DIDACTICA MATHEMATICA, Vol. 33(2015), pp. 17–26

REZOLVAREA NUMERICĂ A ECUAŢIILOR CU DERIVATE PARŢIALE FOLOSIND METODA LINIILOR

Imre Boros

Abstract. This paper discusses the numerical solution of partial differential equations using method of lines approach (MOL) where the spatial dimension is discretized using some finite difference approximation leaving the time dimension to be the only independent variable in the resulting system of initial value problems. Once this is done, we apply an integration algorithm for the initial value ordinary differential equations to compute an approximate numerical solution to the partial differential equation.
MSC 2000. 65M20.
Key words. Method of lines, error estimates.

1. ECUAŢII CU DERIVATE PARŢIALE

Fenomenele din lumea fizică pot fi modelate cel mai bine cu ajutorul ecuaţiilor cu derivate parţiale. Ca o consecinţă metode de rezolvare a acestor ecuaţii ca şi metoda liniilor sunt de interes larg în ştiinţă şi inginerie.

Considerăm exemplul ilustrativ de ecuaţie cu derivate parţiale

$$\frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial x^2} \tag{1}$$

unde

- $u = u(t, x)$ variabilă dependentă
- t variabilă independentă care reprezintă timpul
- x variabilă independentă care reprezintă spaţiul

Observăm că ecuaţia (1) are două variabile independente, tocmai din cauza asta este clasificată ca şi o ecuaţie cu derivate parţiale. O ecuaţie diferenţială cu o singură variabilă independentă se numeşte ecuaţie diferenţială ordinară. O să considerăm mai târziu ecuaţiile diferenţiale ordinare ca o parte a metodei liniilor. Ecuaţia (1) se numeşte ecuaţia căldurii, iar constanta pozitivă D este difuzivitatea termică.

2. CONDIŢII INIŢIALE ŞI CONDIŢII LA FRONTIERĂ

Înainte să rezolvăm ecuaţia (1), trebuie să specificăm condiţii auxiliare pentru a completa enunţul problemei cu derivate parţiale. Numărul necesar de condiţii este determinat de ordinul cel mai mare a derivatei în fiecare variabilă. Cum (1) este de ordinul unu în t şi de ordinul doi în x va fi necesar o condiţie în t şi două condiţii în x.

Variabila t se mişcă pe un interval finit $t_0 \leq t \leq t_f$ sau poate să fie şi infinit $0 \leq t \leq \infty$, este necesară o valoare iniţială, pentru că se începe în t_0 şi avansează în timp, fără alte condiţii asupra lui.

Pentru variabila spaţială $x_0 \leq x \leq x_f$ se pun două condiţii care de obicei corespund frontierelor unor sisteme fizice, tocmai de aici vine denumirea de condiţii pe frontieră.

Exemple de condiţii auxiliare pentru ecuaţia (1),

- O condiţie iniţială poate fi

$$u(x, t_0) = u_0(x) \tag{2}$$

  unde u_0 este o funcţie dată.
- Condiţii pe frontieră pot fi de forma

$$u(x_0, t) = u_b \tag{3}$$
$$\frac{\partial u(x_f, t)}{\partial x} = 0 \tag{4}$$

  unde u_b este o frontieră dată a lui u pentru orice t.

Condiţiile pe frontieră pot fi de trei tipuri:

(1) Dacă este de forma (3), atunci se numeşte condiţie Dirichlet.
(2) Dacă este de forma (4) , atunci se numeşte condiţie Neumann.
(3) Iar dacă intervin (3) şi (4) atunci se numeşte condiţie de tip Robin.

Relaţiile (1),(2),(3) şi (4) formează o ecuaţie cu derivate parţiale completă. O soluţie pentru ecuaţia cu derivate parţiale este o funcţie care defineşte variabila dependentă ca funcţie de variabilele independente, în cazul nostru $u(x, t)$. Soluţia poate să fie de două tipuri

- Soluţie analitică dacă soluţia este o funcţie matematică. Soluţiile analitice sunt exacte, dar sunt foarte dificil de obţinute chiar şi pentru cele mai simple ecuaţii cu derivate parţiale.
- Soluţie numerică dacă funcţia $u(x, t)$ este dat numeric într-un tabel. Soluţiile numerice sunt aproximaţii pentru soluţiile analitice, noi vrem ca aceste soluţii numerice să fie apropiate de cele analitice.

Cu metoda liniilor putem obţine soluţii numerice care aproximează soluţia analitică cu acurateţe foarte bună.

3. METODA LINIILOR

Ideea de bază este să înlocuim derivatele spaţiale în ecuaţii cu derivate parţiale cu aproximări algebrice. În acest mod derivatele spaţiale numai apar în formă explicită, numai variabila de timp rămâne. Cu alte cuvinte ne rămâne un sistem de ecuaţii diferenţiale ordinare care aproximează ecuaţia cu derivate parţiale. Provocarea este să formulăm acest sistem pentru aproximarea ecuaţiei cu derivate parţiale. Când avem sistemul putem să aplicăm orice metodă numerică pentru aproximarea soluţiei. Un avantaj a metodei liniilor este că putem să folosim metode existente şi bine cunoscute pentru ecuaţii diferenţiale ordinare.

Pentru a ilustra metoda considerăm ecuaţia hiperbolică de ordin unu

$$\frac{\partial u}{\partial t} + v\frac{\partial u}{\partial x} = 0. \tag{5}$$

Prima dată înlocuim derivata spaţială cu o aproximare cu diferenţe finite

$$\frac{\partial u}{\partial x} \approx \frac{u_i - u_{i-1}}{\Delta x} \tag{6}$$

unde am considerat o grilă echidistantă cu pasul Δx, cu punctele de pe grilă $x_i = (i-1)\Delta x, i = 1, \ldots, M$. Aproximarea cu metoda liniilor pentru ecuaţia (5) este

$$\frac{du_i}{dt} = -v\frac{u_i - u_{i-1}}{\Delta x}, \qquad 1 \le i \le M. \tag{7}$$

Se observă că (7) este un sistem de ecuaţii diferenţiale pentru că numai t a rămas ca şi variabilă independentă. Această transformare a lui (5) în (7) ilustrează esenţa metodei liniilor. Apoi pentru a calcula soluţia pentru ecuaţia cu derivate parţiale rezolvăm sistemul cu ecuaţii diferenţiale ordinare. Dar înainte de asta trebuie să ţinem cont de condiţia iniţială şi condiţia pe frontieră pentru (5) care pot fi luate de forma

$$u(x,0) = f(x) \tag{8}$$

$$u(0,t) = g(x). \tag{9}$$

Cum sistemul (7) este format din M ecuaţii, sunt necesare M condiţii iniţiale şi din (8) aceste sunt

$$u(x_i,0) = f(x_i), \qquad 1 \le i \le M \tag{10}$$

Iar dacă folosim condiţia de frontieră (9) găsim pentru primul punct din grilă $i = 1$

$$u(x_1, t) = g(t) \tag{11}$$

Relaţiile (7), (10),(11) reprezintă metoda liniilor completă pentru ecuaţia (5) cu condiţiile (8) şi (9). Soluţia pentru acest sistem ne dă M funcţii

$$u_1(t), u_2(t), \ldots, u_M(t) \tag{12}$$

care sunt aproximaţii pentru $u(x,t)$ în punctele din grilă $x_1, x_2, \ldots, x_M$.

4. INTEGRAREA ECUAŢIILOR DIFERENŢIALE ÎN METODA LINIILOR

Considerăm acum integrarea numerică a sistemului de ecuaţii diferenţiale (7). Dacă derivata $\frac{du_i}{dt}$ este aproximat cu diferenţă finită progresivă de ordin unu

$$\frac{du_i}{dt} \approx \frac{u_i^{n+1} - u_i^n}{\Delta t} + O(\Delta t), \tag{13}$$

unde n este index pentru variabila t pe un grid echidistant cu pasul Δt, atunci aproximarea cu diferenţă finită de ordinul unu pentru sistemul (7) este

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = -v\frac{u_i^n - u_{i-1}^n}{\Delta x} \tag{14}$$

sau în formă explicită

$$u_i^{n+1} = u_i^n - v\frac{\Delta t}{\Delta x}(u_i^n - u_{i-1}^n), \quad i = 1, 2, \ldots, M. \tag{15}$$

Ecuaţia (15) dă explicit pe u_i^{n+1}, adică putem să calculăm soluţiile pe punctul t_{n+1} din soluţiile de pe punctul t_n. Metoda obţinută astfel pentru sistemul de ecuaţii diferenţiale (7) este metoda lui Euler progresivă, care este cea mai cunoscută şi cea mai simplă pentru integrarea sistemelor de ecuaţii diferenţiale ordinare.

Forma explicită a metodei (15) este convenabil din punct de vedere computaţional, dar are alte limitări. Dacă pasul de timp Δt este peste o valoare critică atunci metoda nu va fi stabilă. De fapt pentru ca metoda (15) să rămână stabilă, trebuie ca $(v\Delta t/\Delta x)$ care se numeşte numărul lui Courant-Friedrichs-Lewy, să fie mai mică decât 1. Dacă vrem să creştem acurateţea soluţiei micşorând pasul Δx atunci ca numărul C-F-L să rămână mai mic ca 1 trebuie să micşorăm şi pasul Δt. Există un conflict între îmbunătăţirea acurateţei şi menţinerea stabilităţii metodei.

O cale de a îmbunătăţii metoda (15) este să considerăm metoda lui Euler regresivă, care este obţinută prin folosirea diferenţei finite regresive

$$\frac{du_i}{dt} \approx \frac{u_i^n - u_i^{n-1}}{\Delta t} + O(\Delta t), \tag{16}$$

astfel obţinem aproximarea pentru (7)

$$\frac{u_i^n - u_i^{n-1}}{\Delta t} = -v\frac{u_i^n - u_{i-1}^n}{\Delta x}$$

sau dacă notăm $\alpha = v\Delta t/\Delta x$ avem

$$(1+\alpha)u_i^n + \alpha u_{i-1}^n = u_i^{n-1}, i = 1, 2, \ldots, M \tag{17}$$

Observăm că acum nu putem obţine explicit pe u_i^n în funcţie de de soluţiile pe pasul anterior u_i^{n-1}, ecuaţia (17) este implicită în u_i^n pentru că şi u_{i-1}^n este necunoscut. Pentru a rezolva ecuaţia (17) trebuie să considerăm toate ecuaţiile $1, 2, \ldots, M$ de unde ajungem la un sistem bidiagonal. Metoda (17) nu are limite de stabilitate dar din punct de vedere computaţional este mult mai greu de abordat, pentru că trebuie să rezolvăm un sistem de ecuaţii algebrice, iar dacă sistemul de ecuaţii ordinare care aproximează ecuaţia cu derivate parţiale este neliniară atunci trebuie să rezolvăm sisteme neliniare. Sistemele de ecuaţii de obicei sunt rezolvate cu o variantă a metodei lui Newton care poate deveni foarte solicitant.

5. APLICAŢIE PENTRU ECUAŢIA CĂLDURII

Considerăm ecuaţia (1) cu $D = 1$

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} \tag{18}$$

cu condiţia iniţială

$$u(x,0) = \sin\left(\frac{\pi x}{2}\right). \tag{19}$$

Luăm condiţiile pe frontieră

$$u(0,t) = 0 \tag{20}$$

$$\frac{\partial u(1,t)}{\partial x} = 0 \tag{21}$$

Soluţia analitică pentru ecuaţia cu derivate parţiale (18)-(21) este

$$u(x,t) = e^{-(\pi^2/4)t}\sin\left(\frac{\pi x}{2}\right). \tag{22}$$

În continuare vom rezolva numeric ecuaţia cu derivate parţiale (18)-(21) în Matlab cu două solvere cu ode15s care este implementat în Matlab şi este folosit pentru rezolvarea problemelor stiff. Celălalt solver face parte din librăria Sundials care a fost implementat în limbajul de programare C, limbajul C foloseşte pointeri pentru gestionarea memoriei şi din cauza asta de obicei programele scrise în $C/C{+}{+}$ sunt cele mai rapide. Din această librărie am folosit funcţia $CVODE$ care foloseşte metode de ordin variabil, cu mai mulţi paşi şi cu pas variabil, care sunt bazate pe formule de forma

$$\sum_{i=0}^{K_1} \alpha_{n,i} y_{n-i} + h_n \sum_{i=0}^{K_2} \beta_{n,i} f(t_{n-i}, y_{n-i}) = 0. \tag{23}$$

Unde y_n sunt aproximaţii pentru $y(t_n)$, iar $h_n = t_n - t_{n-1}$ este pasul. În **CVODE** putem alege metode de tip Adams-Moulton care sunt caracterizate de $K_1 = 1$ şi $K_2 = q$, unde ordinul q variază între 1 şi 12. Iar pentru probleme stiff putem alege formule cu diferenţe retrograde care sunt caracterizate de $K_1 = q$ şi $K_2 = 0$ cu ordinul q între 1 şi 5.

Alegem intervalul de timp $t \in (0, 2.5)$ şi discretizarea ei cu $n = 21$ de puncte. Pentru condiţia iniţială a sistemului (19) construim vectorul cu valorile iniţiale în Matlab cu un ciclu for

```
for i=1:n
    u0(i)=sin((pi/2.0)*(i-1)/(n-1));
end
```

Apoi construim funcţia f a sistemului de ecuaţii diferenţiale care aproximează ecuaţia cu derivate parţiale (18) prin

```
function ut=f(t,u)
n=length(u);
dx2=((xu-xl)/(n-1))^2;
ut=zeros(1,n);
for i=1:n
    if(i==1) ut(i)=0.0;
    elseif(i==n) ut(i)=2.0*(u(i-1)-u(i))/dx2;
    else ut(i)=(u(i+1)-2.0*u(i)+u(i-1))/dx2;
    end
end
```

unde în ciclul for in prima linia impunem condiţia Dirichlet (20), apoi pe linia următoare va fi impusă condiţia Neumann (21) care vine din aproximarea cu diferenţe finite

$$\frac{\partial u}{\partial x} \approx \frac{u(i+1) - u(i-1)}{\Delta x} = 0,$$

iar pentru calculul punctelor interioare folosim aproximarea cu diferenţe finite centrate de ordin doi

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{u(i+1) - 2u(i) + u(i-1)}{\Delta x^2}.$$

Am construit astfel sistemul de ecuaţii diferenţiale

$$u'(t) = f(t,u), \quad f : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^n \tag{24}$$

$$u(0) = u0 \in \mathbb{R}^n \tag{25}$$

Folosim acuma solverul din Matlab ode15s unde cerem ca toleranţa absolută şi toleranţa relativă să fie 10^{-5}. Eroarea în punctul $x = 0.5$ se poate vedea în figura de mai jos

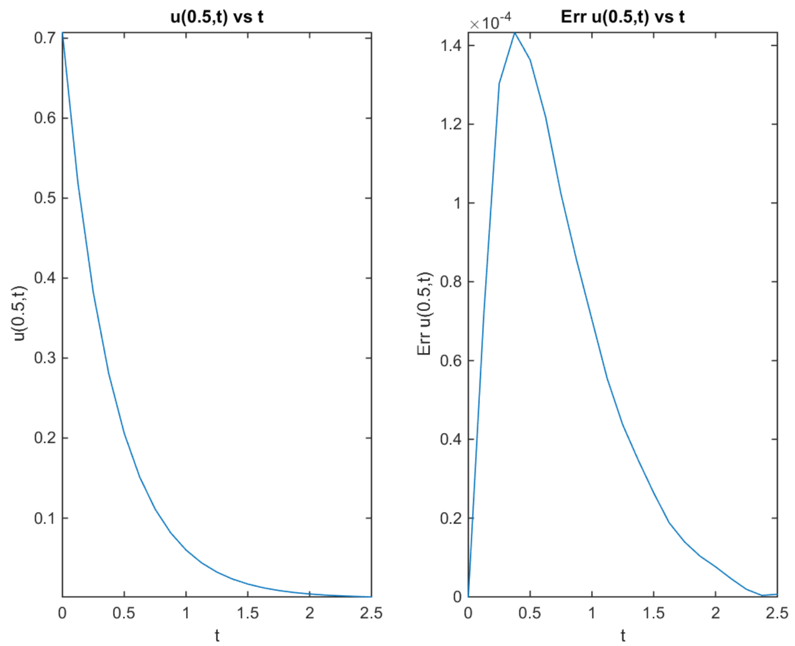

FIG. 5.1 – Eroarea de aproximare pentru $u(0.5,t)$ cu ode15s

Se observă că toleranţa cerută nu este satisfăcut. Asta se datorează faptului că am ales o discretizare în timp prea mică pentru solverul ode15s.

Ca să atingem toleranţa cerută va trebui să dublăm numărul nodurilor de timp, dar astfel creştem dimensiunea sistemului cea ce din punct de vedere computaţional ne îngreunează rezolvarea sistemului.

Dacă folosim solverul **CVODE** din librăria Sundials şi ne cerem toleranţa absolută şi relativă să fie tot 10^{-5} atunci solverul furnizează soluţia:

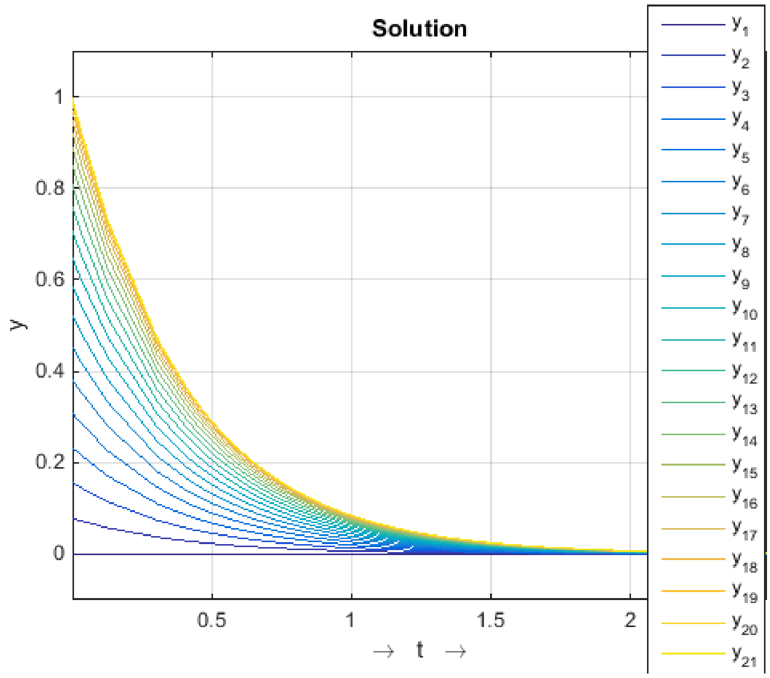

FIG. 5.2 – Soluţia sistemului de ecuaţii (24)

Iar dacă comparăm aproximările obţinute cu valorile exacte în punctul $x = 0.5$ obţinem graficul 5.3 cu erorile mai mici decât toleranţa cerută. Astfel observăm că solverul **CVODE** este mai performant decât solverul ode15s din Matlab. Dacă încercăm **CVODE** cu toleranţele 10^{-8} şi tot cu $n = 21$ de puncte în t atunci se observă pe figura 5.4 că solverul stisface toleranţele cerute fără problemă.

În figura 5.5 am reprezentat în spaţiu soluţia obţinută cu **CVODE** pentru $n = 21$ de puncte în timp şi cu toleranţele 10^{-8}.

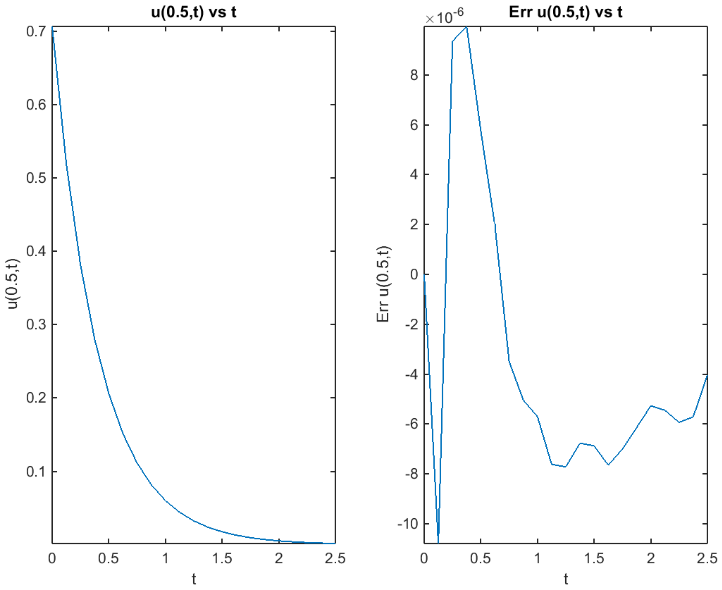

FIG. 5.3 – Eroarea de aproximare pentru $u(0.5, t)$ cu **CVODE**

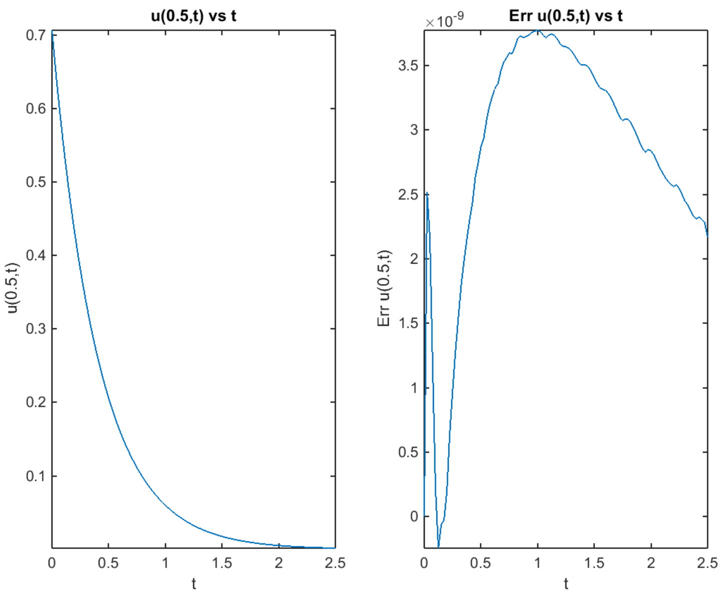

FIG. 5.4 – Eroarea de aproximare pentru $u(0.5, t)$ cu **CVODE**

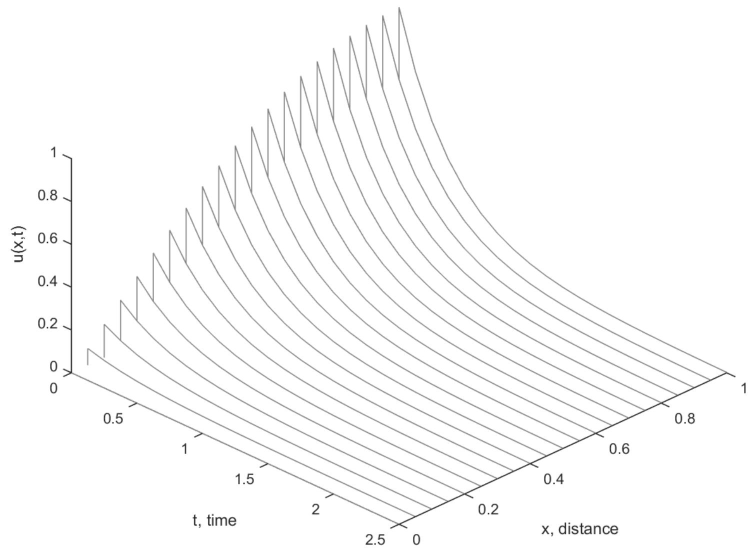

FIG. 5.5 – Aproximarea lui u cu metoda liniilor

BIBLIOGRAFIE

[1] Graham W. Griffiths, William E. Schiesser. *A compendium of partial differential equation models: method of lines analysis with Matlab*. Cambridge University Press, 2009.

[2] Graham W. Griffiths, William E. Schiesser. *Traveling wave analysis of partial differential equations: numerical and analytical methods with MATLAB and Maple*. Academic Press, 2010.

[3] Saucez, Ph., William E. Schiesser, A. Vande Wouwer. *Adaptive method of lines.* CRC Press, 2001.

[4] Strikwerda, John C. *Finite difference schemes and partial differential equations*. Siam, 2004.

[5] SUNDIALS (SUite of Nonlinear and DIfferential/ALgebraic equation Solvers), https://computation.llnl.gov/casc/sundials/main.html

[6] Wouwer, A. Vande, et al. *A MATLAB implementation of upwind finite differences and adaptive grids in the method of lines*. Journal of computational and applied mathematics 183.2 (2005): 245-258.

e-mail: boros.math@gmail.com

Received: 22 Mai 2015