 CMES, vol.74, no.2, pp.139-159, 2011

A Temporally-Piecewise Adaptive Algorithm to Solve Transient Convection-Diffusion Heat Transfer Problems

Xiao Zhao[1], Haitian Yang[1,2] and Qiang Gao[1]

Abstract: A piecewised adaptive algorithm in the time domain is presented to solve the transient convection-diffusion heat transfer problem. By expanding all variables at a time interval, an initial and boundary value problem is decoupled into a series of recursive boundary value problems which can be solved by FEM or other well developed numerical schemes to deal with boundary value problems. A steady computing accuracy can be adaptively maintained via the power increase of the expansion, particularly when the step size varies in the whole computing process. Additionally for the nonlinear cases, there is no requirement of iteration and additional assumption for the proposed algorithm. Three numerical examples are provided to verify the presented algorithm,and satisfactory results have been achieved.

Keywords: convection-diffusion; heat transfer; adaptive algorithm; time domain; FEM.

1 Introduction

There have been a number of well developed numerical methods to solve various steady/transient convection-diffusion problems. These methods are generally classified into two pools. One consists of those to deal with the boundary value problem, such as the characteristic finite volume method [Rui, H. X. (2008)], and time-space finite element method, etc.. In this pool, the Galerkin FE method and its alternatives constitute an important family that includes SUPGM (Streamline Upwind Petrov-Galerkin method) [Brooks, A. N.; Hughes, T. J. R. (1982), Franca, L. P.; Valentin, F. (2000)], GLSM (Galerkin/Least-Squares Method) [Hughes T. J. R.; Franca, L. P.; Hulbert, G. M. (1989)], USFEM (Unusual Stabilized Finite Element Method) [Franca, L. P.; Valentin, F. (2000)],the upstream-weighted higher

[1] State Key Lab of Structural analysis for industrial Equipment, Dept. of Engineering Mechanics, Dalian University of Technology, Dalian, 116024, Dalian, P.R. China

[2] Corresponding author. E-mail address: Haitian@dlut.edu.cn, Tel.: +86-41184708394; fax: +86-41184708393

order schemes [Leonard, B. P. (1979), Cox, R. A.; Nishikawa, T. (1991)], the characteristic-based Galerkin methods [Peraire, J.; Zienkiewicz, O. C.; Morgan, K. (1986), Li, X. K.; Wu, W. H.; Zienkiewicz, O. C. (2000)], MFEM (Multiscale Finite Element Method) [Hou, T. Y.; Wu, X. H. (1997)], and the discontinuous Galerkin method [Gopalakrishnan, J.; Kanschat, G. (2003)], etc..

Another pool is mainly relevant to the numerical technique to deal with the initial value problem, such as the consistent splitting scheme [Pontaza, J. P. (2007)], the space–time conservation element and solution element scheme (CE-SE) [Ponsoda, E.; Defez, E.; Roselló, M. D.; Romero, J. V. (2008)], the conservative dissipative scheme [Corre, C.; Lerat, A. (2008)], various time integration methods [De Palma, P.; Pascazio, G.; Rubino, D. T.; Napolitano, M. (2006)], and the predictor-multicorrector approach that requires iterations in the computing process [Brooks, A. N.; Hughes, T. J. R. (1982)], etc.. A big portion of this pool is occupied by the finite difference family [Rubio, A. D.; Zalts, A.; El Hasi, C. D. (2008), Rodríguez, N. J.; Davey, K.; Vázquez Feijoo, J. A.; Juárez-Hernandez, A. (2009), Witek, M. L.; Teixeira, J.; Flatau, P. J. (2008), Formaggia, L.; Nobile, F. (2004), Wong, K. T. M.; Lee, J. H. W.; Choi, K. W. (2008)] that includes the Euler method [Li, X. K.; Wu, W. H.; Zienkiewicz, O. C. (2000), Rui, H. X. (2008), Formaggia, L.; Nobile, F. (2004), Caliari, M.; Vianello, M.; Bergamaschi, L. (2007)], the Crank-Nicholson scheme [Donea, J. (1984), Wong, K. T. M.; Lee, J. H. W.; Choi, K. W. (2008)], θ-method [Labeur, R. J.; Wells, G. N. (2007), Karahan, H. (2006)] that will directly turn to the Forward Euler, the Backward Euler, or the Crank–Nicolson methods with special value of θ ($\theta = 1, 0, \frac{1}{2}$ respectively) [Khaliq, A. Q. M.; Voss, D. A.; Kazmi, K. (2008)], and the Runge-Kutta methods [Rui, H. X. (2008), Klaij, C. M.; van Raalte. M. H.; van der Ven, H.; van der Vegt, J. J. W. (2007), Klaij, C. M.; van der Vegt, J. J. W.; van der Ven, H. (2006)], etc.. We notice that among these algorithms, the local truncation order of the 4/5-stage Runge-Kutta methods are $O\left(\Delta t^4\right)$[Burden, R. L.; Faires, J. D. (1988)], others are lower than$O\left(\Delta t^4\right)$. The fact is that a relatively lower local truncation order at a time interval may possibly lead to inaccurate or even unacceptable computing results, as shown in the numerical examples. The major interest of this paper focuses on improving the local truncation order at a time interval, so as to keep a steady computing accuracy in the whole computing process.

Enlightened by the $p-$refinement skill of FEM [Zienkiewicz, O. C.; Taylor, R. L. (2000)], a temporally piecewised adaptive algorithm is developed to solve transient convection-diffusion heat transfer problems in this paper, by which an initial and boundary value problem is decoupled into a series of recurrent boundary value problems which can be solved by FEM or other well-developed numerical techniques to deal with boundary value problems, and a steady computing accuracy

can adaptively be maintained via the power increase of the expansion. Additionally for the nonlinear cases there is no requirement of additional assumption and iteration for the proposed algorithm.

3 numerical examples are presented to verify the proposed approach, and good accordance can be observed in comparison with the results given by the analytical, Runge-Kutta and Crank-Nicolson methods.

2 Recursive governing equations of transient convection-diffusion heat transfer problems

The governing equation of transient convection-diffusion heat transfer problems can be written as [Platten, J. K.; Legros, J. C. (1984)]

$$c\left[T_{,t}+u_iT_{,i}\right]=\left[k_iT_{,i}\right]_{,i}+Q \quad x_i\subset\Omega \tag{1}$$

where T denotes temperature, t refers to time, Q is a source or reaction term , c and k_i are heat capacity and thermal conductivity, respectively, u_i represents the vector of fluid velocity, x_i is the vector of the coordinates, and Ω represents the space domain of the problem, and subscript i refers to a summation index (i=1,2 for the 2D problem, i=1,2,3 for the 3D problem).

The initial condition is describe by

$$T|_{t=0}=T_0 \tag{2}$$

where T_0 is a prescribed function.

The boundary condition is given by

$$T=T_B \quad x_i\in\Gamma_1 \tag{3}$$

$$n_ik_iT_{,i}=q \quad x_i\in\Gamma_2 \tag{4}$$

where T_B and q are prescribed functions, $\Gamma=\Gamma_1+\Gamma_2$ represents the whole boundary of Ω, and n_i refers to the outward unit normal along Γ_2.

We divide time domain into a number of intervals, initial points and sizes of intervals are defined by $t_0,t_1,t_2.....,t_\lambda,.....$ and $t_{s_1},t_{s_2}.....,t_{s_\lambda},.....$, respectively.

At $\lambda-th(\lambda\geq 1)$ interval, all variables are expanded in the form

$$T=\sum_{m=0}T^ms^m \tag{5}$$

$$c=\sum_{m=0}c^ms^m \tag{6}$$

$$k_i = \sum_{m=0} k_i^m s^m \tag{7}$$

$$Q = \sum_{m=0} Q^m s^m \tag{8}$$

$$u_i = \sum_{m=0} u_i^m s^m \tag{9}$$

$$T_B = \sum_{m=0} T_B^m s^m \tag{10}$$

$$x_i \in \Gamma_1 \tag{11}$$

$$q = \sum_{m=0} q^m s^m \tag{12}$$

$$x_i \in \Gamma_2 \tag{13}$$

$$s = \frac{t - t_{\lambda-1}}{t_{s_\lambda}} \tag{14}$$

whereT^m denotes the expansion coefficient of T, c^m, k_i^m, Q^m, u_i^m, q^m and T_B^m represent expansion coefficients of c, k_i, Q, u_i, q and T_B, respectively.

Utilizing

$$\frac{\partial}{\partial t} = \frac{\partial}{\partial s}\frac{\partial s}{\partial t} = \frac{1}{t_{s_\lambda}}\frac{\partial}{\partial s} \tag{15}$$

we have

$$T_{,t} = \sum_{m=0} \frac{(m+1)}{t_{s_\lambda}} T^{m+1} s^m \tag{16}$$

Substituting Eqns. (5)–(11) and (14) for Eqns. (1), (3) and (4) and equating the powers of s at two sides of the equations then yields

$$\frac{(n+1)}{t_{s_\lambda}} c^0 T^{n+1} + \sum_{m=0}^{n} \frac{m}{t_{s_\lambda}} c^{n-m+1} T^m + \sum_{m=0}^{n} \sum_{p=0}^{n-m} c^{n-m-p} u_i^p \left(T^m\right)_{,i} = \sum_{m=0}^{n} \left[k_i^{n-m} \left(T^m\right)_{,i}\right]_{,i} + Q^n \tag{17}$$

$$T^n = T_B^n \text{ on } \Gamma_1 \tag{18}$$

$$\sum_{m=0}^{n} n_i k_i^{n-m} T_{,i}^m = q^n \text{ on } \Gamma_2 \tag{19}$$

3 Implementation of FEM

In the above section, an initial and boundary value problem is decoupled into a series of recursive boundary value problems which can be solved using FEM or other well-developed numerical techniques to deal with boundary value problems.

A utilization of a weighted residual method on Eqns. (15) $\sim$ (17) leads to [Zienkiewicz, O. C.; Taylor, R. L. (2000)]

$$\begin{aligned}
&\int_\Omega W\frac{(n+1)}{t_{s_\lambda}}c^0T^{n+1}d\Omega+\sum_{m=0}^{n}\int_\Omega W\frac{m}{t_{s_\lambda}}c^{n-m+1}T^m d\Omega\\
&+\sum_{m=0}^{n}\sum_{p=0}^{n-m}\int_\Omega Wc^{n-m-p}u_i^p\left(T^m\right)_{,i}d\Omega\\
&-\sum_{m=0}^{n}\int_\Omega W\left[k_i^{n-m}\left(T^m\right)_{,i}\right]_{,i}d\Omega=\int_\Omega WQ^n d\Omega-\int_{\Gamma_2}W\left(\sum_{m=0}^{n}n_ik_i^{n-m}T_{,i}^m-q^n\right)d\Gamma_2
\end{aligned}\tag{20}$$

where W denotes a weighting function.

The application of integration by parts gives

$$\begin{aligned}
&\int_\Omega W\frac{(n+1)}{t_{s_\lambda}}c^0T^{n+1}d\Omega+\sum_{m=0}^{n}\int_\Omega W\frac{m}{t_{s_\lambda}}c^{n-m+1}T^m d\Omega\\
&+\sum_{m=0}^{n}\sum_{p=0}^{n-m}\int_\Omega Wc^{n-m-p}u_i^p\left(T^m\right)_{,i}d\Omega\\
&-\sum_{m=0}^{n}\int_\Omega W_{,i}\left[k_i^{n-m}\left(T^m\right)_{,i}\right]d\Omega=\int_\Omega WQ^n d\Omega+\int_{\Gamma_2}Wq^n d\Gamma_2
\end{aligned}\tag{21}$$

The choice of the weighting function W leads to various approximation methods, in the Galerkin FE method, T^j and W are approximated by

$$T^j=[N]\{T\}_e^j \tag{22}$$

$$W=[N]\{W\}_e \tag{23}$$

where $[N]$ represents a matrix of shape functions, $\{T\}_e^j$ and $\{W\}_e$ represent the local nodal vectors of T^j and W, respectively.

Substituting Eqns. (20) $\sim$ (21) into Eqn. (19) and considering the arbitrariness of

$\{W\}_e$ then yields

$$\frac{(n+1)}{t_{s_\lambda}} C^0 \{T\}^{n+1} + \sum_{m=0}^{n} \frac{m}{t_{s_\lambda}} C^{n-m+1} \{T\}^m + \sum_{m=0}^{n} \sum_{p=0}^{n-m} M^{n-m} \{T\}^m + \sum_{m=0}^{n} K^{n-m} \{T\}^m = \{Q\}^n + \{q\}^n \quad (24)$$

where $\{T\}^m$ represents the *m-th* coefficient vector of $\{T\} = \sum_{m=0} \{T\}^m s^m$. $\{T\}$ is the global nodal vector of T,

$$C^0 = \sum_e \int_{\Omega_e} [N]^T c^0 [N] d\Omega_e \quad (25)$$

$$C^{n-m+1} = \sum_e \int_{\Omega_e} [N]^T c^{n-m+1} [N] d\Omega_e \quad (26)$$

$$M^{n-m} = \sum_e \int_{\Omega_e} c^{n-m-p} [N]^T u_i^p [N_{,i}] d\Omega_e \quad (27)$$

$$K^{n-m} = \sum_e \int_{\Omega_e} [N_{,i}]^T k_i^{n-m} [N_{,i}] d\Omega_e \quad (28)$$

$$\{Q\}^n = \sum_e \int_{\Omega_e} [N]^T Q^n d\Omega_e \quad (29)$$

$$\{q\}^n = \sum_e \int_{\Gamma_{2e}} [N]^T q^n d\Gamma_{2e} \quad (30)$$

where $[N_{,i}]$ denotes a matrix of derivatives of the shape functions.

We also derived a recursive equation similar to Eqn. (22) by combining Eqns. (15) $\sim$ (17) with the multilevel discontinuous Galerkin methods [Gopalakrishnan, J.; Kanschat, G. (2003)]. Owing to the limited capacity for an article, this recursive equation and its derivation are not presented here.

4 Adaptive process

At the first time interval, $\{T\}^0$ is prescribed by the initial condition, at the λ*-th* time interval ($\lambda > 1$), $\{T\}^0$ is provided by

$\{T\}^0 = \{T\}^m|_{m=0}$(at λ-th time interval)

$$= \sum_{m=0} \{T\}^m s^m \Big|_{s=1} = \sum_{m=0} \{T\}^m \text{ ((at } (\lambda \text{ - } 1)\text{-th time interval)} \quad (31)$$

when $\{T\}^0$ is given, $\{T\}^m$ $(m \geq 1)$ can be calculated using Eqn. (22), and $\{T\}$ can be attained by

$$\{T\} = \sum_{m=0} \{T\}^m s^m \tag{32}$$

An adaptive computing process at a time interval can be realized with the increase of m until the following criteria is tenable.

$$\frac{\| \{T\}^m s^m|_{s=1} \|_2}{\left\| \sum_{j=0}^{m-1} \{T\}^j s^j \Big|_{s=1} \right\|_2} \leq \beta \tag{33}$$

where β is a prescribed error tolerance, and $\|\bullet\|_2$ represents a L^2-norm.

If Eqn. (31) holds on continuously for three times at $m = Mmax$, the computation at the current time interval will stop and step into the next one. At different time intervals, $Mmax$ usually is different. Regarding to Eqn. (30), the local truncation order of the proposed algorithm is $Mmax + 1$ at a time interval.

The above process is similar to the *p-refinement* process in which the adaptive computing is fulfilled by increasing terms of polynomials [Harper, C. (1976)].

5 Numerical verification

This section provides 3 numerical examples with time dependent boundary conditions. The thermal conductivity in the Example 1 and the source term in the Example 3 are temperature dependent. The eight node quadrilateral element is employed in FE analysis.

For the simplicity, all the computing parameters are assumed dimensionless.

A L^2-norm error [Sudirham, J. J.; van der Vegt, J. J. W.; van Damme, R. M. J. (2006)] and a summation of relative errors are employed to evaluate the computing accuracy, i.e.

$$err(t) = \|\{T_n(t)\} - \{T_a(t)\}\|_2 \tag{34}$$

$$REerr(t) = \sqrt{\sum_{r=1} \left(\frac{T_n(r,t) - T_a(r,t)}{T_a(r,t)} \right)^2} \tag{35}$$

where $\{T_n(t)\}$ and $\{T_a(t)\}$ stand for the vectors of numerical and analytical solutions at all FE nodes, and r refers to *r-th* component of $\{T_n(t)\}$ and $\{T_a(t)\}$.

All the computing tasks are completed on a PC (Pentium D, CPU: 2.80 GHz, Memory: 1 Gbyte), and all the computing programs are coded in Matlab.

Example 1 considers a nonlinear problem in a 1×0.1 rectangular domain where k_1 is temperature dependent [Shih, T. M. (1983)]. The Galerkin FEM is employed to deal with boundary value problems, and a 10×1 uniform FE grid is used, as schematically illustrated in Fig. 1.

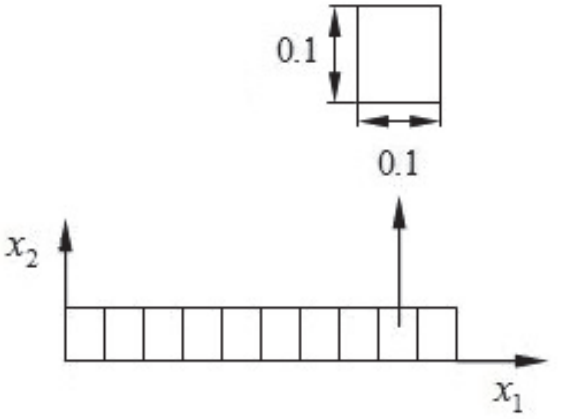

Figure 1: The FE mesh of Example 1

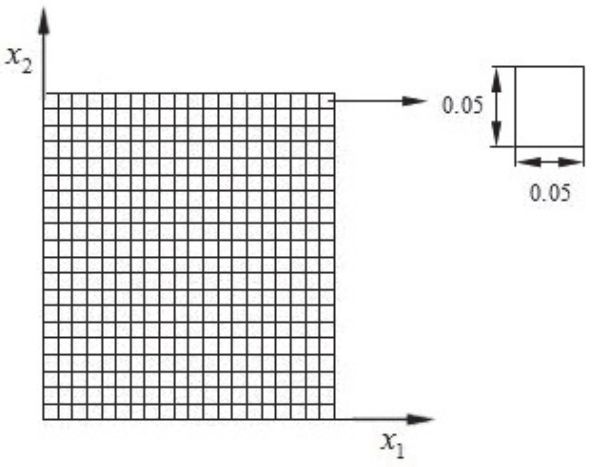

Figure 2: The FE mesh of Example 2

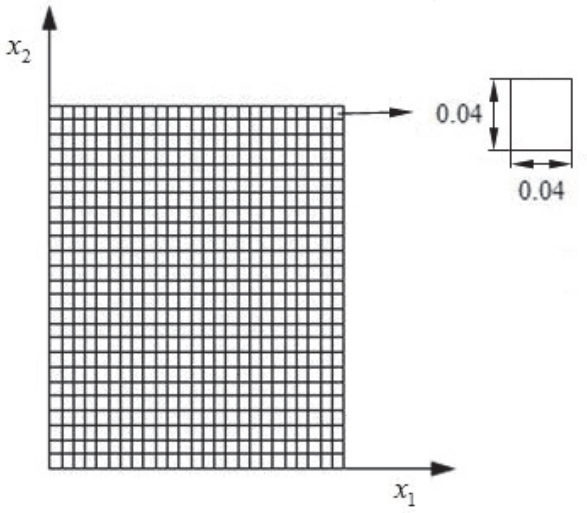

Figure 3: The FE mesh of Example 3

The boundary condition is specified by

$$T_B(0,x_2,t) = \frac{2^2}{24-6t} \quad T_B(1,x_2,t) = \frac{3^2}{24-6t}.$$

The initial condition is

$$T_0 = \frac{(x_1+2)^2}{24}.$$

Other computing parameters are given by

$$Q = 0, \quad c = 1.0, \quad u_1 = 0.0, \quad u_2 = 0.0$$

$$k_1 = T, \quad k_2 = 0, \quad t \in [0,3.8], \quad \beta = 1e-8.$$

The exact solution of this problem is given by [Shih, T. M. (1983)]

$$T = \frac{(x_1+2)^2}{24-6t}.$$

Regarding to Eqn. (10), T_B is expended in the form

$$T_B(0,x_2,t) = \frac{2^2}{24-6t} = \frac{2}{3}\cdot\frac{1}{4-\left(t_{\lambda-1}+s\cdot t_{s_\lambda}\right)}$$
$$= \frac{2}{3\cdot(4-t_{\lambda-1})}\cdot\frac{1}{\left(1-\frac{t_{s_\lambda}\cdot s}{4-t_{\lambda-1}}\right)} = \frac{2}{3\cdot(4-t_{\lambda-1})}\cdot\left[\sum_{n=0}\left(\frac{t_{s_\lambda}\cdot s}{4-t_{\lambda-1}}\right)^n\right]\ (\lambda\geq 1)$$

and

$$T_B(1,x_2,t) = \frac{(2+1)^2}{24-6t} = \frac{3}{2}\cdot\frac{1}{4-\left(t_{\lambda-1}+s\cdot t_{s_\lambda}\right)}$$
$$= \frac{3}{2\cdot(4-t_{\lambda-1})}\cdot\frac{1}{\left(1-\frac{t_{s_\lambda}\cdot s}{4-t_{\lambda-1}}\right)} = \frac{3}{2\cdot(4-t_{\lambda-1})}\cdot\left[\sum_{n=0}\left(\frac{t_{s_\lambda}\cdot s}{4-t_{\lambda-1}}\right)^n\right]\ (\lambda\geq 1)$$

The solution obtained by the proposed algorithm is exhibited in Tab. 1 and Figs. 4-8, and compared with the solutions given by Heun's method [Burden, R. L.; Faires, J. D. (1988)], the Modified Euler method [Burden, R. L.; Faires, J. D. (1988)] and the 4-order Runge-Kutta method [Burden, R. L.; Faires, J. D. (1988)]. The comparisons indicate that the presented algorithm is a bit computationally expensive than three others, but it is able to keep a steady computing accuracy with the increase of step size, as shown in Fig. 7. When $t_{s_\lambda} = 2\times 10^{-3}(\lambda = 1,2,3,...,500)$, all the other three methods are unable to obtain reasonable solutions,as shown in Fig. 8. Figs. 9 and 10 mean that a relatively smaller β is required to maintain a steady computing accuracy when step size becomes relatively larger.

Example 2 considers a convection-diffusion problem in a domain $\Omega = (0,1)^2$ [Sudirham J. J.; van der Vegt, J. J. W.; van Damme, R. M. J. (2006)]. The Galerkin FEM is employed to deal with boundary value problems, and a 20×20 uniform FE grid is used, as schematically illustrated in Fig. 2.

The initial condition is

$$T_0 = \sin(\pi x_1)\cdot\sin(\pi x_2),$$

The boundary condition is specified by

$$T_B(0,x_2,t) = \sin(-\pi u_1 t)\cdot\sin(\pi(x_2-u_2 t))\cdot\exp\left(-2D\pi^2 t\right),$$

$$T_B(1,x_2,t) = \sin(\pi(1-u_1 t))\cdot\sin(\pi(x_2-u_2 t))\cdot\exp\left(-2D\pi^2 t\right),$$

$$T_B(x_1,0,t) = \sin(\pi(x_1-u_1 t))\cdot\sin(-\pi u_2 t)\cdot\exp\left(-2D\pi^2 t\right),$$

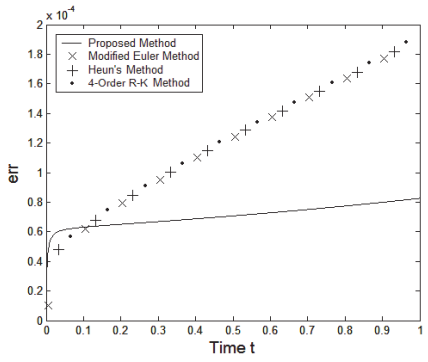

Figure 4: A companion of $err(t)$ with $t_{s_\lambda} = 5\times10^{-4}\,(\lambda = 1,2,3,...,2000)$

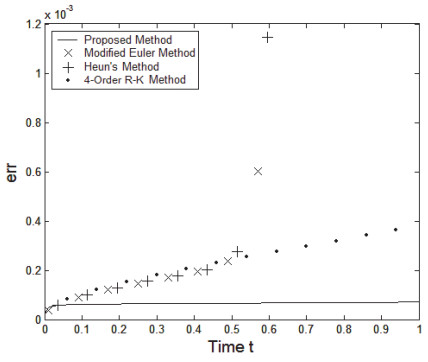

Figure 5: A companion of $err(t)$ with $t_{s_\lambda} = 1\times10^{-3}\ (\lambda = 1,2,3,...,1000)$

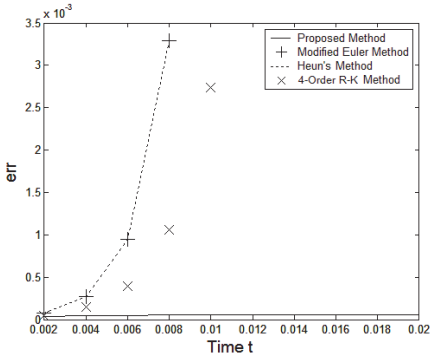

Figure 6: A companions of $err(t)$ with $t_{s_\lambda} = 2\times10^{-3}\ (\lambda = 1,2,3,...,10)$

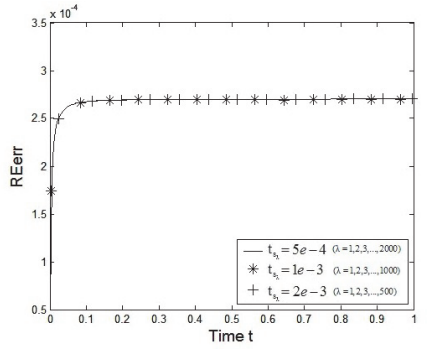

Figure 7: A companion of $REerr(t)$ of the proposed method with different t_{s_λ}

$$T_B(x_1,1,t) = \sin(\pi(x_1-u_1t))\cdot\sin(\pi(1-u_2t))\cdot\exp\left(-2D\pi^2t\right),$$

The computing parameters are given by

$$Q=0,\quad c=1.0,\quad u_1=1,\quad u_2=1,\quad k_1=1,\quad k_2=1,\quad \beta=1\times10^{-8}.$$

The analytical solution is given by [Sudirham, J. J.; van der Vegt, J. J. W.; van Damme, R. M. J. (2006)].

$$T(x_1,x_2,t) = \sin(\pi(x_1-u_1t))\cdot\sin(\pi(x_2-u_2t))\cdot\exp\left(-2D\pi^2t\right).$$

Since T_B seems difficult to expand directly in the term of s it is approximated using a polynomial $P_{ij}\left(x_{1_i},x_{2_j},t,\kappa\right)$ at each FE node $\left(x_{1_i},x_{2_j}\right)$ along the boundary.

$$P_{ij}\left(x_{1_i},x_{2_j},t,\kappa\right) = a_0\left(x_{1_i},x_{2_j}\right)+a_1\left(x_{1_i},x_{2_j}\right)t+a_2\left(x_{1_i},x_{2_j}\right)t^2+\cdots+a_\kappa\left(x_{1_i},x_{2_j}\right)t^\kappa$$

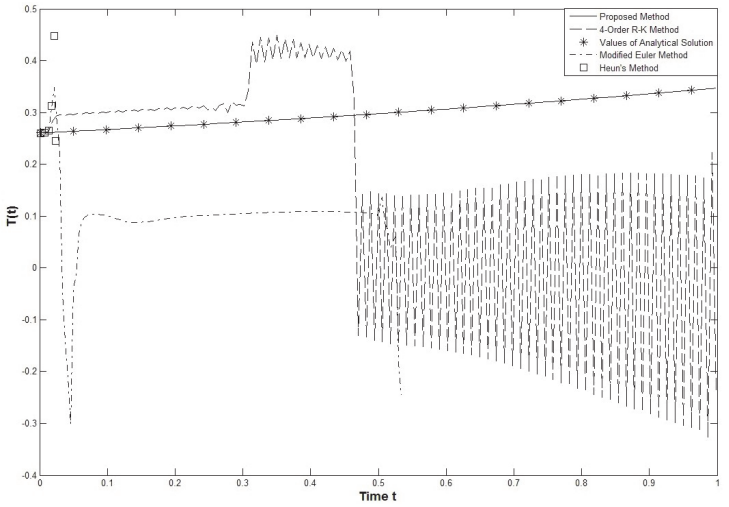

Figure 8: A companion of $T(x_1,x_2,t)|_{x_1=0.5,x_2=0}$ with $t_{s_\lambda} = 2 \times 10^{-3}$ $(\lambda = 1,2,3,...,500)$

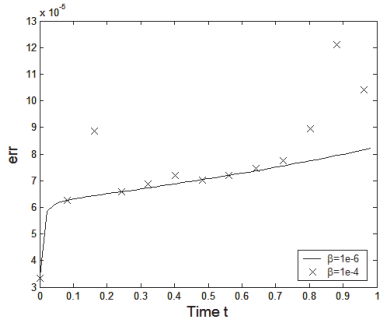

Figure 9: The effects of β on $err(t)$ with $t_{s_\lambda} = 2 \times 10^{-3}$ $(\lambda = 1,2,3,...,500)$

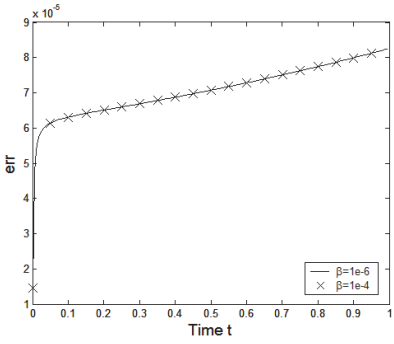

Figure 10: The effects of β on $err(t)$ with $t_{s_\lambda} = 5 \times 10^{-4}$ $(\lambda = 1,2,3,...,2000)$

$$a_0\left(x_{1_i},x_{2_j}\right), \quad a_1\left(x_{1_i},x_{2_j}\right), \quad a_2\left(x_{1_i},x_{2_j}\right), \quad \ldots, \quad a_\kappa\left(x_{1_i},x_{2_j}\right)$$

are determined by fitting $P_{ij}\left(x_{1_i},x_{2_j},t,\kappa\right)$ with $T_B\left(x_{1_i},x_{2_j},t\right)$ at N_f points uniformly distributed within the whole time domain. T_B^m can be obtained easily via $P_{ij}\left(x_{1_i},x_{2_j},t,\kappa\right)$. The fitting criteria is

Table 1: Numerical Comparison of Example 1 (at $t = 1$)

	Proposed Method	Modified Euler Method	Heun's Method	4-order R-K Method
t_{s_λ}		$5\times10^{-4}\ (\lambda=1,2,3,...,2000)$		
err	0.000082574	0.00019082	0.000190825	0.00019288
CPU time (s)	898.22	176.5	182.58	367.06
t_{s_λ}		$1\times10^{-3}\ (\lambda=1,2,3,...,1000)$		
err	0.000082574	0.69251	2.052	0.00037976
CPU time (s)	457.36	88.61	88.375	177.25
t_{s_λ}		$2\times10^{-3}\ (\lambda=1,2,3,...,500)$		
err	0.00008261978	\	\	\
CPU time (s)	644.31	\	\	\

$$\left(\sum_{l=1}^{N_f}\left(T_B\left(x_{1_i},x_{2_j},t_l\right)-P_{ij}\left(x_{1_i},x_{2_j},t_l,\kappa\right)\right)^2\right)^{\frac{1}{2}}<\eta,\ (\eta\le\beta)$$

In this example $N_f = 2000, \kappa = 5$, $\eta = 1\times10^{-8}$.

A numerical comparison is exhibited in Tab. 2 and Figs. 11-13 where both the Modified Euler method and the 4-order Runge-Kutta method failed when $t_{s_\lambda} = 10^{-4}$ $(\lambda = 1,2,3,...,2000)$ and $t_{s_\lambda} = 10^{-3}$ $(\lambda = 1,2,3,...,200)$.

Example 3 considers an autonomous advective Fisher Equation in which $Q = \gamma\cdot T^2(x_1,x_2,t)\cdot(1-T(x_1,x_2,t))$ is temperature dependent in the domain $\Omega = (0,1)^2$[Caliari, M.; Vianello, M.; Bergamaschi, L. (2007)]. The multilevel discontinuous Galerkin FEM is employed to deal with boundary value problems, and a 25×25 uniform FE grid is used, as schematically illustrated in Fig. 3.

The other computing parameters are

$$\gamma = 0.01,\quad u_1 = u_2 = \alpha = -1,$$

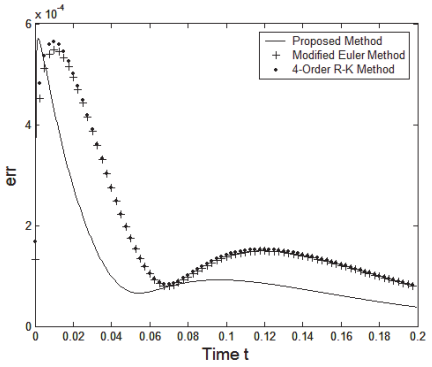

Figure 11: A companion of $err(t)$ with $t_{s_\lambda} = 5\times10^{-5}\ \ (\lambda = 1,2,3,...,4000)$

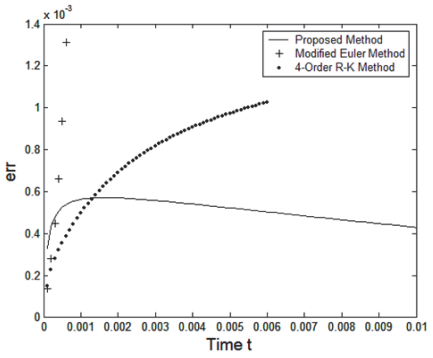

Figure 12: A companion of $err(t)$ with $t_{s_\lambda} = 1\times10^{-4}\ \ (\lambda = 1,2,3,...,2000)$

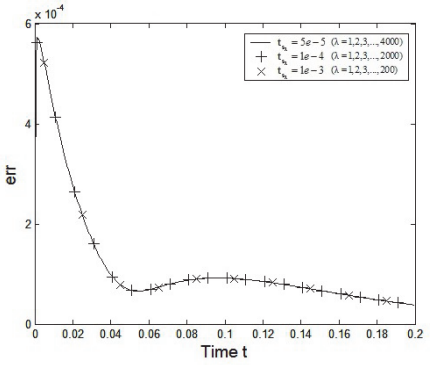

Figure 13: A companion of $err(t)$ of the proposed method with different t_{s_λ}

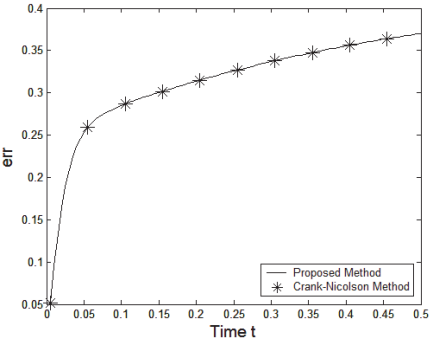

Figure 14: A companion of $err(t)$ with $t_{s_\lambda} = 5\times10^{-3}\ \ (\lambda = 1,2,3,...,100)$

$$k_1 = k_2 = \varepsilon = 0.001, \quad \beta = 1\times10^{-8}, \quad c = 1.0.$$

The initial and boundary conditions are prescribed by

$$T_0 = \frac{1}{1+\exp\left(a\cdot(x_1+x_2)+p\right)},$$

$$T_B(0,x_2,t) = \frac{1}{1+\exp\left(a\cdot(x_2-b\cdot t)+p\right)},$$

$$T_B(1,x_2,t) = \frac{1}{1+\exp\left(a\cdot(1+x_2-b\cdot t)+p\right)},$$

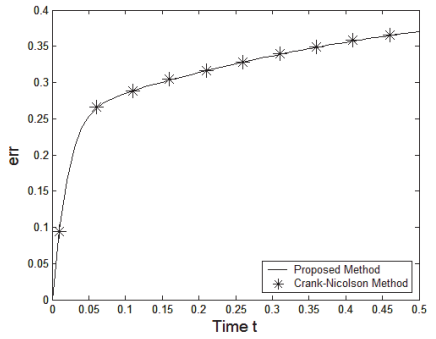

Figure 15: A companion of $err(t)$ with $t_{s_\lambda} = 1 \times 10^{-2} \quad (\lambda = 1,2,3,...,50)$

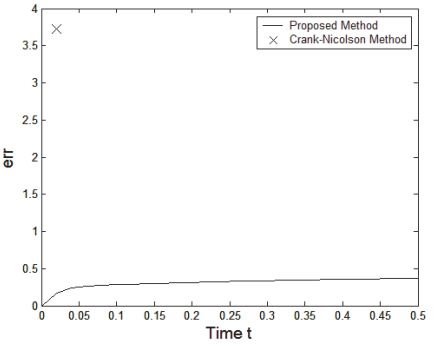

Figure 16: A companion of $err(t)$ with $t_{s_\lambda} = 2 \times 10^{-2} \quad (\lambda = 1,2,3,...,25)$

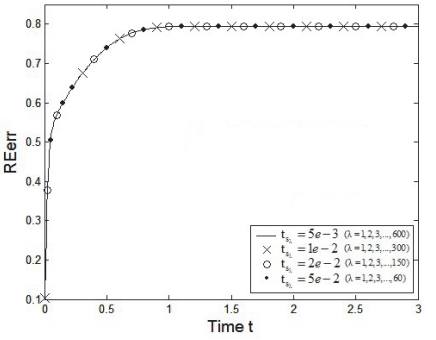

Figure 17: A companion of $REerr(t)$ of the proposed method with different t_{s_λ}

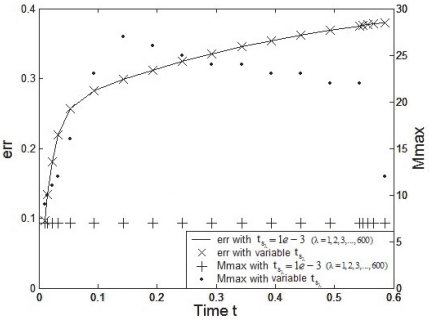

Figure 18: The companions of $Mmax$ and $err(t)$ with constant and variable t_{s_λ}

$$T_B(x_1,0,t) = \frac{1}{1+\exp(a\cdot(x_1 - b\cdot t)+p)},$$

$$T_B(x_1,1,t) = \frac{1}{1+\exp(a\cdot(x_1 + 1 - b\cdot t)+p)},$$

The exact solution is given by [Caliari, M.; Vianello, M.; Bergamaschi, L. (2007)]

$$T(x_1,x_2,t) = \frac{1}{1+\exp(a\cdot(x_1 + x_2 - b\cdot t)+p)},$$

where $a = \sqrt{\frac{\gamma}{4\varepsilon}}$, $b = 2\alpha + \sqrt{\gamma\varepsilon}$ and $p = a(b-1)$.

Table 2: Numerical Comparison of Example 2 (at $t = 0.2$)

	Proposed Method	Modified Euler Method	4-order R-K Method
t_{s_λ}		$5\times10^{-5}\ (\lambda = 1,2,3,...,4000)$	
err	0.000037905	0.0000778	0.0000778
CPU time (s)	119226	56887	58682
t_{s_λ}		$1\times10^{-4}\ (\lambda = 1,2,3,...,2000)$	
err	0.000037905	\	\
CPU time (s)	96301	\	\
t_{s_λ}		$1\times10^{-3}\ (\lambda = 1,2,3,...,200)$	
err	0.000037905	\	\
CPU time (s)	99505	\	\

T_B^m is determined in the same way as Example 2 with $\eta = 1\times10^{-8}$, $\kappa = 4$, N_f=5000. A numerical comparison with the Crank-Nicolson method [Caliari, M.; Vianello, M.; Bergamaschi, L. (2007)] is given in Tab. 3 and Figs. 14-16. $REerr(t)$curves in Fig. 17 exhibit a steady computing accuracy maintained by the proposed method with different size of t_{s_λ}. Numerical results with constant ($t_{s_\lambda} = 1\times10^{-3}$,($\lambda = 1,2,3,...,500$)) and variable step sizes (as shown in Tab. 4) in the whole computing process are compared in Fig. 18. As we expected, the adaptive process effectively guarantees a steady computing accuracy when the step size varies.

However the Crank-Nicolson method that is a sound baseline benchmark for an advection-diffusion solver [Caliari, M.; Vianello, M.; Bergamaschi, L. (2007)] failed when $t_{s_\lambda} = 0.02$ $(\lambda = 1,2,3,...,25)$ and $t_{s_\lambda} = 0.05$ $(\lambda = 1,2,3,...,10)$, and when the step size varies.

Table 3: Numerical Comparison of Example 3 (at $t = 0.5$)

	Proposed Method	Crank-Nicolson Method
t_{s_λ}	$0.005\ (\lambda = 1,2,3,...,100)$	
$err(t)$	0.37021	0.37004
CPU time (s)	30517	17103
t_{s_λ}	$0.01\ (\lambda = 1,2,3,...,50)$	
$err(t)$	0.37021	0.3714
CPU time (s)	9375.3	7889.2
t_{s_λ}	$0.02\ (\lambda = 1,2,3,...,25)$	
$err(t)$	0.37021	\
CPU time (s)	4746.9	\
t_{s_λ}	$0.05\ (\lambda = 1,2,3,...,10)$	
$err(t)$	0.37021	\
CPU time (s)	11166	\

6 Conclusion

A temporally piecewised adaptive algorithm is presented to solve transient convection-diffusion heat transfer problems. The major merits of the proposed algorithm include

1. An initial and boundary value problem is decoupled into a series of boundary value problems which can be solved by FEM, finite volume method, meshless method, or other well-developed numerical techniques to deal with boundary value problems.

2. For the nonlinear case, there is no any requirement of additional assumption and iteration for the proposed algorithm.

Table 4: The variable process of t_{s_λ} in Example 3

λ	Time $[t_{\lambda-1}, t_\lambda]$	Step t_{s_λ}
1	$[0, 0.01]$	1×10^{-3}
2	$[0.01, 0.015]$	5×10^{-3}
3	$[0.015, 0.023]$	8×10^{-3}
4	$[0.023, 0.033]$	1×10^{-2}
5	$[0.033, 0.053]$	2×10^{-2}
6	$[0.033, 0.093]$	4×10^{-2}
7	$[0.543, 0.548]$	5×10^{-3}
8	$[0.548, 0.556]$	8×10^{-3}
9	$[0.556, 0.566]$	1×10^{-2}
10	$[0.566, 0.586]$	2×10^{-2}

3. The adaptive process can provide a steady computing accuracy when step size varies.

Acknowledgement: The research leading to this paper is funded by NSF [10421002, 10772035, 10721062, 11072043], and NKBRSF [2005CB321704, 2010CB832703].

References

Brooks, A. N.; Hughes, T. J. R. (1982): Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes Equations. *Comput Methods Appl M*, vol. 32, pp. 199-259.

Burden, R. L.; Faires, J. D. (1988): Numerical Analysis, 4^{th} (ed), PWS-KENT, Boston.

Caliari, M.; Vianello M.; Bergamaschi L. (2007): The LEM exponential integrator for advection–diffusion–reaction Equations. *J Comput Appl Math*, vol. 210, pp. 56-63.

Corre, C.; Lerat, A. (2008): High-order residual-based compact schemes for advection–diffusion problems. *Comput & Fluids*, vol. 37, pp. 505-519.

Cox, R. A.; Nishikawa, T. (1991): A new total variation diminishing scheme for the solution of advective-dominant solute transport. *Water Resour Res*, vol. 27, pp. 2645–2654.

De Palma, P.; Pascazio, G.; Rubino, D. T.; Napolitano, M. (2006): Residual distribution schemes for advection and advection–diffusion problems on quadrilateral cells. *J Comput Phys*, vol. 218, pp. 159-199.

Donea, J. (1984): A Taylor-Galerkin method for convective transport problems. *Int J Numer Meth Eng*, vol. 20, pp. 101-119.

Formaggia, L.; Nobile, F. (2004): Stability analysis of second-order time accurate schemes for ALE–FEM. *Comput Methods Appl M*, vol. 193, pp. 4097-4116.

Franca, L. P.; Valentin, F. (2000): On an improved unusual stabilized finite element method for the advective–reactive–diffusive Equation. *Comput Methods Appl M*, vol. 190, pp. 1785–1800.

Gopalakrishnan, J.; Kanschat, G. (2003): A multilevel discontinuous Galerkin method. *Numerische Mathematik*, vol. 95, pp. 527-550.

Harper, C. (1976): Introduction to Mathematical Physics, Prentice-Hall Inc., Englewood Cliffs, New Jersey.

Hou, T. Y.; Wu, X. H. (1997): A multiscale finite element method for elliptic problems in composite materials and porous media. *Comput Methods Appl M*, vol. 134, pp. 169–189.

Hughes, T. J. R.; Franca, L. P.; Hulbert, G. M. (1989): A new finite element formulation for computational fluid dynamics: VIII. The Galerkin–least–squares method for advective–diffusive Equations. *Comput Methods Appl M*, vol. 73, pp. 173–189.

Karahan, H. (2006): Implicit finite difference techniques for the advection–diffusion Equation using spreadsheets. *Adv Eng Softw*, vol. 37, pp. 601-608.

Khaliq, A. Q. M.; Voss, D. A.; Kazmi, K. (2008): Adaptive θ-methods for pricing American options. *J Comput Appl Math*, vol. 222, pp. 210-227.

Klaij, C. M.; van der Vegt, J. J. W.; van der Ven, H. (2006): Pseudo-time stepping methods for space–time discontinuous Galerkin discretizations of the compressible Navier–Stokes Equations. *J Comput Phys*, vol. 219, pp. 622-643.

Klaij, C. M.; van Raalte, M. H.; van der Ven, H.; van der Vegt, J. J. W. (2007):*h*-Multigrid for space-time discontinuous Galerkin discretizations of the compressible Navier–Stokes Equations. *J Comput Phys*, vol. 227, pp. 1024-1045.

Labeur, R. J.; Wells, G. N. (2007): A Galerkin interface stabilisation method for the advection–diffusion and incompressible Navier–Stokes Equations. *Comput Methods Appl M*, vol. 196, pp. 4985-5000.

Leonard, B. P. (1979): A stable and accurate convective modelling procedure based on quadratic upstream interpolation. *Comput Methods Appl M*, vol. 19, pp. 59–98.

Li, X. K.; Wu, W. H.; Zienkiewicz, O. C. (2000): Implicit Characteristic Galerkin Method for convection-diffusion Equations. *Int J Numer Meth Eng*, vol. 47, pp. 1689-1708.

Peraire, J.; Zienkiewicz, O. C.; Morgan, K. (1986): Shallow water problems: a general explicit formulation. *Int J Numer Meth Eng*, vol. 22, pp. 547–574.

Platten, J. K.; Legros, J. C. (1984): Convection in Liquids, Springer-Verlag, New York.

Ponsoda, E.; Defez, E.; Roselló, M. D.; Romero, J. V. (2008): A stable numerical method for solving variable coefficient advection–diffusion models. *Comput Math Appl*, vol. 56, pp. 754-768.

Pontaza, J. P. (2007): A new consistent splitting scheme for incompressible Navier–Stokes flows: A least-squares spectral element implementation. *J Comput Phys*, vol. 225, pp. 1590-1602.

Rodríguez, N. J.; Davey, K.; Vázquez Feijoo, J. A.; Juárez-Hernandez, A. (2009): A. Juárez-Hernandez, Numerical modelling of unsteady convective–diffusive heat transfer with a control volume hybrid method. *Appl Math Model*, vol. 33, pp. 897-923.

Rubio, A. D.; Zalts, A.; El Hasi, C. D. (2008): Numerical solution of the advection–reaction–diffusion Equation at different scales. *Environ Modell Softw*, vol. 23, pp. 90-95.

Rui, H. X. (2008): A Conservative Characteristic Finite Volume Element Method for solution of the Advection-diffusion Equation. *Comput Method Appl M*, vol. 197, pp. 3862-3869.

Shih, T. M. (1983): Numerical Properties and Methodologies in Heat Transfer, Springer-Verlag, New York.

Sudirham, J. J.; van der Vegt, J. J. W.; van Damme, R. M. J. (2006): Space-time discontinuous Galerkin method for advection-diffusion problems on time-dependent domains. *Appl Numer Math*, vol. 56, pp. 1491-1518.

Witek, M. L.; Teixeira, J.; Flatau. P. J. (2008): On stable and explicit numerical methods for the advection–diffusion Equation. *Math Comput Simulat*, vol. 79, pp. 561-570.

Wong, K. T. M.; Lee, J. H. W.; Choi, K. W. (2008): A deterministic Lagrangian particle separation-based method for advective-diffusion problems. *Commun Nonlinear Sci*, vol. 13, pp. 2071-2090.

Yuan, W.F.; Tan, K.H. (2009): Heat Transfer in Composite Beams using Combined Cellular Automaton and Fibre Model. *CMC: Computers, Materials & Continua*, vol.13, no.1, pp.49-62.

Zienkiewicz, O. C.; Taylor, R. L.(2000): The finite element method, 5^{th} (ed), Butterworth-Heinemann, Oxford.