FINITE DIFFERENCES SCHEMES FOR PDE

Matthieu Gomez

February 1, 2016

Contents

1 Write Monotonous Scheme

- Barles Souganadis theorem Take the following PDE

$$0 = u_t + F(t, x, v, \partial v, \partial^2 v)$$

with F elliptic, i.e.if $X - Y$ positive semi definite $S(x, v, \partial v, X) \geq S(x, v, \partial v, Y)$
Denote a scheme a function

$$0 = S(\Delta a, v_j^{n+1}; ...)$$

A scheme is convergent if it is decreasing in all variables in ..., i.e. typically $v_{i+1}^{n+1}, v_{i-1}^{n+1}, v_{i+1}^{n}, v_i^n, v_{i+1}^n$ (i.e. all values except potentially v_j^{n+1})

- Interpretation in term of monte carlo chain For The PDE

$$\rho V = f(x, u) + \alpha(x, u)DV(x) + \sigma^2(x, u)D^2V(x)$$

A fully explicit finite difference schemes can be rewritten

$$V = V + \Delta f(x, u) + \sum p(x_k, y|u)V^h(y)$$

where $p(X_k, y|u) \geq 0$. An explicit monotonous scheme can be interpreted in term of a monte carlo chain. A monte carlo chain approximates the solution if it is consistent (denonting markov chain ξ_n^h)

$$E\Delta\xi = b(x)\Delta T$$

$$Var\Delta\xi = \sigma^2(x)\Delta T$$

1.1 Monotonous Scheme for first derivative

$$\rho V = f(V, x) + m(x)\partial V + a(x)\partial^2 V$$

How to satisfy monotonicity ?

- Monotinicity in v_{j+1} and v_{j-1} by upwinding

$$S = -\rho v_{n+1,j} + f(x_j) + \frac{v_{j+1} - v_j}{\Delta a} m(x)^+ - \frac{v_j - v_{j-1}}{\Delta a} m(x)^-$$

$$S = -\rho v_{n+1,j} + f(x_j) + \frac{v_j - v_{j-1}}{\Delta a} c_{j,B} - \frac{v_{j+1} - v_j}{\Delta a} (w + ra_j)$$

Economic interpretation: savings are positive, what matters is how the value function changes when wealth increases by a small amount; and vice versa when savings are negative. The right thing to do is therefore to approximate the derivative in the direction of the movement of the state.

- Monotinicity in v_j. v_j appears negatively in first and second derivative term. Thus does not satisfy Barles Souganadis theorem.

- One solution consists in adding a time derivative at the LHS. Then this counterbalances the negative term by a $\frac{1}{\Delta t}$. The issue is that, so that scheme is monotonous wrt v_t^n, one needs to bpick Δt low enough. This forces the time step to be so small that the rounding error dominates the total computational error
- Another solution consists in replacing everything by $n+t$ (explicit scheme)
  * If f is linear in v then replace by v^{n+1} here too
  * If f is monotone in v, then use v^n there. This includes case of optimal policy in v by envelop theorem.
- If f is non linear in v and non monotonous, this requires to use a non linear algorithm

- Second derivative
  - Naive approximation is

$$\partial_{ij} v = \frac{v_{i+1,j+1} + v_{i-1,j-1} - v_{i+1,j-1} - v_{i-1,j+1}}{4\Delta x^2}$$

    This is never monotonous because the term $v_{i+1,j-1}$ enters negatively and does not appear anywhere else
  - Better scheme
    * If $a_{ij}(x) \geq 0$

$$\begin{aligned}\partial_{ij} v = \frac{1}{2}(&\frac{v_{i+1,j+1} + v_{i-1,j-1} - 2v_{i,j}}{\Delta x_i \Delta x_j} \\ &- \frac{v_{i+1,j} + v_{i-1,j} - 2v_{i,j}}{(\Delta x_i)^2} \\ &- \frac{v_{i,j+1} + v_{i,j-1} - 2v_{i,j}}{(\Delta x_j)^2})\end{aligned}$$

    * if negative, a classical scheme is

$$\begin{aligned}\partial_{ij} v = \frac{1}{2}(&-\frac{v_{i+1,j-1} + v_{i-1,j+1} - 2v_{i,j}}{\Delta x_i \Delta x_j} \\ &+ \frac{v_{i+1,j} + v_{i-1,j} - 2v_{i,j}}{(\Delta x_i)^2} \\ &+ \frac{v_{i,j+1} + v_{i,j-1} - 2v_{i,j}}{(\Delta x_j)^2})\end{aligned}$$

    * This scheme is monotonous when the negative terms in $v_{i+1,j}$ and $v_{i,j+1}$ are compensated by the diagonal terms of the Hessian. More precisely, the scheme is monotonous iff

$$a_{ii}(x) \geq \sum |a_{ij}(x)|$$

- Rewrite in term of directional derivatives. Denote $a(x)$ such that the second order term is $\sum a_{i,j}(x)\delta_{ij}v$.
  * If $a = \sum \lambda_i \xi_i \xi_i$, the second order term can be rewritten only in term of directional derivatives:

    $$\sum \lambda_i D_\xi^2 v$$

    $D_\xi v$ is the directional derivative along the direction ξ

    $$D_\xi^2 v \approx \frac{v(t, x + \xi\Delta x) + v(t, x - \xi\Delta x) - 2v(x)}{|\xi|^2 \Delta x^2}$$

    The corresponding scheme is monotonous, similarly to the case with diagonal Hessian. How to find this decomposition?
  * Bonnans. Choose an order p and solve the problem

    $$\begin{aligned} &\min_{\lambda_i} ||a - \sum_{\xi \in \Xi_p} \lambda_i \xi_i \xi_i'||^2 \\ &\Xi_p = \{\xi \in Z^d \text{ s.t. } \max_i |\xi_i^d| \leq p\} \\ &\lambda_i \geq 0 \end{aligned}$$

    p needs to be equal to 5 so that mistake is not too large.
  * Me. For PDE obtained by HJB only. Denote $\sigma(s_t)$ the matrix such that $ds_t = \mu dt + \sigma dW_t$, the second order term has the form

    $$Tr(\sigma\sigma^T D^2 V)$$

    The matrix $\sigma\sigma^T$ is symetric therefore diagonalizable. There exists positive λ_i and directions ξ_i such that

    $$\sigma\sigma^T = \sum \lambda_i \xi_i \xi_i^T$$

    The issue is that $\xi \notin Z^d$. One can rewrite this as a scheme on the grid by using use a linear interpolation to approximate $v(t, x + \beta\Delta x)$ as a weighted average of points of the grid (with positive weights). One can always normalize v so that $v_1 = 1$. In particular, in two dimensions, one only needs to interpolate linearly along the second dimension, which simplifies the code (see image2015831195641.png)

1.2 Others

1.2.1 Optimization

$$\rho V = \max_u f(V, u) + \mu(x, u)\partial V + \frac{1}{2}\sigma^2(x, u)\partial^2 V$$

You need to make sure the terms that diappear in the envelop theorem do disappear in your setup.

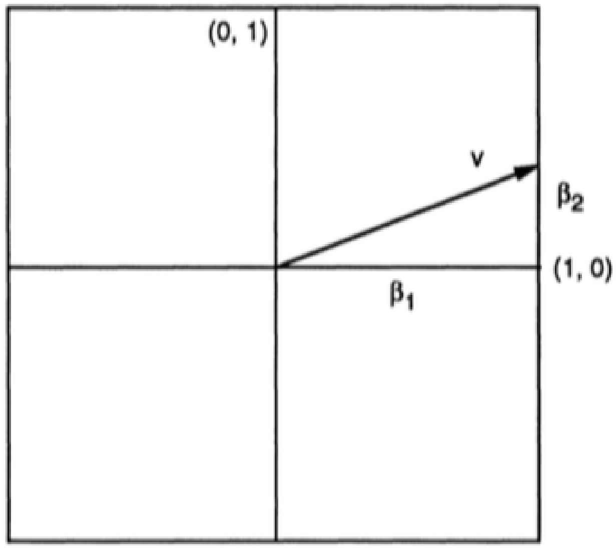

Figure 1: caption

- In the implicit scheme, the case of one state variable, this means that y

$$u'(c) = V_c c$$

  i.e. you need to use the same derivative to compute c so that the drift has the same sign

  Since v is generally concave in a, $s_F \leq s_B$, where s_F is the drift computed using forward difference for consumption and s_B the backward. Therefore the following situation happens: positive difference for ∂V when computing c gives negative drift, while positive difference gives positive drift). For these grid points, set the derivative of the value function equal to the case where drift is zero (i.e stationary value)

- This circularity becomes too complicated with multiple controls. The trick in this case is to divide a drift in two. Then when deriving and obtaing FOC, you can obtain differen backward/forwad for the derivative V_b in different FOC

  For instnce

$$V_a d = V_b d$$

  means that you must use the same backward/forward in V_a V_b and when computing d

$$u'_c = V_b c$$

means that you must hte same backward/forawrd in V_b when computing c. But if you split the drift in b in c and d they now can be different.

It is done in the Moll notes for the problem fixed cost

- If you have a semi explicit scheme, the term used when computing c_t is the past derivative while V_c is the future derivative. So I'm not sure why circular upwinding is important /sufficient

1.2.2 Discrete maximization

- Discrete maximiation is handled through splitting method (optimial time / fixed cost). Suppose there is fixed cost to convert illiquid asset a into liguid asset. Denote $v^*(w)$ the value function conditionally to paying the fixed cost. It is only a function of wealth.

$$
\begin{aligned}
v^*(a+b) &\equiv \max_{a',b'} v(a',b') \\
a' + b'* &= a + b - \kappa
\end{aligned}
$$

Then the HJB equation is the same, for v but we have new constraint

$$\forall a, b v(a,b) \geq v^*(a+b)$$

THis problem is solved in two step

$$
\begin{aligned}
v^{n+1/2} &\ldots \text{(usual HJB)} \\
v^{n+1}(a',b') &= \max\Big(\max_{a'+b'=a+b-\kappa} v^{n+1/2}(a',b'), v^{n+1/2}(a',b')\Big)
\end{aligned}
$$

1.2.3 Non linear PDE

$$\rho V = f(x,V) + m(x, \partial V(x))\partial V(x) + a(x, \partial V(x))\partial^2 V(x)$$

- Case of ditella μ_x depends on derivative ξ_ν etc. The same situation appears when substituting out optimal control by derivative, but in this case the envelop theorem does not apply
- Choose different direction in $\partial V(x)$ (even within a grid) for different terms, so that we make sure everything works

Is there existence theorem for the scheme?

1.2.4 Borders

- Upwind scheme has an adantage when using boundary conditions
  - saving is negative at the top of state space therefore forward difference is not used and no boundary condition needs to be imposed

- since upwind scheme selects a particular derivative when saving (ie drift) is negative, and since condition is about the drift, then we just replace the backward difference derivative by $u'(z + ra_1)$.

- First derivative $v_{I+1} - v_I$ is handled through upwiding
  - Exogeneous state constraint typically does not appear thanks to upwinding (i.e. when we're at the border of state, the drift is mean reverting)
  - True endogeneous state constraint (like $a \geq 0$) or artificial endogeneous state constraint (like $a \leq a_m ax$ - not really necessary but "sometimes helps numerical stability") Since FOC holds even at the constraint, a constraint of c is just a constraint on v'.

    $$v'(\underline{a}) \geq u'(z + r\underline{a})$$

    Note that this constraint binds iff the saving is negative. Thanks to upwind scheme, the boundary condition is therefore

    $$\partial v_1^b = u'(z + r\underline{a})$$

- Second derivative $v_{I+1} + v_{I-1} - 2v_I$
  - if exogeneous mean reverting state variable (like μ), use reflecting barrier $v'(x) = 0$ and therefore $v_{I+1} = v_I$.
  - if endogeneous state variable
    * policy funciton $c(a_{max})$ is approximately linear, and so from FOC this allows to express second derivative in term of first derivative

      $$v''(a_{max}) = -\gamma v'(a_{max})^{1+1/\gamma}\overline{c}$$

1.3 HJB

$$\hat{c}_t = \xi_t^{1-\psi}\rho^\psi$$

$$\sigma_w = \frac{\pi}{\gamma} - \frac{\gamma - 1}{\gamma}\sigma^\xi$$

$$\frac{\rho}{1 - \frac{1}{\psi}} = -\frac{\rho^\psi}{1-\psi}\xi^{1-\psi} + r_f + \sigma_w'\pi + \mu_\xi - \frac{\gamma}{2}(\sigma_w^2 + \sigma_\xi^2 - 2(1-\gamma)\sigma_w\sigma_\xi)$$

2 Method of line

The idea is to write the explicit system with t non discretized. So instead of writing

$$\frac{v_i^{n+1} - v_i^n}{\Delta t} = S(v^{n_i}, v^{n_{i-1}}, v^{n_{i+1}})$$

you solve the system of N first order ODE

$$\dot{v_i^n} = S(v^{n_i}, v^{n_{i-1}}, v^{n_{i+1}})$$