

Difference algebra aided discretization of quasilinear evolution equations

V.P. Gerdt¹, Yu.A. Blinkov², K.B. Marinov³

¹ Joint Institute for Nuclear Research, Dubna, Russia, gerdt@jinr.ru

² Saratov State University, Saratov, Russia, blinkovua@info.sgu.ru

³ University "Dubna", Dubna, Russia, marinov.kohctahtih@gmail.com

Let ∂_x be the derivation operator w.r.t. x and $\mathcal{R} := \mathbb{Q}(a_1, \dots, a_i)\{u\}$ be the ordinary differential polynomial ring over the parametric field $\mathbb{Q}(a_1, \dots, a_i)$ of constants. Here we consider a class of quasilinear evolution equations of the form

$$u_t = au_m + F(u_{m-1}, \dots, u_1, u), \quad 0 \neq a \in \mathbb{Q}, \quad m \in \mathbb{N}_{>0}, \quad (1)$$

where $u_k := \partial_x^k u$ ($0 \leq k \leq m$), $u_0 := u$ and $F \in R$ is a differential polynomial of the order $m-1$ in ∂_x (denotation: $\text{ord}(F) = m-1$) such that there is a differential polynomial $P \in \mathcal{R}$ satisfying

$$F = \partial_x P = \sum_{k=0}^{m-2} u_{k+1} \frac{\partial P}{\partial u_k}. \quad (2)$$

Given F , one can algorithmically verify whether or not such P exists and construct it in the case of existence. The equality (2) means that (1) admits the conservation law form

$$u_t = \partial_x (au_{m-1} + P), \quad P \in \mathcal{R}, \quad \text{ord}(P) = m-2. \quad (3)$$

The set of evolution equations admitting the polynomial conservation law (3) contains most of classical evolution equations, e.g., the Kortevog-de Vries (KdV) equation and KdV hierarchy, the Burgers equation and Burgers hierarchy, the Kuramoto-Sivashinsky equation, the Burgers-Huxley equation, etc., and their various generalizations (cf. [1]). All these equations have exact solutions that are useful in analysis of numerical methods for their solving.

To discretize equation (3), we follow the approach of paper [2] and convert the equation into the equivalent integral form

$$\oint_{\Gamma} (P + au_{m-1}) dt + u dx = 0, \quad (4)$$

where Γ is an arbitrary singly connected integration contour. Using the standard notation $u_j^n = u(t_n, x_j)$ for the grid function and the Cartesian grid with $t_{n+1} - t_n = \tau$, $x_{j+1} - x_j = h$ we choose the rectangular integration contour as a "control volume" (cf. [2]) and add $m-2$ integral relations

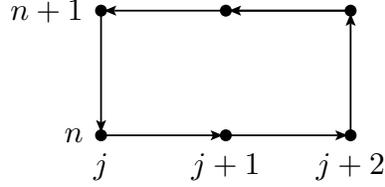


Figure 1: Basic integration contour

$$\int_{x_j}^{x_{j+1}} u_{k+1} dx = u_k(t, x_{j+1}) - u_k(t, x_j), \quad k = 1, \dots, m-2. \quad (5)$$

Now, to discretize (4) we apply a numerical evaluation method to the contour integral (4) in order to express it in terms of the grid functions and also (possibly different) numerical evaluation methods to the integrals in the left-hand sides of (5). Thereby, we obtain a system of difference equations containing $u_j^n, u_{1j}^n, \dots, u_{m-1j}^n$. The last step in generation of a finite difference approximation (FDA) to (1) is algebraic elimination of the grid functions $u_{1j}^n, \dots, u_{m-1j}^n$, which correspond to the proper partial derivatives of u , from the discrete system obtained. Such elimination can be done by means of the MAPLE package *LDA* [3] which is freely available (<http://wwwb.math.rwth-aachen.de/Janet/>).

We illustrate the above described approach by example of the KdV equation

$$u_t + u_{xxx} + 6uu_x = 0. \quad (6)$$

Its integral conservation law form for the contour Γ of Figure 1 reads

$$\oint_{\Gamma} (u_{xx} + 3u^2) dt + u dx = 0. \quad (7)$$

To approximate numerically the contour integral, we apply the trapezoidal rule to the integration over t as well as to the integration over x . For numerical approximations of the integral relations we apply the trapezoidal rule for the integration of u_x and the midpoint rule for the integration of u_{xx} . This leads to the difference approximation to (6) which is outputted by the following MAPLE code shown in Figure 3 with $P := 3u^2$. The output is the left-hand side of the FDA to (6) written in the conventional form as

$$\begin{aligned} & \frac{u_j^{n+1} - u_j^n}{\tau} + \frac{(P_{j+1}^{n+1} - P_{j-1}^{n+1}) + (P_{j+1}^n - P_{j-1}^n)}{4h} + \\ & + \frac{(u_{j+2}^{n+1} - 2u_{j+1}^{n+1} + 2u_{j-1}^{n+1} - u_{j-2}^{n+1}) + (u_{j+2}^n - 2u_{j+1}^n + 2u_{j-1}^n - u_{j-2}^n)}{4h^3} = 0. \end{aligned}$$

```

> with(LDA):
> L:= [ (- (P(n,j)+P(n+1,j)-P(n,j+2)-P(n+1,j+2)) -
      (uxx(n,j)+uxx(n+1,j)-uxx(n,j+2)-uxx(n+1,j+2))) *tau/2+
      (u(n+1,j+1)-u(n,j+1)) *2*h,
      (ux(n,j+1)+ux(n,j)) *h/2- (u(n,j+1)-u(n,j)),
      2*uxx(n,j+1)*h- (ux(n,j+2)-ux(n,j)) ]:
> JanetBasis(L, [n,j], [uxx,ux,u,P], 2):
> collect(%[1,1]/(4*tau*h**3), [tau,h]):

$$\frac{\frac{1}{4} P(n+1, j+3) + \frac{1}{4} P(n, j+3) - \frac{1}{4} P(n+1, j+1) - \frac{1}{4} P(n, j+1)}{h}$$


$$+ \frac{1}{h^3} \left( \frac{1}{4} u(n+1, j+4) - \frac{1}{2} u(n+1, j+3) + \frac{1}{4} u(n, j+4) - \frac{1}{2} u(n, j+3) + \frac{1}{2} u(n+1, j+1) - \frac{1}{4} u(n+1, j) + \frac{1}{2} u(n, j+1) - \frac{1}{4} u(n, j) \right)$$


$$+ \frac{u(n+1, j+2) - u(n, j+2)}{\tau}$$


```

Figure 2: Construction of FDA to KdV with MAPLE

Since the obtained FDA to (6) has quadratic nonlinearity (due to $P = 3u^2$) in the grid function on the next time layer, in order to construct a numerical solution we use the following approximate linearization

$$v_{k+1}^2 = v_{k+1}^2 - v_k^2 + v_k^2 = (v_{k+1} - v_k)(v_{k+1} + v_k) + v_k^2 \approx v_{k+1} \cdot 2v_k - v_k^2.$$

By taking this linearization into account, we implemented a numerical procedure for construction of a solution to KdV in Python 2.7 freely downloadable from the Web page <https://www.python.org/download/releases/2.7/>. Figure 4 demonstrates the time evolution of numerical solution in the domain $x \in [0, 200]$ with $h = 0.4$ and $\tau = 0.2$ and for the initial value (Cauchy) problem with the initial data

$$u(t=0) := f(x, 0, 0.4) + f(x-20, 0, 0.2)$$

where

$$f(x, t, \kappa) := \frac{2\kappa^2}{\cosh^2[\kappa(x - 4\kappa^2 t)]}$$

is the exact one-soliton solution to (6). As Figure 4 shows, the constructed numerical solution reveals a behavior inherent to localized solutions of KdV.

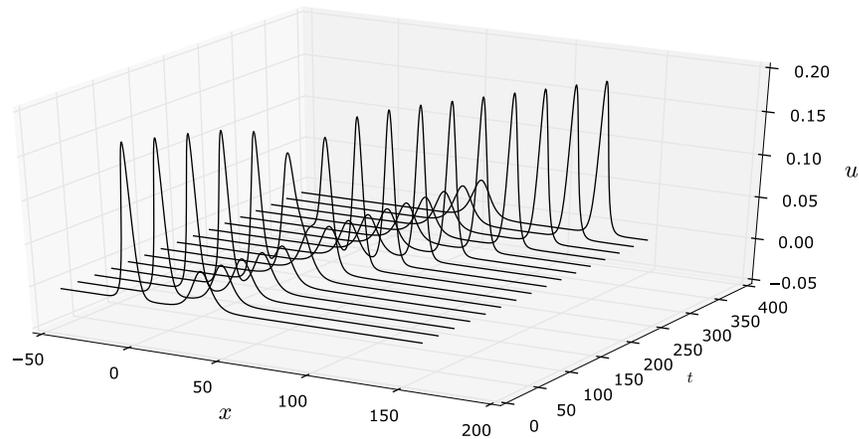


Figure 3: Dynamics of solution to KdV

References

- [1] N.A.Kudryashov. Methods of Nonlinear Mathematical Physics. Publishing House “Intellect”, 2010 (in Russian).
- [2] V. P. Gerdt, Yu. A. Blinkov, and V. V. Mozzhilkin. Gröbner Bases and Generation of Difference Schemes for Partial Differential Equations. *Symmetry, Integrability and Geometry: Methods and Applications*, 2:051, 2006, 26 pages. arXiv:mathRA/0605334
- [3] V. P. Gerdt and D. Robertz. A Maple Package for Computing Gröbner Bases for Linear Recurrence Relations. *Nuclear Instruments and Methods in Physics Research*, A559:215–219, 2006. arXiv:cs.SC/0509070.