131 Mathcad[1] Worksheets for Partial Differential Equations

Address:	Prof. Dr. K. Schittkowski Department of Mathematics University of Bayreuth D - 95440 Bayreuth
Phone:	+921 553278 (office) +921 32887 (home)
Fax:	+921 35557
E-mail:	klaus.schittkowski@uni-bayreuth.de
Web:	http://www.klaus-schittkowski.de
Date:	March 23, 2004

Abstract

The purpose of the paper is to introduce a set of Mathcad worksheets containing partial differential equations (PDEs). They can be used to become familiar with Mathcad implementation of PDEs and with the behavior of dynamical systems in general. The problems are taken from a collection of test examples for data fitting in dynamical systems, see Schittkowski [59]. The report contains a summary of 131 partial differential equations that have been transferred to Mathcad and a detailed example. All worksheets can be downloaded from the home page of the author[2]. A particular advantage of executing these problems from Mathcad is the possibility to plot corresponding solutions very easily.

[1] ©2003 Mathsoft Engineering & Education Inc.
[2] http://www.klaus-schittkowski.de

1 Introduction

Time-dependent partial differential equations can be considered as extensions of ordinary differential equations, if we allow that the right-hand side depends on first and second derivatives of the model functions with respect to an additional space or spatial variable x. For example, they could describe the dynamic behavior of a vibrating beam over time.

In our case, we consider only one-dimensional partial differential equations with $x \in [x_L, x_R]$, and the state variables are denoted by

$$u(x,t) = (u_1(x,t), \ldots, u_{n_p}(x,t))^T$$

depending on the time variable t and the spatial variable x. x_L and x_R are called the left and right boundaries of x.

First, we introduce the subscript notation

$$u_t(x,t) = \frac{\partial u(x,t)}{\partial t} \quad , \quad u_x(x,t) = \frac{\partial u(x,t)}{\partial x} \quad , \quad u_{xx}(x,t) = \frac{\partial^2 u(x,t)}{\partial x^2} \quad .$$

The general one-dimensional partial differential equation under consideration is

$$u_t = F(u, u_x, u_{xx}, x, t) \quad . \tag{1}$$

If we consider individual coefficient functions

$$F(u, u_x, u_{xx}, x, t) = (F_1(u, u_x, u_{xx}, x, t), \ldots, F_{n_p}(u, u_x, u_{xx}, x, t))^T \quad ,$$

we write (1) also in the form of a system of scalar equations

$$\begin{array}{rcl} \frac{\partial u_1}{\partial t} & = & F_1(u, u_x, u_{xx}, x, t) \quad , \\ & \cdots & \\ \frac{\partial u_{n_p}}{\partial t} & = & F_{n_p}(u, u_x, u_{xx}, x, t) \quad . \end{array} \tag{2}$$

We call (1) or (2), respectively, a time-dependent system in explicit formulation, since it contains the partial derivative with respect to t explicitly at the left-hand side.

We do not require that the right-hand side of (2) always depends explicitly on u_x or u_{xx}, respectively. It is even allowed that both spatial derivatives vanish. Moreover, linear and nonlinear equations are handled in the same way. Since systems of partial differential equations depend on several independent parameters, x and t in our case, they are also called distributed systems. The model function F is defined on $\mathbb{R}^{n_p} \times \mathbb{R}^{n_p} \times \mathbb{R}^{n_p} \times \mathbb{R} \times \mathbb{R}$, and has function values in $\mathbb{R}^{n_p}$. We assume for the moment that F is at least a continuous function in all variables.

To complete the definition of a partial differential equation (PDE), we need initial values

$$u(x,0) = u_0(x) \quad , \tag{3}$$

where we assume again that the integration is to be performed over a time interval of the form $[0,T]$ with a sufficiently large bound T. $u_0(x)$ is a given function defined on the integration area $[x_L, x_R]$.

Classical boundary conditions are either of Dirichlet type

$$\begin{aligned} u(x_L,t) &= u^L(t) \quad , \\ u(x_R,t) &= u^R(t) \quad , \end{aligned} \tag{4}$$

with given functions $u^L(t)$ and $u^R(t)$ defined on the whole time interval $(0,T]$, or of Neumann type

$$\begin{aligned} u_x(x_L,t) &= \hat{u}^L(t) \quad , \\ u_x(x_R,t) &= \hat{u}^R(t) \quad , \end{aligned} \tag{5}$$

where derivative or flux functions $\hat{u}^L(t)$ and $\hat{u}^L(t)$ are known for all $t \in (0,T]$. We have the option to fix either the function values of the solution or its spatial derivatives, to mix both conditions, or to omit one or another boundary condition completely.

The number and order of boundary conditions depends on the model structure, for example whether F depends on u_{xx} or not, in order to get a uniquely defined solution. As a rule of thumb, the total number of scalar boundary conditions should coincide with the number of all partial differentiations, where the highest order of spatial differentiation is counted for each state variable. Boundary and initial conditions are called inconsistent, if they do not fit continuously at connecting points, for instance if $u^L(0) \neq u_0(x_L)$ in case of a left Dirichlet boundary condition.

As indicated above, a solution is denoted by $u(x,t)$. However, we must be very careful when trying to define the *solution* of a PDE in a mathematically rigorous way. The answer is not obvious, since we allow inconsistent boundary conditions and the propagation of non-continuous transitions through the integration area, for example in case of propagation of shocks. In these cases, one has to develop a concept of weak solutions of a PDE. More theoretical investigations of this question are found in textbooks on partial differential equations, see Zachmanoglou and Thoe [81] or Renardy and Rogers [51]. In most situations, it is sufficient to assume for $u(x,t)$ to become a solution, that u satisfies (2) for all $x \in (x_L, x_R)$ and for all $t \in (0,T)$.

Typically, partial differential equations are classified by geometric terms like *parabolic* or *hyperbolic*. Originally, they were defined for two-dimensional, second-order partial differential equations, linear in the second-order terms with constant coefficients, see van Kan and Segal [29], Schittkowski [59], or any other introductory textbook on partial differential equations.

To give an example, let us consider parabolic partial differential equations In this case, we assume that F does not depend on the first spatial derivative u_x, only on u and u_{xx}. Thus, we write a parabolic PDE in the form

$$u_t = F(u, u_{xx}, x, t)$$

with suitable initial and boundary conditions of the type (3), (4) or (5), respectively. Some of the most important physical interpretations of parabolic equations are diffusion processes describing transportation phenomena through media.

Example 1.1 *The probably most popular partial differential equation is the heat equation describing the conduction of heat in a solid, which is considered now in the form of a special variant. Let T be the temperature of the solid, x the spatial position along a dominating direction of the solid, and t the time. From Fourier's second law for heat conduction, we get the equation*

$$T_t = D\, T_{xx} \tag{6}$$

with a thermal diffusion constant $D > 0$. The spatial variable x varies from 0 to L and the time variable t must be non-negative, $t \in [0, T]$. Numerous initial values and boundary conditions are found in the literature. In our case, we set an initial heat distribution $T(x, 0) = \sin(\pi x/L)$ at time $t = 0$ for all $x \in (0, L)$ and Dirichlet boundary conditions $T(0, t) = T(L, t) = 0$ for all $t \geq 0$. It is easy to verify by insertion that

$$T(x, t) = e^{-tD\pi^2/L^2} \sin(x\pi/L)$$

is a solution of the PDE. Typical for parabolic equations is the exponential damping of the initial values along the t-axis, see Figure1.

The report is one out of a series of Mathcad test problem collections by which numerical routines are tested and the implementation of optimization problems and dynamical systems is outlined, i.e.,

1. nonlinear programming [61],
2. data fitting [62],
3. ordinary differential equations [63],
4. differential algebraic equations [64],
5. partial differential algebraic equations [65].

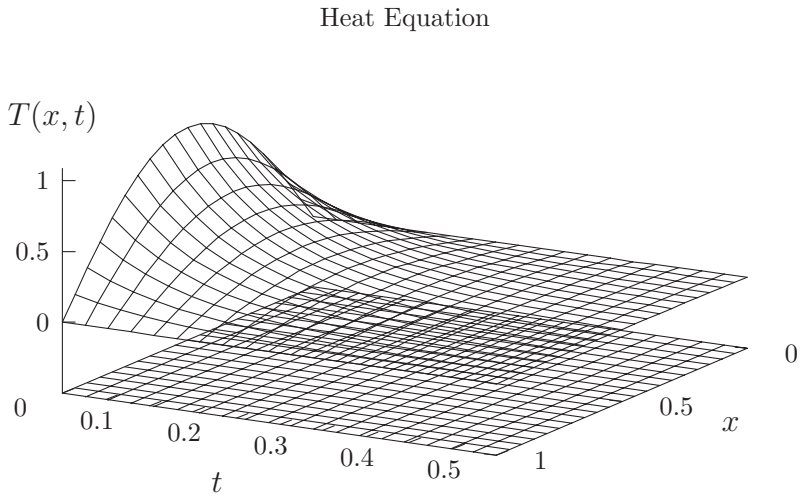

Figure 1: Solution of a Parabolic Equation

Section 2 contains a brief outline of the method of lines by which a PDE is transformed into a system of ordinary differential equations (ODEs). Subsequently, an implicit integration routine is used to integrate these equations. This basic idea is implemented in the routine *Pdesolve* in Mathcad, which is used for all test cases. A simple example is shown in Section 3 to illustrate the preparation of the model equations for a partial differential equation. A list of the Mathcad worksheet files and some further details about problem structure, background, and source is given in Section 4.

2 The Method of Lines

The underlying idea is to transform the partial differential equations into a system of ordinary differential equations by discretizing the model functions with respect to the spatial variable x. This approach is known as the *method of lines*, see for example Schiesser [53] or Schittkowski [59]. We denote the number of spatial discretization points by n, and proceed from a uniform distribution of grid points for simplicity. Instead of a function $u(x,t)$, we consider now a family of approximating time-dependent functions $u_i(t) = u(x_i,t)$ for $i = 1, \ldots, n$, where

$$x_i = x_L + \frac{i-1}{n-1}(x_R - x_L) \quad . \tag{7}$$

The next step consists of computing an appropriate approximation of the first spatial

derivative $u_x(x,t)$ at a line $x = x_i$. To demonstrate the principal idea, we apply the two-sided central difference formula

$$u_x(x_i,t) \simeq \frac{1}{2h}\Big(u_{i+1}(t) - u_{i-1}(t)\Big) \tag{8}$$

with $h = 1/(n-1)$ and $i = 2, \ldots, n-1$. For approximating derivatives at boundaries, we use

$$\begin{aligned} u_x(x_1,t) &\simeq \frac{1}{2h}\Big(-u_3(t) + 4u_2(t) - 3u_1(t)\Big) \ , \\ u_x(x_n,t) &\simeq \frac{1}{2h}\Big(3u_n(t) - 4u_{n-1}(t) + u_{n-2}(t)\Big) \ . \end{aligned} \tag{9}$$

For numerical realization, one has to take care of the desired discretization accuracy. Higher-order formulae are available that guarantee the same approximation order at the boundaries and the interior of the integration area.

Assume that Dirichlet boundary conditions (4) are given. Then we know the boundary functions

$$u_1(t) = u^L(t) \ , \quad u_n(t) = u^R(t) \tag{10}$$

exactly which are inserted into (9).

For approximating $u_{xx}(x,t)$, we have either the possibility to use the formula for first derivatives recursively, or to derive a special formula for second derivatives. Assume now that second derivatives at $x = x_i$ are approximated by

$$u_{xx}(x_i,t) \simeq \frac{1}{h^2}\Big(u_{i+1}(t) - 2u_i(t) + u_{i-1}(t)\Big) \tag{11}$$

for $i = 2, \ldots, n-1$, and that boundary approximations are obtained from

$$\begin{aligned} u_{xx}(x_1,t) &\simeq \frac{1}{h^2}\Big(u_3(t) - 2u_2(t) + u_1(t)\Big) \ , \\ u_{xx}(x_n,t) &\simeq \frac{1}{h^2}\Big(u_n(t) - 2u_{n-1}(t) + u_{n-2}(t)\Big) \ . \end{aligned} \tag{12}$$

Again, boundary functions (10) are inserted directly as far as known. If Neumann boundary conditions of the form (5) are given, we replace approximations of first derivatives at boundaries by known functions $\hat{u}^L(t)$ or $\hat{u}^R(t)$,

$$u_x(x_1,t) = \hat{u}^L(t) \ , \quad u_x(x_n,t) = \hat{u}^R(t) \ , \tag{13}$$

and insert them into the approximation formulae for second derivatives leading to

$$\begin{aligned} u_{xx}(x_1,t) &\simeq \frac{1}{h}\Big(\frac{1}{2h}(u_3(t) - u_1(t)) - \hat{u}^L(t)\Big) \ , \\ u_{xx}(x_n,t) &\simeq \frac{1}{h}\Big(\hat{u}^R(t) - \frac{1}{2h}(u_n(t) - u_{n-2}(t))\Big) \end{aligned} \tag{14}$$

in our simplified case study.

After inserting (8) to (14) into the right-hand side function F of (1), we get a system of ordinary differential equations of the form

$$\dot{u}_i(t) = F(u_i(t), d_i^1(u,t,h), d_i^2(u,t,h), x_i, t) \quad , \tag{15}$$

$i = 1, \ldots, n$. Here we replace the partial derivative subject to t by the total derivative indicated by the dot, and $d_i^1(u,t,h)$, $d_i^2(u,t,h)$ denote the approximation formulae for first and second derivatives with $u(t) = (u_1(t), \ldots, u_n(t))^T$. If n_p denotes the number of partial differential equations, we obtain a set of $n_p\, n$ ordinary differential equations after discretization.

Redundant differential equations belonging to Dirichlet boundary conditions are neglected to reduce the total number of equations. If, for example, each component of the PDE has a left and right Dirichlet condition, index i in (15) runs from 2 to $n-1$, and we get a system of $n_p\,(n-2)$ ordinary differential equations.

From the initial values (3), $u(x,0) = u_0(x)$, we obtain immediately initial values for the ordinary differential equations

$$u_i(0) = u_0(x_i) \tag{16}$$

for $i = 1, \ldots, n$. This completes the transformation of one-dimensional, time-dependent partial differential equations into a system of ordinary differential equations by the method of lines. The resulting ODE can be solved by any available solver, for example the implicit Radau-type method of Hairer and Wanner [23].

However, there is one difficulty when applying the method of lines. It turns out that the system of ordinary differential equations becomes stiff with decreasing discretization accuracy h, i.e., with increasing number of discretization points n. Thus, it is recommended that implicit solution methods are implemented as soon as h becomes sufficiently small, requiring computation of the Jacobian matrix of the right-hand side of (15) subject to u_i, $i = 1, \ldots, n$. Depending on the discretization procedure chosen, the matrix has a band structure that can be exploited when solving the corresponding system of linear equations.

3 A Mathcad Worksheet Example

Mathcad (`http://www.mathcad.com`) is an interactive GUI with a large number of built-in mathematical functions. Special commands allow to solve systems of one-dimensional partial differential equations. The subsequent lines describe the usage of *Pdesolve* for solving PDEs, see also the Mathcad documentation

The Pdesolve function is used within solve blocks, allowing for natural notation, and are the easiest to use and interpret.

Pdesolve(u, x, xrange, t, trange, [xpts], [tpts])) returns a function or vector of functions of x and t that solve a one-dimensional nonlinear PDE or a system of PDEs, see (1). Values are interpolated from a matrix of solution points calculated using the numerical method of lines.

Arguments:

- u is the explicit vector of function names (with no variable names included) precisely as they appear within the solve block. This argument degenerates to a scalar in the case of a single PDE.
- x is the spatial variable.
- *xrange* is a 2-element column vector containing the boundary values for x. Both values must be real.
- t is the time variable.
- *trange* is a 2-element column vector containing the boundary values for t. Both values must be real.
- *xpts* (optional) is the integer number of spatial discretization points.
- *tpts* (optional) is the integer number of temporal discretization points.

A *solve block* refers to a group of steps involved when solving a system of differential equations. Needed are initial the key word *Given*, a set of equations, and the solving function *Pdesolve*. Collectively, these steps are known as a solve block.

To give an impression how a test problem is implemented, we consider problem HEAT, see Example 6. Subsequently, the Mathcad implementation is listed, see Figure 2.

HEAT

Partial Differential Equation

Description: Heat equation

Constants: $L := 1 \qquad T := 0.5$

Differential Equations: Given

$$u_t(x,t) = u_{xx}(x,t)$$

Initial Values:

$$u(x,0) = \sin\left(\pi \cdot \frac{x}{L}\right)$$

Boundary Values: $u(0,t) = 0 \qquad u(1,t) = 0$

Integration:

$$u := \text{Pdesolve}\left[u, x, \begin{pmatrix} 0 \\ L \end{pmatrix}, t, \begin{pmatrix} 0 \\ T \end{pmatrix}, 20, 30\right]$$

Plot: $G := \text{CreateMesh}(u, 0, L, 0, T, 20, 30)$

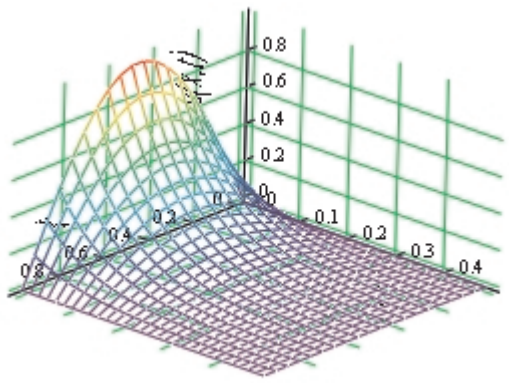

Figure 2: Mathcad Implementation: Heat Equation

4 List of All Test Problems

The subsequent table contains a list of all test problems together with the number of partial differential equations n_p, a brief description of the practical or mathematical background, and some references. The differential equations have first been implemented in the modelling language PCOMP, see Dobmann et al. [12] or Schittkowski [59, 58, 60]. The transformation into Mathcad worksheets follows a unified format based on the PCOMP equations. Thus, the implementations do not exploit all possible features of Mathcad to get the most elegant and compact description. All mcd-files can be downloaded from the home page of the author[3].

It should be noted that some solutions are highly oscillating or at least very time-consuming because of a narrow grid. Among the test problems are a few hyperbolic equations with non-continuous boundary conditions, which can be accurately solved only by special shock-absorbing upwind formulae, see Schittkowski [59] for details. Among them are several variants of advection equations and Burger's equation, for example.

Partial Differential Equations

name	n_p	*background*	*ref*
ADV_DIFF	3	Advection-diffusion equation with Riemann initial data	[30]
ADV_VC	1	Advection equation with variable coefficient	[68]
ADV_VTC	2	Advection equations with variable time coefficient	[68]
ADVEC_2N	1	Nonlinear unsteady advection (n=2)	[75]
ADVEC_5N	1	Nonlinear unsteady advection (n=5)	[75]
ADVEC_LU	1	Linear unsteady advection-diffusion	[75]
ADVEC_S	1	Linear steady advection-diffusion with source term	[75]
ADVECT	1	Advection equation, first-order hyperbolic PDE	[53]
ADVECT_N	1	Advection equation with a nonlinear source term	[45]
ADVECT_R	1	Advection equation with right boundary value	[53]
ADVECT2	2	Two advection equations (different flux directions)	[53]
BLOW_UP	1	Degenerated parabolic equation with blow-up	[21]
BRAIN	1	Transport phenomena in brain tissue	[5]
BRUSSEL	2	Brusselator with diffusion	[20], [57]

(continued)

[3] http://www.klaus-schittkowski.de

name	n_p	*background*	*ref*
BSE	1	Black-Scholes equation governing price of derivative security	[6], [79]
BUBB_BIO	3	Bubble column bio-reactor	[41]
BUBBLE	2	Dynamic oxygen uptake of water in bubble column	
BURGER	1	Parabolic Burger's equation with exact solution	[67]
BURGER_E	1	Viscous Burger's equation with exact solution, mue=0.01	[53], [57]
BURGER_I	1	Burger's equation in the inviscid limit	[80], [45]
BURST	2	Crisis induced intermittent bursting in reaction-diffusion chemical systems	[18], [17]
CD_TRANS	1	Convective-dispersive transport equation with nonlinear reactions	[31]
CNT_CUR1	2	Counter-current separation of fluid phase concentrations with phase equilibrium	[47]
CON_DIV1	1	Periodic convection dominated diffusion	[35]
CON_DIV2	1	Periodic convection dominated diffusion	[35]
CSE	2	Cubic Schroedinger equation with one soliton	[54]
CTFLOW_P	2	Two incompressible counter-current flows of binary liquid mixture with permeable wall	[40]
CUBIC	1	Cubic conservation law with Riemann data	[24]
DC_TUBE	1	Diffusion-convection in a tube	
DEHYDRO	2	Dehydrogenization of ethylbenzene to styrene in a tubular reactor	[76]
DESIGN	1	First-order hyperbolic PDE, inhomogeneous part	[71]
DIFF_1D	1	Diffusion problem with Dirichlet and Neumann boundary conditions	
DIFF_ADS	2	Diffusion and absorption reaction	
DIFF_ETH	1	Diffusion of ethanol in water	[76], [25]
DIFF_NLB	1	Nonlinear diffusion with nonlinear boundary condition	[67]
DIFFPT	1	Diffusion and partitioning in biological systems, non-continuous transition	[38]
DIFFUS	1	Diffusion equation with constant parameters	
DISRE	1	Non-isothermal tubular reactor with axial dispersion	[28], [57]
DISRET	2	Non-isothermal tubular reactor with axial dispersion	[28], [57]
ECOLOGY	2	Population ecology with planktonit predator-prey and crowding	[32]
ELECTRO	2	Electrodynamic application	[7], [57]
ELLIPTIC	1	Elliptic test problem	[67]

(continued)

name	n_p	*background*	*ref*
ENZDYN	2	Dynamic diffusion and enzymatic reaction	[28]
FILTWASH	1	Filter washing	[28]
FINAG	2	Nerve conduction	[42]
FIXBED	2	Catalytic fixed bed reactor with one exothermal reaction	[13], [15]
FLAME	2	Dwyer-Sanders flame propagation model	[14], [73], [57]
FRONT	2	Flame propagation model with non-constant moving front	[46]
GAS_CONV	1	Gas convection	
GROWTH	1	Logistic model of population growth (Fisher's equation)	[70]
HEAT	1	Heat equation	[53], [57]
HEAT_BD3	1	Nonlinear heat equation, boundary conditions of third type	
HEAT_CD	1	One-dimensional heat conduction	[52]
HEAT_CW	1	Graetz problem with constant wall heat flux	[52]
HEAT_CYL	1	Cylindrical heat transfer	[67]
HEAT_EX	1	Tubular heat exchanger	[52]
HEAT_MS	2	Heat transport equation at the microscale (3rd order)	[78]
HEAT_NLB	1	Heat equation with nonlinear boundary condition of Stefan-Boltzmann type	[69]
HEAT_TDC	1	Heat diffusion with time-dependent diffusion parameter	
HOT_SPOT	1	'Hot Spot' problem from combustion theory	[73], [57]
HYG_POLY	1	Diffusion of water into a hygroscopic polymer	
HYP2ND	2	Hyperbolic equation of second order, alternating cosine waves	
HYPER	2	System of two advection equations, first-order hyperbolic PDEs	
HYPERBO1	2	Hyperbolic test system	[4]
HYPERBO2	2	Hyperbolic test system	[4]
HYPERBO3	2	Hyperbolic test system	[4]
HYPERBO4	2	Hyperbolic test system	[4]
HYPERBO5	2	Hyperbolic test system	[4]
IN_LAYER	2	Catalyst with inert layers (diffusion, absorption, desorption)	
INV_PROB	1	Inverse problem in heat conduction	[22]
KILN	1	Heating a probe in a kiln	[10]
LAM_FLOW	1	Unsteady laminar flow in a circular tube	[52], [27]
LDCP	1	Linear diffusion-convection equation	[49], [47]
LIN_HC	1	Linear heat conduction	[1]
LIN_HYP1	1	First-order linear hyperbolic equation	[77]

(continued)

name	n_p	*background*	*ref*
LOSSLESS	2	Lossless electric transmission line	[54]
MOL_DIFF	1	Molecular diffusion (boundary value problem)	[37]
MOVFRONT	1	Moving front (Burger's equation)	[1]
NERVE	2	Nerve pulse	[74]
NL_HEAT	1	Nonlinear heat equation	[70]
NL_PDE	1	Highly nonlinear PDE with exact solution	[54]
NLINPDE	2	Two nonlinear PDE's with exact solution	[53], [34]
NLSE	2	Nonlinear Schroedinger equation, exact soliton solution (complex)	[55]
NOISE	1	Nonlinear deblurring and noise removal	[36]
NON_AD	1	Nonlinear advection-diffusion equation	[30]
OBSTACLE	2	Shallow water flow over an obstacle	[33]
ONESTEP	2	One-step reaction with diffusion and non-unit Lewis number	[1]
OSC_SOL	3	Oscillatory solution of hyperbolic PDE	[19]
PAR_CTRL	1	Parabolic optimal control problem	[39]
PAR_SIN	1	Parabolic PDE with inhomogeneous sinus-term	[50], [48]
PARAB1	1	Brain transport	[5]
PARAB2	2	Parabolic equation, identifiability test	[5]
PARAB3	1	Parabolic equation, identifiability test	[5]
PARAB4	1	Parabolic equation, identifiability test	[5]
PARAB5	1	Parabolic equation, identifiability test	[5]
PARAB6	1	Parabolic equation, identifiability test	[5]
PHYP_PBC	1	Parabolic-hyperbolic equation with periodic boundary conditions	[2]
POLLUTN	4	SST pollution in the stratosphere	[67], [57]
POLY_DYN	1	Chain length of polymerization process	
QUENCH1	1	Degenerate nonlinear quenching	[66]
QUENCH2	1	Degenerate nonlinear quenching	[66]
REA_DIF1	1	Reaction-diffusion equation	[16]
REA_DIF2	1	Reaction-diffusion equation	[16]
RESERVOI	1	Reservoir simulation by the Buckley-Leverett equation	[72]
ROD	1	Rod of solid explosive	
SH_FRONT	1	PDE with sharp front, exact solution known	[11]
SHEAR	3	Shear band formation	[43], [20], [57]
SIN_GOR1	2	Sine-Gordon equation, exact kink-soliton solution	[55]
SIN_GOR2	2	Sine-Gordon equation, exact kink-kink-collision solution	[55]

(continued)

name	n_p	*background*	*ref*
SINGSTEP	1	Single-step reaction with diffusion	[1]
SLAB	3	Dwyer-Sanders flame propagation model	[44]
SOLID	1	Heating of solid sphere	[3]
SOLITON	2	Kink soliton (Sine-Gordon equation)	
SPHERE	1	Heat conduction in sphere with exothermic chemical reaction	[53]
STARTBED	1	Diffusion	
STFFDET1	1	Stiffness detection	[16]
STFFDET2	1	Stiffness detection	[16]
STR_FISH	1	Stream fish tracked by mark-recapture technique	
TELEGRPH	2	Telegraph equation	[54]
TIME_OPT	1	Time-optimal heat distribution	[56]
TONGUE	1	Motion of glacier tongue	[9]
TRAV_WAV	1	Traveling waves (Burger's equation, exact solution known)	[1]
TUBE0	1	Zero-order reaction in a catalytic-walled tube	[8]
TWO_POPS	2	Two populations	[70]
VAR_VELO	1	First-order linear hyperbolic equation with variable velocity field	[77]
VISCOUS	5	Variable viscosities with periodic boundary	[2]
WATER	2	Flow of shallow water over a barrier	[67], [26]
WAVE1	2	Hyperbolic wave equation (exact solution known)	[53]
WAVE2	2	Wave equation in form of two hyperbolic equations	
WAVE3	2	Hyperbolic wave equation	[76]
WAVE4	2	Two waves travelling in opposite directions, semi-hyperbolic system	[73], [57]

Acknowledgement: The author would like to thank Kathrin Maul, Fiona Fleischmann, Angela Busse, Irina Hübner, Dominik Stadelmaier, and Florian Bauer for preparing the Mathcad worksheets.

References

[1] Adjerid S., Flaherty J.E. (1986): *A moving finite element method with error estimation and refinement for one-dimensional time dependent partial differential equations*, SIAM Journal on Numerical Analysis, Vol. 23, 778-796

[2] Ascher U., Ruuth S., Wettin B. (1995): *Implicit-explicit methods for time-dependent partial differential equations*, SIAM Journal on Numerical Analysis, Vol. 32, 797-823

[3] Balsa-Canto E., Alonso A.A., Banga J.R. (2002): *A novel, efficient and reliable method for thermal process design and optimization. Part I: Theory*, Journal of Food Engineering, Vol. 52, 227-234

[4] Banks H.T., Crowley J.M., Kunisch K. (1983): *Cubic spline approximation techniques for parameter estimation in distributed systems*, IEEE Transactions on Automatic Control, Vol. AC-28, No. 7, 773-786

[5] Banks H.T, Kunisch K. (1989): *Estimation Techniques for Distributed Parameter Systems,* Birkhäuser, Boston, Basel, Berlin

[6] Black F., Scholes M. (1973): *The pricing of options and corporate liabilities*, Journal of Political Economics, Vol. 81, 637-659

[7] Blom J.G., Zegeling P.A. (1994): *Algorithm 731: A moving grid interface for systems of one-dimensional time-dependent partial differential equations*, ACM Transactions on Mathematical Software, Vol. 20, No. 2, 194-214

[8] Caracotsios M., Stewart W.E. (1985): *Sensitivity analysis of initial value problems with mixed ODE's and algebraic equations*, Computers and Chemical Engineering, Vol. 9, 359-365

[9] Collatz L. (1960): *The Numerical Treatment of Differential Equations*, Springer, Berlin

[10] Dennis J.E.jr., Heinkenschloss M., Vicente L.N. (1998): *Trust-region interior-point SQP algorithm for a class of nonlinear programming problems*, SIAM Journal on Control, Vol. 36, No. 5, 1750-1794

[11] Dietrich E.E., Eigenberger G. (1996): *Compact finite difference methods for the solution of chemical engineering problems*, in: Scientific Computing in Chemical Engineering, Keil, Mackens, Voss, Werther eds., Springer, Berlin

[12] Dobmann M., Liepelt M., Schittkowski K. (1995): *Algorithm 746: PCOMP: A Fortran code for automatic differentiation,* ACM Transactions on Mathematical Software, Vol. 21, No. 3, 233-266

[13] van Doesburg H., De Jong W.A. (1974): *Dynamic behavior of an adiabatic fixed-bed methanator*, in: Advances in Chemistry, Vol. 133, International Symposium on Reaction Engineering, Evanston, 489-503

[14] Dwyer H.A., Sanders B.R. (1978): *Numerical modeling of unsteady flame propagation*, Acta Astronautica, Vol. 5, 1171-1184

[15] Eigenberger G., Butt J.B. (1976): *A modified Crank-Nocolson technique with non-equidistant space steps*, Chemical Engineering Sciences, Vol. 31, 681-691

[16] Ekeland K., Owren B., Oines E. (1998): *Stiffness detection and estimation of dominant spectra with explicit Runge-Kutta methods*, ACM Transaction on Mathematical Software, Vol. 24, No. 4, 368-382

[17] Elezgaray J., Arneodo A. (1992): *Crisis induced intermittent bursting in reaction-diffusion chemical systems*, Physical Reviews Letters, Vol. 68, 714-717

[18] Engleborghs K., Lust K., Roose D. (1999): *Direct computation of periodic doubling bifurcation points of large-scale systems of ODE's using a Newton-Picard method*, IMA Journal of Numerical Analysis, Vol. 19, 525-547

[19] Engquist, B. (1986): *Computation of oscillatory solutions to partial differential equations*, in: C. Carasso, P.-A. Raviart, D. Serre eds., Nonlinear Hyperbolic Problems, Lecture Notes in Mathematics, No. 1270, Springer, Berlin

[20] Flaherty J.E., Moore P.K. (1995): *Integrated space-time adaptive hp-refinement methods for parabolic methods*, Applied Numerical Mathematics, Vol. 16, 317-341

[21] Friedman A., McLead B. (1986): *Blow-up of solutions of nonlinear degenerate parabolic equations*, Archive for Rational Mechanics and Analysis, Vol. 96, 55-80

[22] Hao D.N., Reinhardt H.-J. (1998): *Gradient methods for inverse heat conduction problems*, in: Inverse Problems in Engineering, Vol. 6, No. 3, 177-211

[23] Hairer E., Wanner G. (1991): *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems,* Springer Series Computational Mathematics, Vol. 14, Springer, Berlin

[24] Hayes B.T., Lefloch P.G. (1998): *Nonclassical shocks and kinetic relations: finite difference schemes*, SIAM Journal on Numerical Analysis, Vol. 35, No. 6, 2169-2194

[25] Hines A.L., Maddox R.N. (1985): *Mass Transfer*, Prentice-Hall, Englewood-Cliffs

[26] Houghton D.D., Kasahara A. (1968): *Nonlinear shallow flow over an isolated ridge*, Communications of Pure and Applied Mathematics, Vol. 21, 1-23

[27] Hughes W.F., Brighton J.A. (1991): *Theory and Problems of Fluid Dynamics*, McGraw Hill, New York

[28] Ingham J., Dunn I.J., Heinzle E., Prenosil J.E. (1994): *Chemical Engineering Dynamics*, VCH, Weinheim

[29] van Kan J.J.I.M., Segal A. (1995): *Numerik partieller Differentialgleichungen für Ingenieure,* Teubner

[30] Karlsen K.H., Lie K.-A. (1999): *An unconditionally stable splitting scheme for a class of nonlinear parabolic equations*, IMA Journal of Numerical Analysis, Vol. 19, 609-635

[31] Kojouharov, Chen B.M. (1999): *Nonstandard methods for the convective-dispersive transport equation with nonlinear reactions*, Numerical Methods for Partial Differential Equations, Vol. 16, No. 1, 107-132

[32] Lang J. (1993): *KARDOS: Kascade reaction diffusion one-dimensional system*, Technical Report TR 93-9, ZIB Berlin

[33] Liska R., Wendroff B. (1998): *Composite schemes for conservation laws*, SIAM Journal on Numerical Analysis, Vol. 35, No. 6, 2250-2271

[34] Madsen N.K., Sincovec R.F. (1976): *Software for partial differential equations*, in: Numerical Methos for Differential Systems, L. Lapidus, W.E. Schiesser eds., Acedemic Press, New York

[35] Marion M., Mollard A. (1999): *A multilevel characteristics method for periodic convection-dominated diffusion problems*, Numerical Methods for Partial Differential Equations, Vol. 16, No. 1, 107-132

[36] Marquina A, Osher S. (2000): *Explicit algorithms for a new time-dependent model based on level set motion for nonlinear deblurring and noise removal*, Report, Dept. of Mathematics, University of California, Los Angeles

[37] Meissner E. (2000): *Messung von kurzen Konzentrationsprofilen mit Hilfe der analytischen TEM-EDX am Beispiel der Bestimmung von Diffusionskoeffizienten für Mg-Fe Interdiffusion in Olivin, Dissertation*, Faculty of Biology, Chemistry, and Geological Sciences, University of Bayreuth

[38] Missel P.J. (2000): *Finite element modeling of diffusion and partioning in biological systems*, Report, Drug Delivery, Alcon Research Ltd., Fort Worth, USA

[39] Mittelmann H.D. (2001): *Sufficient optimality for discretized parabolic and elliptic control problems*, in: Fast Solution of Discretized Optimization Problems, K.-H. Hoffmann, R.H.W. Hoppe, and V. Schulz (eds.), ISNM 138, Birkhäuser, Basel

[40] Molander M. (1990): *Computer aided modelling of distributed parameter process*, Technical Report No. 193, School of Electrical and Computer Engineering, Chalmers University of Technology, Göteborg, Sweden

[41] Munack A. (1995): *Simulation bioverfahrenstechnischer Prozesse*, in: Prozessimulation, H. Schuler ed., VCH, Weinheim, 409-455

[42] Naguma J., Arimoto S., Yoshizawa (1962): *An active pulse transmission line simulating nerve axon*, Proceedings of the IRE, Vol. 50, 2061-2070

[43] Nowak U. (1995): *A fully adaptive MOL-treatment of parabolic 1D-problems with extrapolation techniques*, Preprint SC 95-25, ZIB Berlin

[44] Otey G.R., Dwyer H.A. (1979): *Numerical study of the interaction of fast chemistry and diffusion*, AIAA Journal, Vol. 17, 606-613

[45] Pennington S.V., Berzins M. (1994): *New NAG Library software for first-order partial differential equations*, ACM Transactions on Mathematical Software, Vol. 20, No. 1, 63-99

[46] Peters N., Warnatz J. eds. (1982): *Numerical Methods in Laminar Flame Propagation*, Notes on Numerical Fluid Dynamics, Vol. 6, Vieweg, Braunschweig

[47] Pfeiffer B.-M., Marquardt W. (1996): *Symbolic semi-discretization of partial differential equation systems*, Mathematics and Computers in Simulation, Vol. 42, 617-628

[48] Pfleiderer J., Reiter J. (1991): *Biplicit numerical integration of partial differential equations with the transversal method of lines*, Report No. 279, DFG SPP *Anwendungsbezogene Optimierung und Steuerung*, Technical University, Dept. of Mathematics, Munich

[49] Price H., Varga R., Warren J. (1966): *Application of oscillation matrices to diffusion-convection equations*, Journal of Methematical Physics, 301-311

[50] Rektorys K. (1982): *The Method of Discretization in Time and Partial Differential Equations*, Reidel, Dordrecht

[51] Renardy M., Rogers R.C. (1993): *An Introduction to Partial Differential Equations,* Texts in Applied Mathematics, Vol. 13, Springer, Berlin

[52] Schiesser W.E., Silebi C.A. (1997): *Computational Transport Phenomena*, Cambridge University Press

[53] Schiesser W.E. (1991): *The Numerical Method of Lines,* Academic Press, New York, London

[54] Schiesser W.E. (1994): *Computational Mathematics in Engineering and Applied Science*, CRC Press, Boca Raton

[55] Schiesser W.E. (1994): *Method of lines solution of the Korteweg-de Vries equation*, Computers in Mathematics and Applications, Vol. 28, No. 10-12, 147-154

[56] Schittkowski K. (1979): *Numerical solution of a time-optimal parabolic boundary-value control problem*, Journal of Optimization Theory and Applications, Vol. 27, 271-290

[57] Schittkowski K. (1997): *Parameter estimation in one-dimensional time dependent partial differential equations*, Optimization Methods and Software, Vol. 7, No. 3-4, 165-210

[58] Schittkowski K. (2001): *EASY-FIT: A software system for data fitting in dynamic systems*, Structural and Multidisciplinary Optimization, Vol. 23, No. 2, 153-169

[59] Schittkowski K. (2002): *Numerical Data Fitting in Dynamical Systems - A Practical Introduction with Applications and Software*, Kluwer Academic Publishers

[60] Schittkowski K. (2004): *PCOMP: A modeling language for nonlinear programs with automatic differentiation*, in: *Modeling Languages in Mathematical Optimization*, J. Kallrath ed., Kluwer, Norwell, MA, 349-367

[61] Schittkowski K. (2004): *110 Mathcad worksheets for nonlinear programming*, Report, Dept. of Computer Science, University of Bayreuth, Germany

[62] Schittkowski K. (2004): *178 Mathcad worksheets for data fitting*, Report, Dept. of Computer Science, University of Bayreuth, Germany

[63] Schittkowski K. (2004): *295 Mathcad worksheets for differential equations*, Report, Dept. of Computer Science, University of Bayreuth, Germany

[64] Schittkowski K. (2004): *28 Mathcad worksheets for differential algebraic equations*, Report, Dept. of Computer Science, University of Bayreuth, Germany

[65] Schittkowski K. (2004): *17 Mathcad worksheets for partial differential algebraic equations*, Report, Dept. of Computer Science, University of Bayreuth, Germany

[66] Sheng Q., Khalic A.Q.M. (1999): *A compound adaptive approach to degenerate nonlinear quenching problems*, Numerical Methods for Partial Differential Equations, Vol. 16, No. 1, 107-132

[67] Sincovec R.F., Madsen N.K. (1975): *Software for nonlinear partial differential equations*, ACM Transactions on Mathematical Software, Vol. 1, No. 3, 232-260

[68] Strikwerda J.C. (1997): *Finite Difference Schemes and Partial Differential Equations*, Chapman and Hall, New York

[69] Troeltzsch F. (1999): *Some remarks on second order sufficient optimality conditions for nonlinear elliptic and parabolic control problems*, in: Proceedings of the Workshop 'Stabilität und Sensitivität von Optimierungs- und Steuerungsproblemen', Burg (Spreewald), Germany, 21.-23.4.99

[70] Tveito A., Winther R. (1998): *Introduction to Partial Differential Equations*, Springer, New York

[71] Ulbrich S. (1995): *Stabile Randbedingungen und implizite entropiedissipative numerische Verfahren für Anfangs-Randwertprobleme mehrdimensionaler nichtlinearer Systeme von Erhaltungsgleichungen mit Entropie*, Dissertation, TU München, Institut für Angewandte Mathematik und Statistik

[72] Vande Wouwer A., Saucec Ph., Schiesser W.E. (2001): *Adaptive Methods of Lines*, Chapman and Hall/CRC, Boca Raton

[73] Verwer J.G., Blom J.G., Furzeland R.M., Zegeling P.A. (1989): *A moving grid method for one-dimensional PDEs based on the method of lines*, in: Adaptive Methods for Partial Differential Equations, J.E. Flaherty, P.J. Paslow, M.S. Shephard, J.D. Vasilakis eds., SIAM, Philadelphia, Pa., 160-175

[74] Verwer J.G., Blom J.G., Sanz-Serna J.M. (1989): *An adaptive moving grid method for one-dimensional systems of partial differential equations*, Journal of Computational Physics, Vol. 82, 454-486

[75] Vreugdenhil C.B., Koren B. eds. (1993): *Numerical Methods for Advection-Diffusion Problems*, Vieweg, Braunschweig

[76] Walas S.M. (1991): *Modeling with Differential Equations in Chemical Engineering*, Butterworth-Heinemann, Boston

[77] Wang H., Al-Lawatia M., Sharpley R.C. (1999): *A characteristic domain decomposition and space-time local refinement method for first-order linear hyperbolic equations with interface*, Numerical Methods for Partial Differential Equations, Vol. 15, No. 1, 1-28

[78] Weizhong D., Nassar R. (1999): *A finite difference scheme for solving the heat transport equation at the microscale*, Numerical Methods for Partial Differential Equations, Vol. 15, No. 6, 697-708

[79] Wolmott P., Dewynne J.N., Howison S.D. (1993): *Option Pricing: Mathematical Models and Computation*, Oxford Financial Press

[80] Yang H.Q., Przekwas A.J. (1992): *A comparative study of advanced shock-capturing schemes applied to Burgers' equation*, Journal of Computational Physics, Vol. 102, 139-159

[81] Zachmanoglou E.C., Thoe D.W. (1986): *Introduction to Partial Differential Equations with Applications*, Dover