Benchmark Examples

The MatMOL Group (2009)

Introduction

Inside the Benchmark_Examples directory one can find, among others, the Matlab© .m-files for the examples used in Logist et al. (1). The examples to simulate the Buckley-Leverett equations for an oil well, a dispersive jacketed tubular reactor and fixed bed bioreactor. The methods employed are based on (*i*) a method of lines approach (2) and (*ii*) an operator splitting approach (3).

These example codes are available free of charge and on an as is basis. The authors cannot be held liable for any deficiency, fault or inconvenience resulting from their use.

Contents

The benchmark problems have been included in five different folders. The folders with XXX_MatMOL contains the examples using the MatMOL toolbox, while the folders with XXX_SeqMeth, contain the examples exploiting the splitting methods. Here, XXX refers to one of the benchmark cases, i.e., DFR for the dispersive tubular reactor, FBBR for the fixed bed bioreactor and Buckley_Leverett for the oil well example. In order to test the configuration, in each of the folders, files have been added which contain the function calls to all examples: (*i*) Auto_Pe_1.m, Auto_Pe_100.m, and Auto_Pe_10000.m, for the tubular reactor and fixed bed bioreactor examples, and (*ii*) Auto_eps_01.m, Auto_eps_001.m, and Auto_eps_0001.m for the oil well example.

As indicated in the paper (1), all benchmark example codes have been programmed as Matlab© functions. This approach allows the user to directly choose between several options and to readily experiment with the codes. Each of the available options is documented at the beginning of the .m-file. To maintain a clean overview over the different example codes, all required subfunctions have each time been implemented in the same file as nested functions. One exception, however, are the files BC_DynGrid.m, Flux_DynGrid.m, and dFlux_DynGrid_dx.m which for the dynamic regridding approach specify the boundary conditions, the fluxes, and their Jacobian, respectively.

In addition, one auxiliary file `AddIntStats.m` is added to enable the availability of the integration statistics. Tables 1, 2 and 3 illustrate the available files and describe their features for the three benchmark examples.

Installation

To be able to run the benchmark examples, the MatMOL toolbox has to be downloaded and installed (see http://www.matmol.org/). Also make sure that the main MatMOL directory and all subdirectories have been added to the Matlab© path. To add these directories to the path, open Matlab©, go to `File > Set Path` in order to open the `Set Path` dialog box. Choose `Add with subfolders ...` and select the main MatMOL directory (i.e., the one to which you unzipped MatMOL toolbox). Choose `Save` if you want use these settings for a future session, or `Close` if you want to use the settings only for the current session and not for future sessions.

Note on the calling of ODE/DAE solvers in Matlab©

In most of the files the Matlab© ODE/DAE integrators are called in order to return a structure which apart from the solution trajectories also contain the integration statistics (e.g., number of steps, number of function evaluations, ...). This options has been selected for illustrative reasons.

```
   sol = solver(odefun,[t0 tf],y0...)
Stats = sol.stats;
```

However, in this formulation the evolution of the independent variables is stored at each of the steps taken by the integrator in between the initial point `t0` and the final `tf` in the interval. Hence, whenever fine steps have to be taken for large ODE/DAE systems (which originate, e.g., from a PDE discretisation) the amount of information to be stored increases rapidly. When this amount of data approaches the size, or exceeds the amount of the available memory, parts of the data have to be stored in the virtual memory (i.e., the hard disk). Access to this information requires each time read and write operations which tremendously slow down the computation. Evaluating the dependent variables at certain desired points `tspan` is finally done by employing the function:

```
   Y = deval(sol,tspan);
```

By consequence, when the number of points at which the dependent variables have to be evaluated, is rather limited, it can be advantageous to call the integrator as follows:

```
   [T,Y] = solver(odefun,tspan,y0);
```

which only stores the dependent variables at the points provided in `tspan`. However, it should be noted that in this last case the integration statistics are not accessible.

Dispersive tubular reactor		
File	Integrator	Description
`TDFR_SeqMeth_Euler.m`	none	Sequencing Method with Euler scheme for reaction part
`TDFR_SeqMeth_Transition.m`	none	Sequencing Method with transition matrix formulation for reaction part
`TDFR_SeqMeth_Explicit.m`	`ode45`	Sequencing Method with explicit ODE solver for reaction part
`TDFR_SeqMeth_Implicit.m`	`ode15s`	Sequencing Method with implicit ODE solver for reaction part
`TDFR_MatMOL.m`	`ode15s`	MatMOL with low- and high order spatial approximations Explicit algebraic boundary condition formulation (→ DAE formulation)
`TDFR_MatMOL_ForcingFunction.m`	`ode45`	MatMOL with low- and high order spatial approximations Forcing function approach for boundary conditions (→ ODE formulation)
`TDFR_MatMOL_Elimination.m`	`ode45`	MatMOL with low- and high order spatial approximations Elimination of algebraic boundary conditions (→ ODE formulation)
`TDFR_MatMOL_FluxLimiter.m`	`ode15s`	MatMOL with flux limiting functions (e.g., Koren, Minmod, Mc, Smart, Superbee, Van Leer)
`TDFR_MatMOL_StaticRegridding.m`	`ode23s`	MatMOL with static regridding
`TDFR_MatMOL_DynamicRegridding.m`	`ode23s`	MatMOL with dynamic regridding

Table 1: Overview and description of the files for the dispersive tubular reactor benchmark example

Fixed bed bioreactor		
File	Integrator	Description
`TFBBR_SeqMeth_Euler.m`	none	Sequencing Method with Euler scheme for reaction part
`TFBBR_SeqMeth_Transition.m`	none	Sequencing Method with transition matrix formulation for reaction part
`TFBBR_SeqMeth_Explicit.m`	`ode45`	Sequencing Method with explicit ODE solver for reaction part
`TFBBR_SeqMeth_Implicit.m`	`ode15s`	Sequencing Method with implicit ODE solver for reaction part
`TFBBR_MatMOL.m`	`ode15s`	MatMOL with low- and high order spatial approximations Explicit algebraic boundary condition formulation (→ DAE formulation)
`TFBBR_MatMOL_ForcingFunction.m`	`ode45`	MatMOL with low- and high order spatial approximations Forcing function approach for boundary conditions (→ ODE formulation)
`TFBBR_MatMOL_Elimination.m`	`ode45`	MatMOL with low- and high order spatial approximations Elimination of algebraic boundary conditions (→ ODE formulation)
`TFBBR_MatMOL_FluxLimiter.m`	`ode15s`	MatMOL with flux limiting functions (e.g., Koren, Minmod, Mc, Smart, Superbee, Van Leer)
`TFBBR_MatMOL_StaticRegridding.m`	`ode23s`	MatMOL with static regridding
`TFBBR_MatMOL_DynamicRegridding.m`	`ode23s`	MatMOL with dynamic regridding

Table 2: Overview and description of the files for the fixed bed bioreactor benchmark example

Oil well		
File	Integrator	Description
Buckley_Leverett_MatMOL.m	ode15s	MatMOL with low- and high order spatial approximations
Buckley_Leverett_FluxLimiter.m	ode15s	MatMOL with flux limiting functions (e.g., Koren, Minmod, Mc, Smart, Superbee, Van Leer)
Buckley_Leverett_MatMOL_StaticRegridding.m	ode23s	MatMOL with static regridding
Buckley_Leverett_MatMOL_DynamicRegridding.m	ode23s	MatMOL with dynamic regridding

Table 3: Overview and description of the files for the oil well benchmark example

References

[1] F. Logist, P. Saucez, J.F. Van Impe, and A. Vande Wouwer 2009. Simulation of (bio)chemical processes with distributed parameters using Matlab. *Accepted for publication in Chemical Engineering Journal* doi:10.1016/j.cej.2009.08.017.

[2] A. Vande Wouwer, P. Saucez, and W.E. Schiesser 2004. Simulation of distributed parameter systems using a Matlab-based method of lines toolbox: Chemical engineering applications, *Industrial and Engineering Chemistry Research*, 43, 3469-3477.

[3] S. Renou, M. Perrier, D. Dochain, and S. Gendron 2003. Solution of the convection-dispersion-reaction equation by a sequencing method. *Computers and Chemical Engineering*, 27, 615-629.