Operator Splitting Methods for the Advection-Diffusion-Reaction Equation

Sudhir Srivastava

October 15, 2008

Outline

Model Problem

Model Problem

$$
\begin{aligned}
u_t + \nabla.(\underline{a}u) &= \nabla.(D\nabla u) + f(u) && \forall (x,t) \in \ \Omega \text{ x } [0,T] \\
\partial_n u &= g_N && \forall (x,t) \in \ \partial\Omega_N \text{ x } [0,T] \\
u(x,t) &= g_D && \forall (x,t) \in \ \partial\Omega_D \text{ x } [0,T] \\
u(x,0) &= u_0 && \forall \ x \ \in \ \overline{\Omega}
\end{aligned}
$$

1-Dimension case

$$
\begin{aligned}
u_t + au_x &= \epsilon u_{xx} + \lambda u(1-u) && \forall (x,t) \in \ (0,1) \text{ x } [0,T] \\
u_x(0,t) &= 0 && \forall \ t \ \in \ [0,T] \\
u(1,t) &= (1+\sin(\omega t)/2 && \forall \ t \ \in \ [0,T] \\
u(x,0) &= 1/2 && \forall x \in \ [0,1]
\end{aligned}
$$

◂ ADR Splitting ◂ 1D Numerical Solution ◂ Dimension Splitting

Operator Splitting

Common splitting strategies are

- Algebraic/Time splitting
- Dimension Splitting

Algebraic/Time splitting

Nonlinear ODE system

$$u' = F(t, u(t)), \qquad t > 0, \qquad u(0) = u_0$$

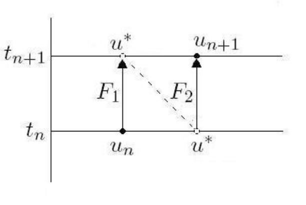
Solution :

$$u(t+\tau) = S_\tau(u(t))$$

Two term splitting

$$F(t, v) = F_1(t, v) + F_2(t, v)$$

First Order splitting



Let $u_n \approx u(t_n)$.

$$\begin{aligned} \frac{d}{dt}u^* &= F_1(t, u^*(t)) \quad for\ t_n < t \le t_{n+1}\ ,\ u^*(t_n) = u_n \\ \frac{d}{dt}u^{**} &= F_2(t, u^{**}(t)) \quad for\ t_n < t \le t_{n+1}\ ,\ u^{**}(t_n) = u^*(t_{n+1}) \\ u_{n+1} &= u^{**}(t_{n+1}) \end{aligned}$$

$$u_{n+1} = S_{2,\tau}(S_{1,\tau}(u_n))$$

Inserting exact solution $u(t)$
$\Rightarrow$

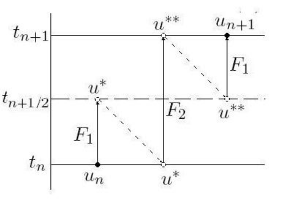
$$u(t+\tau) = S_{2,\tau}(S_{1,\tau}(u(t_n))) + \tau\rho_n$$

$\tau\rho_n$ is called ***local*** **splitting error**

$$\rho_n = \frac{\tau}{2}\left[\frac{\partial F_1}{\partial u}F_2 - \frac{\partial F_2}{\partial u}F_1\right](t_n, u(t_n)) + O(\tau^2)$$

If first term of expression vanishes then local truncation error is at least $O(\tau^2)$***.***

Second Order Symmetrical splitting



$$
\begin{aligned}
\frac{d}{dt}u^* &= F_1(t, u^*(t)) \quad for\ t_n < t \le t_{n+1/2}\ ,\ u^*(t_n) = u_n, \\
\frac{d}{dt}u^{**} &= F_2(t, u^{**}(t)) \quad for\ t_n < t \le t_{n+1}\ ,\ u^{**}(t_n) = u^*(t_{n+1/2}), \\
\frac{d}{dt}u^{***} &= F_1(t, u^{***}(t)) \quad for\ t_{n+1/2} < t \le t_{n+1}\ ,\ u^{***}(t_{n+1/2}) = u^{**}(t_{n+1}), \\
u_{n+1} &= u^{***}(t_{n+1})
\end{aligned}
$$

Second Order Symmetrical splitting

$\Rightarrow$

$$u_{n+1} = S_{1,\tau/2}(S_{2,\tau}(S_{1,\tau/2}(u_n)))$$

$$\rho_n = \frac{\tau^2}{24}\Bigg[\frac{\partial}{\partial w}\left(\frac{\partial F_1}{\partial w}F_1\right)F_2 - 2\frac{\partial}{\partial w}\left(\frac{\partial F_1}{\partial w}F_2\right)F_1 + \frac{\partial}{\partial w}\left(\frac{\partial F_2}{\partial w}F_1\right)F_1$$

$$-2\frac{\partial}{\partial w}\left(\frac{\partial F_2}{\partial w}F_2\right)F_1 + 4\frac{\partial}{\partial w}\left(\frac{\partial F_2}{\partial w}F_1\right)F_2 - 2\frac{\partial}{\partial w}\left(\frac{\partial F_1}{\partial w}F_2\right)F_2\Bigg] + O(\tau^4)$$

Evaluated at $t = t_{n+1/2},\ u = u(t_{n+1/2})$

If first term vanishes local truncation error is $O(\tau^4)$.

Lie Operator Formalism

For this discussion

$$u_t = \mathbf{f}(u), \qquad \mathbf{f}(u) = -\nabla.(\underline{a}u) + \nabla.(D\nabla u) + f(u)$$

Exact solution

$$u(t+\tau) = \mathbf{S}_\tau(u(t))$$

To any operator $\mathbf{f}$ acting on solution space $\mathcal{U}$ we can associate an operator $\mathcal{F}$ called **Lie Operator**.

$$\mathbf{g} \mapsto \mathcal{F}\mathbf{g} \qquad \forall\ \mathbf{g} \text{ on } \mathcal{U}$$

$$\mathcal{F}\mathbf{g}(v) := \mathbf{g}'(v)\mathbf{f}(v) \qquad \forall v \in \mathcal{U}$$

for the solution $u(t)$ we have

$$\mathcal{F}\mathbf{g}(u(t)) = \frac{\partial}{\partial t}\mathbf{g}(u(t)).$$

Lie Operator Formalism

Similarly for $k > 1$

$$\mathcal{F}^k \mathbf{g}(u(t)) = \frac{\partial^k}{\partial t^k} \mathbf{g}(u(t))$$

Using exponential form of Lie operator

$$\begin{aligned} e^{\tau \mathcal{F}} \mathbf{g}(u(t)) &= \sum_{k=0}^{\infty} \frac{\tau^k}{k!} \mathcal{F}^k \mathbf{g}(u(t)) \\ &= \sum_{k=0}^{\infty} \frac{\tau^k}{k!} \frac{\partial^k}{\partial t^k} \mathbf{g}(u(t)) \\ &= \mathbf{g}(u(t+\tau)) \\ &= \mathbf{g}(\mathbf{S}_\tau(u(t))) \end{aligned}$$

formally we have

$$e^{\tau \mathcal{F}} \mathbf{g}(\cdot) = \mathbf{g}(\mathbf{S}_\tau(\cdot))$$

$$e^{\tau \mathcal{F}} \equiv \mathbf{S}_\tau \quad i.e. \; u(t_{n+1}) = e^{\tau \mathcal{F}}(u(t_n))$$

.

Lie Operator Formalism

Consider Splitting

$$\mathbf{f} = \mathbf{f}_1 + \mathbf{f}_2$$

$$u_t = \mathbf{f}_k(u) \qquad \begin{array}{ccc} \mathbf{f}_k & \longrightarrow & \mathbf{S}_{k,\tau} \\ \downarrow & & \\ \mathcal{F}_k & & \end{array}$$

$$\mathcal{F} = \mathcal{F}_1 + \mathcal{F}_2$$

First order splitting Let $u_n \approx u(t_n)$

$$u_{n+1} = e^{\tau \mathcal{F}_1} e^{\tau \mathcal{F}_2}(u_n)$$

$$u(t_{n+1}) = e^{\tau \mathcal{F}_1} e^{\tau \mathcal{F}_2}(u(t_n)) + \tau \rho_n$$

where $\tau \rho_n$ is *splitting error* and ρ_n local truncation error given as

$$\rho_n = \frac{1}{\tau}\left(e^{\tau \mathcal{F}} - e^{\tau \mathcal{F}_1} e^{\tau \mathcal{F}_2}\right)(u(t_n)) = \frac{\tau}{2}[\mathcal{F}_2, \mathcal{F}_1] + O(\tau^2)$$

$$[\mathcal{F}_2, \mathcal{F}_1] := \mathcal{F}_2 \mathcal{F}_1 - \mathcal{F}_1 \mathcal{F}_2$$

Remark : Commutator is also a Lie operator.

Second order splitting

$$u_{n+1} = e^{\frac{1}{2}\tau\mathcal{F}_1} e^{\tau\mathcal{F}_2} e^{\frac{1}{2}\tau\mathcal{F}_1}(u_n)$$

$$\rho_n = \frac{1}{24}\tau^2([\mathcal{F}_1,[\mathcal{F}_1,\mathcal{F}_2]] + 2[\mathcal{F}_2,[\mathcal{F}_1,\mathcal{F}_2]])u(t_{n+1/2}) + O(\tau^4)$$

Theorem : *Splitting error is zero if commutator* $[\mathcal{F}_2, \mathcal{F}_1]$ *is zero.*

Particular case when PDE gives Linear ODE system

Preposition : $e^{\tau\mathcal{F}_1}e^{\tau\mathcal{F}_2}\mathbf{g}(.) = \mathbf{g}((\mathbf{S}_{2,\tau}(\mathbf{S}_{1,\tau}(.)))$

Particular case of Linear ODE:

$$w'(t) = \mathrm{A}w(t)$$

Let Lie operator associated with A is $\mathcal{A}$

$$w(t_{n+1}) = e^{\tau\mathcal{A}}w(t_n) \qquad w(t_{n+1}) = e^{\tau\mathrm{A}}w(t_n)$$

$$\mathcal{A} = \mathcal{A}_1 + \mathcal{A}_2 \qquad \mathrm{A} = \mathrm{A}_1 + \mathrm{A}_2$$

$$w_{n+1} = e^{\tau\mathcal{A}_1}e^{\tau\mathcal{A}_2}I(w_n) \qquad w_{n+1} = e^{\tau\mathrm{A}_2}e^{\tau\mathrm{A}_1}(w_n)$$

$$w_{n+1} = e^{\tau\mathcal{A}_1}e^{\tau\mathcal{A}_2}(w_n)$$

$$\text{if } [\mathcal{A}_1, \mathcal{A}_2] = 0 \quad \Rightarrow \quad \text{No splitting error}$$

Advection-Diffusion-Reaction Splitting

Consider our ▸ Model Problem. we are going to consider splitting of Advection, Diffusion from Reaction,*i.e.*

$$u_t = \mathbf{f}_{AD}(u) + \mathbf{f}_R(u), \quad \mathbf{f}_{AD} = -\nabla.(\underline{a}u) + \nabla.(D\nabla u), \quad \mathbf{f}_R = f(u)$$

then the commutator will be given as

$$[\mathcal{F}_{AD}, \mathcal{F}_R]I(u) = 0 \;\Leftrightarrow\; [\mathbf{f}_{AD}, \mathbf{f}_R]u = 0$$

where

$$[\mathbf{f}_{AD}, \mathbf{f}_R]u = \mathbf{f}'_{AD}(u)\mathbf{f}_R(u) - \mathbf{f}'_R(u)\mathbf{f}_{AD}(u)$$

Advection-Diffusion-Reaction Splitting

we have

$$\mathbf{f}'_{AD}\mathbf{f}_R = -\nabla.(\underline{a}f(u)) + \nabla.(D.\nabla f(u))$$

and

$$\mathbf{f}'_R\mathbf{f}_{AD} = -f'(u)(\underline{a}.\nabla u) - f'(u)(\nabla.\underline{a})u + f'(u)(\nabla.(D\nabla u))$$

Hence
Operator advection diffusion commutes with reaction if f(u) is linear in u and independent of $\underline{x}$.

Splitting error

$$\rho_n = \frac{\tau}{2}\left[(\nabla.\underline{a})(f(u) - f'(u)u + \nabla.(D.\nabla f(u)) - f'(u)(\nabla.(D\nabla u))\right](t_n) + O(\tau^2)$$

Numerical Experiments

Consider our ▸ 1D Model Problem

Remark : $[\mathbf{f}_{AD}, \mathbf{f}_R] \neq 0$

First order splitting

Second order splitting

$$a = 10, \epsilon = 10, \lambda = 1$$

Implicit Euler for advection diffusion sub problem and Explicit Runge Kutta-4 for reaction sub problem

Numerical Experiments

First order splitting

Second order splitting

$$a = 10, \epsilon = 10, \lambda = 20$$

Implicit Euler for advection diffusion sub problem and Explicit Runge Kutta-4 for reaction sub problem

Numerical Experiments

First order splitting

Second order splitting

$$a = 10, \epsilon = 10, \lambda = 20$$

Implicit Euler for advection diffusion sub problem and Implicit Euler for reaction sub problem

Conclusions

.

- We can use Implicit methods for advection diffusion terms and Explicit for reaction if its non-stiff.

Conclusions

.

- We can use Implicit methods for advection diffusion terms and Explicit for reaction if its non-stiff.
- Decoupling simplify the numerical solution process.

Conclusions

.

- We can use Implicit methods for advection diffusion terms and Explicit for reaction if its non-stiff.
- Decoupling simplify the numerical solution process.
- Massive room for parallel computation.

Conclusions

.

- We can use Implicit methods for advection diffusion terms and Explicit for reaction if its non-stiff.
- Decoupling simplify the numerical solution process.
- Massive room for parallel computation.
- For inhomogeneous boundary conditions second order splitting is affected by large commutators.

Conclusions

.

- We can use Implicit methods for advection diffusion terms and Explicit for reaction if its non-stiff.
- Decoupling simplify the numerical solution process.
- Massive room for parallel computation.
- For inhomogeneous boundary conditions second order splitting is affected by large commutators.
- Order of local truncation error is the order of splitting.

Conclusions

.

- We can use Implicit methods for advection diffusion terms and Explicit for reaction if its non-stiff.
- Decoupling simplify the numerical solution process.
- Massive room for parallel computation.
- For inhomogeneous boundary conditions second order splitting is affected by large commutators.
- Order of local truncation error is the order of splitting.
- Multistep methods ?

Dimension Splitting

CFL condition to solving our model problem in d-dimensional is

$$\sum_{k=1}^{d} \frac{\tau}{h_k}|a_k| \leq C_0 \qquad C_0 > 0$$

splitting our model problem in spatial dimensions we get CFL condition

$$\max_k \left(\frac{\tau}{h_k}|a_k| \right) \leq C_0$$

Remark : *number of commutators in dimensional splitting is d(d-1)/2.*

Dimension Splitting

First order splitting : Consider our ▸ Model Problem in 2-dimension with

$$g_N = 0 \qquad \partial\Omega_N = \{(x,y) : x = 0, 0 < y < 1 \text{ and } 0 < x < 1, y = 0\}$$

$$g_D = 1/2 \qquad \partial\Omega_D = \{(x,y) : x = 1, 0 < y < 1 \text{ and } 0 < x < 1, y = 1\}$$

$$u_t = \underbrace{-(a_1 u)_x + (d_1 u_x)_x}_{\mathbf{f}_1} - \underbrace{(a_2 u)_y + (d_2 u_y)_y}_{\mathbf{f}_2}$$

$$[\mathbf{f}_1, \mathbf{f}_2] = 0$$

$$\Updownarrow$$

$$(a_1)_y = (a_2)_y = (d_1)_y = (d_2)_x = 0$$

Dimension Splitting

Basic first order splitting used with the sequence

$$u^*(x, y, t_n) = u(x, y)_n$$

$$u_t^* = -(a_1 u^*)_x + (d_1 u_x^*)_x, \quad for\ t_n < t \leq t_{n+1}$$

above two holds for all but fixed y

$$u^{**}(x, y, t_n) = u^*(x, y)_{n+1}$$

$$u_t^{**} = -(a_2 u^{**})_y + (d_2 u_y^{**})_y, \quad for\ t_n < t \leq t_{n+1}$$

above two holds for all but fixed x
finally

$$u(x, y)_{n+1} = u^{**}(x, y, t_{n+1})$$

Numerical Experiments

For our numerical solution we used a=[10,100] and D $= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$. For this particular case $[\mathbf{f}_1, \mathbf{f}_2] = 0$

continuous initial value | Discontinuous initial value

$$u(x,0) = \begin{cases} 1/2 & \text{if } x + y \geq 1, \\ x/2 + y/2 & \text{otherwise.} \end{cases} , \quad u(x,0) = \begin{cases} 0 & 0 < x, y < 1/2, \\ 1/2 & \text{otherwise.} \end{cases} .$$

Conclusions

.

- We only have to solve tridiagonal system of algebraic equations.

Conclusions

.

- We only have to solve tridiagonal system of algebraic equations.
- Good possibilities of parallel computation.

Conclusions

.

- We only have to solve tridiagonal system of algebraic equations.
- Good possibilities of parallel computation.
- Zero commutator conditions are not satisfied in general.

Conclusions

.

- We only have to solve tridiagonal system of algebraic equations.
- Good possibilities of parallel computation.
- Zero commutator conditions are not satisfied in general.
- Splitting is not possible if D is full matrix.

Questions ?

Thank you !