to do this stuff.

[comments about DG conference]

7.2 Error estimate for DG

Consider the equation

$$\phi_t + a \phi_x = b \phi,$$

with $a > 0$. We define the discontinuous finite element space

$$V_h = \{v: v|_{I_j} \in P^k(I_j)\}.$$

The scheme is: find $\phi_h \in V_h$ such that

$$B_j(\phi_h, v_h) = \int_{I_j} (\phi_h)_t v_h dx - a \int_{I_j} \phi_h (v_h)_x dx + a (\phi_h)_{j+\frac{1}{2}}^- (v_h)_{j+\frac{1}{2}}^- - a (\phi_h)_{j-\frac{1}{2}}^- (v_h)_{j-\frac{1}{2}}^+ - \int_{I_j} b \phi_h v_h dx = 0$$

for all $v_h \in V_h$.

Proposition 6. (*Cell entropy inequality for DG*)

$$\frac{1}{2} \frac{d}{dt} \int_{I_j} (\phi_h)^2 dx + \hat{F}_{j+\frac{1}{2}} - \hat{F}_{j-\frac{1}{2}} \leq b \int_{I_j} \phi_h^2 dx$$

which implies

$$\frac{1}{2} \frac{d}{dt} \int_0^1 (\phi_h)^2 dx \leq b \int_0^1 (\phi_h)^2 dx$$

Proof. Take $v_h = \phi_h$:

$$\begin{aligned} 0 &= B_j(\phi_h, \phi_h) = \int_{I_j} (\phi_h)_t \phi_h dx - a \int_{I_j} \phi_h (\phi_h)_x dx + a (\phi_h)_{j+\frac{1}{2}}^- (\phi_h)_{j+\frac{1}{2}}^- - a (\phi_h)_{j-\frac{1}{2}}^- (\phi_h)_{j-\frac{1}{2}}^+ - \int_{I_j} b \phi_h \phi_h dx \\ &= \frac{d}{dt} \frac{1}{2} \int_{I_j} (\phi_h)^2 dx - \frac{1}{2} a (\phi_h^2)_{j+\frac{1}{2}}^- + \frac{1}{2} a (\phi_h^2)_{j-\frac{1}{2}}^+ + \dots \\ &= \frac{1}{2} \frac{d}{dt} \int_{I_j} (\phi_h)^2 dx + \hat{F}_{j+\frac{1}{2}} - \hat{F}_{j-\frac{1}{2}} + \Theta_{j-\frac{1}{2}} - b \int_{I_j} \phi_h^2 dx. \end{aligned}$$

where we have defined

$$\hat{F} = -\frac{1}{2} a (\phi_h^2)^- + a (\phi_h^2)^- = \frac{1}{2} a (\phi_h^2)^-,$$

and

$$\Theta = \frac{1}{2} a (\phi_h^2)^- + \frac{1}{2} a (\phi_h^2)^+ - a (\phi_h^-) (\phi_h^+) = \frac{1}{2} a (\phi_h^+ - \phi_h^-)^2 \geq 0$$

□

Proposition 7. (*Error estimate*)

Let ϕ be the exact solution to the PDE. define $e = \phi - \phi_h$. Then we have

$$\int_0^1 (e)^2(x, T) dx \leq O(h^{2k+2})$$

Proof. We have $B_j(\phi, v_h) = 0$ for all $v_h \in V_h$. Thus, we have

$$B_j(e, v_h) = 0$$

for all $v_h \in V_h$. We can rewrite e as

$$e = \phi - \phi_h = (\phi - \mathcal{P}_h\phi) + (\mathcal{P}_h\phi - \phi_h) = \varepsilon + e_h$$

where $e_h \in V_h$. Then we have

$$B_j(e, e_h) = 0$$

which gives us

$$B_j(e_h, e_h) = -B_j(\varepsilon, e_h) \tag{28}$$

The first proposition gives us

$$B_j(e_h, e_h) = \frac{1}{2} \int_{I_j} (e_h)^2 dx + \hat{F}_{j+\frac{1}{2}} - \hat{F}_{j-\frac{1}{2}} + \frac{1}{2} a [[e_h]]_{j-\frac{1}{2}}^2 - b \int_{I_j} (e_h)^2 dx.$$

And also

$$B_j(\varepsilon, e_h) = \int_{I_j} \varepsilon_t e_h dx - a \int_{I_j} \varepsilon (e_h)_x dx + a \varepsilon_{j+\frac{1}{2}}^- (e_h)_{j+\frac{1}{2}}^- - a \varepsilon_{j-\frac{1}{2}}^- (e_h)_{j-\frac{1}{2}}^+ - \int_{I_j} b \varepsilon e_h dx.$$

Now we define the projection \mathcal{P}_h as

$$\int_{I_j} (\mathcal{P}_h w - w) v_h dx = 0 \quad \forall v_h \in P^{k-1}(I_j)$$

and

$$(\mathcal{P}_h w - w)_{j+\frac{1}{2}}^- = 0 \quad \forall j$$

One can check that these two conditions give a unique projection. Then we can reduce $B_j(\varepsilon, e_h)$ to

$$B_j(\varepsilon, e_h) = \int_{I_j} \varepsilon_t e_h dx - \int_{I_j} b \varepsilon e_h dx.$$

We also get

$$\int_{I_j} \varepsilon_t^2 dx \sim O(h^{2k+2}).$$

We preserve this $k+1$ order property because the projection leaves polynomials of degree k untouched. After summing (28) over j we get

$$\frac{d}{dt} \frac{1}{2} \int_0^1 (e_h)^2 dx - b \int_0^1 (e_h)^2 dx \leq O(h^{2k+2}) + \frac{1}{2} \int_0^1 (e_h)^2 dx + O(h^{2k+2}) + \frac{1}{2} \int_0^1 (e_h)^2 dx$$

This implies that

$$\frac{d}{dt} \int_0^1 (e_h)^2 dx \leq (|b| + 2) \int_0^1 (e_h)^2 dx$$

Using Gronwall, we obtain

$$\int_0^1 (e_h)^2(x, T) dx \leq O(h^{2k+2})$$

and from here can get

$$\int_0^1 (e)^2(x, T) dx \leq O(h^{2k+2})$$

□