Guidance on the selection of efficient computational methods for multimedia fate models

DOI:10.1021/es201928d

Guidance on the selection of efficient computational methods for multimedia fate models

Journal:	*Environmental Science & Technology*
Manuscript ID:	es-2011-01928d.R2
Manuscript Type:	Article
Date Submitted by the Author:	n/a
Complete List of Authors:	Semplice, Matteo; University of Insubria, Dep. of Science and High Technology Ghirardello, Davide; University of Insubria, Dep. of Science and High Technology Morselli, Melissa; University of Insubria, Dep. of Science and High Technology Di Guardo, Antonio; University of Insubria, Dep. of Science and High Technology

Guidance on the selection of efficient computational methods for multimedia fate models

M. Semplice,* D. Ghirardello, M. Morselli, and A. Di Guardo

Dep. of Science and High Technology

University of Insubria – Via Valleggio, 11 – Como (Italy)

E-mail: matteo.semplice@uninsubria.it

Abstract

Dynamic Multimedia fate models (MFMs) have to deal with the temporal and spatial variation of physical-chemical properties, environmental scenarios and chemical emissions. In such complex simulation tools, an analytical solution is not practically feasible and even a numerical approach requires a suitable choice of the method in order to obtain satisfying speed and reliability, particularly when certain combinations of modelling scenarios and chemical properties occur. In this paper, considering some examples of a wide range of realistic chemical and scenario properties, some sources of stiffness in MFM equations are pinpointed. Next, a comparison of the performances of several numerical schemes (chosen as representatives of three wide classes) is performed. The accuracy and the computational effort required by each method is evaluated, illustrating the general effectiveness of automatically adapted timesteps in numerical algorithms and the pros and cons of implicit timestepping. The results show that automatic error control methods can significantly improve the quality of the computed solutions and most often lead to relevant savings in computing time. Additionally, explicit and implicit methods are compared, indicating that an implicit method of medium order (around 5) is the best choice as a general purpose MFM computing engine.

*To whom correspondence should be addressed

Introduction

MFMs are nowadays standard tools to evaluate ecosystem exposure and to assess the indirect exposure of humans. Most of the initial MFMs[1,2] were based on the definition of well mixed standard compartments, such as air, soil, water and sediment, and were simple steady state partitioning models with a fixed environmental scenario.[3–6] Many MFMs then evolved towards unsteady state or dynamic systems,[7] also used to study the behaviour of persistent organic pollutants at a global level.[8,9] The number of compartments also increased as the original four were often subdivided into a number of boxes (e.g. layered soil compartments) in order to gain a more accurate description of chemical movement.[10–13] Spatial variability of emissions and environmental scenarios was later introduced and handled in a variety of ways, from site specific models, to GIS/spatially explicit approaches.[14–16] More recently, a lot of effort was devoted to incorporating the influence of environmental scenario and chemical changes in the models:[17] from seasonal changes, such as vegetation cycles, to monthly/daily/hourly[18] variations of compartment properties, meteorological conditions, and physical-chemical properties.

Among MFMs, compartmental models are the most used in environmental fate studies and are usually composed by boxes that exchange chemicals with each other and in which various processes (partitioning, transformation, etc) occur.[6] The focus of the present paper is on numerical solution techniques suitable for MFMs representing real-world systems that can be simplified to one spatial dimension (e.g. depth in soil, position along a river, etc).

In a MFM, the time evolution of the amount $Y_i(t)$ of a substance in the i^{th} compartment at time t is described by an Ordinary Differential Equation (ODE) that takes the environmental behaviour of the substance and considering all the processes mentioned above. There is an ODE per compartment and, as the equations are intertwined by the exchange/transport terms, they must be solved as a system. This system is often too large or too complex in order to allow the computation of an analytic solution and thus one resorts to numerical methods to compute an approximation. While MFMs become more and more complex, the choice of a numerical method is crucial in order to obtain a good approximation of the solution (with respect to its accuracy, absence of spurious drifts, oscillations, etc.) in a reasonable computational time.

More in details, the discrepancy between the computed result and the true value (accuracy) can be split into two parts: the modelling error, that is to say the difference of the true value of a variable from the exact solution of the ODE, and the numerical error, that is to say the difference between the exact and the computed solution of the ODE. The modelling error is due, for example, to the approximations of chemical and physical processes in the MFM, to the uncertainty in the input data, etc., and it can be reduced only by changing the model or its parameters. However, once these are fixed, the numerical error can be controlled by employing a numerical algorithm that is suitable for the task at hand. Thus, for the purpose of this paper, the quality of the solution is defined as the relative numerical error: a numerical method is better than another one if it can compute a solution with comparable numerical error, using less computing time.

Any numerical method requires the selection of a timestep Δt and computes the amounts Y_i at time $t_{n+1} = t_n + \Delta t$ from the known amounts at time t_n. An appropriate choice of Δt is crucial for the quality of the computed solution. Obviously, one has to take into account the stability requirement of the methods (see TextSI-1 in the Supporting Information), but this is not sufficient to guarantee the accuracy of the computed results: the latter is controlled by a combination of the method (especially its order) and of the choice of timesteps.

The majority of numerical methods for ODEs can be conveniently subdivided according to the following criteria:[19–21]

(a) choice of timestep length

- *fixed timestep*: it is chosen a-priori by the user
- *automatic timesteps*: an error tolerance is set by the user and the algorithm takes care of choosing Δt in order to meet the requirement

(b) time advancement, i.e. the computation of $\mathbf{Y}(t_{n+1})$ from the value $\mathbf{Y}(t_n)$ at the previous timestep

- *explicit methods*: it is achieved simply by evaluating some formula

- *implicit methods* it requires the solution of a system of equations

(3) *order*, a positive integer number (see TextSI-1).

Automatic and implicit methods are more delicate and difficult to implement than fixed timestep and explicit ones, but can have significant advantages: in (a) the reliability of the results deriving from the embedded error control mechanism and in (b) the possibility of taking longer (even though more computationally demanding) timesteps. Higher order methods are expected to give lower errors at a given timestep length, but the evaluation of a numerical method should be performed from a slightly different point of view. In fact, each step of a higher order method will cost more CPU time than a step of a low order; thus, the computational efficiency of a method is a balance of the cost of each step and the Δt needed to ensure a fixed relative numerical error.

The difficulty encountered by numerical methods in approximating the solution of an ODE system is related to the so-called *stiffness* of the ODE. In brief, an ODE is *stiff* if its solution changes abruptly in a timescale which is much shorter than the time span of interest.[21] For an ODE system describing the environmental fate of a chemical, stiffness may arise from all the processes that imply rapid and large changes of the amount of chemical, which could be related to environmental conditions (e.g. high winds, rapidly changing compartment volumes) and particularly extreme physical-chemical properties (e.g. solubility). Degradation processes are also supposed to cause stiffness, because a very degradable chemical could be lost in a few hours, requiring very short time-steps to track accurately its rapidly varying concentrations. Unless a suitable numerical method is employed, the numerical solution of a stiff system requires an unreasonable number of very short timesteps, resulting in long computational times and enhancing the risk of "pollution" of the results by floating point approximation errors.

Additionally, a careful choice and implementation of the numerical scheme becomes vital when computing resources are put under stress, like for spatially explicit situations (e.g. GIS-based approaches) or when considerable time frames must be considered in a highly dynamic situation. The aim of this paper is to provide a guidance for the choice of efficient and reliable numerical methods by enlightening the relationship between environmental/chemical facts and the performance of methods of different types, with regard to their ease of use, implementation, accuracy, robustness and efficiency.

The example developed in the main article is a MFM with one spatial dimensions, while a description of a solver for more general situation (e.g. more space dimensions) can be found in TextSI-4.

Materials and methods

Mathematical description of MFMs A rather general mathematical description of a MFM with N compartments is given by the *system of linear ODEs*

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{Y}(t) = M(t)\cdot\mathbf{Y}(t) + \mathbf{s}(t) \tag{1}$$

where $\cdot$ denotes the matrix-vector product. The column vector $\mathbf{Y}$ collects the amounts (e.g. mol) $Y_j(t)$ of chemical present in compartment j at time t. In the formula M represents an $N\times N$ matrix, whose entries may change with time (e.g. hourly, in response to environmental parameters). Each matrix entry M_{ij} is the transport rate (e.g. mol/h) from compartment j to compartment i, with diagonal entries M_{ii} representing loss terms from the i^{th} compartment (e.g. degradation). For example, in a soil compartment, there will be contributions to M due to biodegradation, advection, runoff, infiltration, etc. The column vector $\mathbf{s}$ collects source terms (e.g. emission rates).

When the boxes of the MFM are ordered so that each box exchanges substances only with two other ones (e.g. layers in air/soil, segments of a river) most M_{ij} will be zero and matrix M will take the *tridiagonal* form

$$M = \begin{bmatrix} -d_1 & u_1 & 0 & \dots & \dots 0 \\ l_2 & -d_2 & u_2 & 0 & \dots 0 \\ 0 & \ddots & \ddots & \ddots & \\ \vdots & & & & \\ 0 & \dots & 0 & l_N & -d_N \end{bmatrix}. \tag{2}$$

Numerical methods for ODEs In this paper, as prototypes of the methods of the different kinds mentioned in the Introduction, several methods of the Runge-Kutta (RK) class are tested (see TextSI-1 for background information). However, the relative performance of an implicit versus an explicit method or of a fixed versus automatic timestepping is largely maintained across the various implementations and thus the results also represent a guide for the choice of a routine in one of the publicly available libraries mentioned below.

The first method is the classic fourth-order Runge-Kutta method,[19,20] which is explicit and uses a fixed timestep that has to be chosen at the beginning of the computation (see TextSI-2). There is no straightforward way to make a good choice of the timestep length based only on the chemical and environmental characteristics for a given simulation run, but formula (SI.4) in TextSI-1 provides a safe choice computed from the coefficients of the matrix M. When this formula is used to analyse the matrix $M(t)$ and to select a timestep which is used for the whole computation, the numerical method will be called RK4. When the timestep is changed at fixed intervals (e.g. every hour, every day, every month) and estimated each time by applying the formula to this smaller time span, the method will be named RK4a.

The other two methods employ the technique of automatic timestepping. These methods vary the timestep during the computation of the solution in order, taking automatically into account the variations in the model coefficients, like those arising from variations of internal transfer and degradation processes as well as those depending on the variations of the advective fluxes that cross the boundaries of the region described by the MFM. As a representative of explicit automatic methods, the Runge-Kutta of order five DOPRI5(4),[19,20] introduced by Dormand and Prince, is used. Finally, the implicit method of order five ESDIRK5(4),[22] introduced by Kværno, was considered. These algorithms choose a trial timestep length according to formula (SI.4) at the beginning of each simulation hour and compute a final timestep together with an estimate of the approximation error committed (see formulas (SI.8) and (SI.9) in TextSI-3). The computed values are accepted only if the error is lower than a tolerance set by the user and rejected otherwise. In both cases a better guess for the timestep is derived and either the rejected step is recomputed with the new (shorter) timestep length or the values are accepted and the next timestep is computed with the new (possibly longer) timestep length. The most common way to choose the error tolerance is to set a parameter, called `RTol`, to 10^{-q}, where q is the number of leading significant digits that the user wants to be correct in the computed values. For the timestep control mechanism to be reliable, $q \geq 3$ must be used; in the following numerical tests $q = 6$ was employed, corresponding to numerical results correct within one part per million. A thorough description of the two methods is given in TextSI-3 and TextSI-4.

For a MFM with N compartments, the computational cost of a single timestep of these methods is approximatively $19N$ for RK4 and RK4a, $48N$ for DOPRI5(4) and $69N$ for ESDIRK5(4), as detailed in the Supporting Information. The overall computational time listed in TableSI-1 is of course the result of the balance between the cost of each step and the timestep length employed.

Implementation of the methods RK4 and ESDIRK5(4) are used in the most recent codes for SoilPlus[12] and AirFug,[13] but for this paper all the methods were implemented in a C++ library, in order to perform a fair comparison. Since the scenario properties in the tests are constant within each hour but change from one hour to the next, loss and transport coefficients for each air and soil box are computed by the MFM and the coefficients of the matrix M and the vector $\mathbf{s}$ are saved to disk for each simulation hour. The coefficients are then loaded from disk, the numerical solution is computed and the results of the simulations are saved, including statistics on the number of accepted, rejected and total timesteps employed, which allow the evaluation of the overall performance of the methods.

In order to evaluate the errors of the computed solutions, an *exact* solution would be needed. As this is not available analytically, following the tradition in the evaluation of numerical methods for ODEs, the approximation calculated by the *fixed timestep* RK4 using 10^6 steps per hour was regarded as being exact. This choice rules out errors coming from the automatic timestep selection. The runs for the reference solutions took 2 hours each on an Intel Xeon running at 2.80 GHz. All the other simulations and CPU times were recorded with an Intel Core2 running at 1.2 GHz and using binaries compiled with the same optimisation flags (`-O2` in `gcc`).

The choice to employ methods of the Runge-Kutta type was guided by the ease of implementation of the automatic timestepping version: sample pseudo-codes are provided in TextSI-2, TextSI-3 and TextSI-4. One should also be aware that many free libraries are available to compute numerical solutions of an ODE system, for example the rather complete collection `odepack`[23] for linear multistep methods and the `GSL` library[24] for Runge-Kutta methods. In this respect, it can be observed that these routines are tailored for nonlinear ODEs and that, in order to achieve high performances on linear equations like Eq. (1), they must be called with $M\mathbf{Y}+s$ as the function defining the ODE and specifying explicitly that M is the derivative function.

Eigenvalues and stiffness The *eigenvalues* of a $N \times N$ matrix are N scalar values associated to the matrix and they can be used to characterise the matrix.[25] Let `maxE` denote the largest absolute value of these N numbers. The study of `maxE` for the tridiagonal matrix M will show the link between the chemical/environmental properties and the behaviour of the different algorithms, thus providing a guide to the choice of a reliable and time-efficient method for computing the solution of Eq. (1).

In fact, in the presence of large values of `maxE` (which is controlled by the elements of the matrix M and thus by the environmental scenario and chemical properties of the substance) or large emission rates in $\mathbf{s}$, explicit methods are usually forced to take very short timesteps due to the very stringent stability requirement, while implicit ones are free to choose Δt basing only on `RTol` (see TextSI-1). Thus, the knowledge of the largest eigenvalue is needed in order to choose a stable timestep length, or otherwise to predict what timesteps will be chosen by an automatic timestepping procedure. On the one hand, the computation of the eigenvalues is a time consuming and delicate task if $N > 4$. On the other hand, in order to compute a stable Δt, only an approximation of `maxE` is required. The maximum row-sum of the absolute values of the matrix entries (Λ in formula (SI.5)) estimates `maxE` from above and thus it can replace `maxE` in formula (SI.4) in order to get a stable timestep length (see TextSI-1).

Environmental model and test-case scenarios The numerical methods described above were compared on a number of simulations performed using the SoilPlus model.[12] This is a site-specific, dynamic model of the fate of organic chemicals, composed by two air compartments (named upper air (UA) and lower air (LA)) and a variable number of litter and soil boxes. An illustration of the MFM is provided in FigureSI-1.

The air compartments,[13] are characterised by hourly changing heights and wind speeds. In the simulations, realistic meteorological conditions of a semi-urban area nearby Milan (Italy) for the period between May and August 2007 (2232 hours) were employed, in order to provide a reasonable range of values for such environmental parameters: UA height (10–2270 m), LA height (10–4000 m), UA wind speed (0.29–35 m/s), LA wind speed (0.26–26 m/s), rainfall (0–24 mm/day), minimum (7.5–21 °C) and maximum (13.5–35 °C) daily temperature and global solar radiation (3–29 MJ/day).

In order to distinguish between the numerical difficulties (stiffness) arising from the meteorological scenario and those arising from the physical-chemical properties of the chemicals, all simulations were performed in the scenario described above (named *dynamic-air scenario*), and in a static one (named *still-air scenario*), which is identical to the previous one, except for UA and LA heights (fixed at 500 m) and wind speed (constant at 0.1 m/s), corresponding to a rather immobile air compartment.

In this work, a 0.3 m thick loamy soil, according to the USDA classification,[26] with an organic carbon fraction in soil set to 0.02, is subdivided into 60 boxes, with two additional ones for the air compartments. This results in an ODE system of the form of Eq. (1) with 62 ODEs and coefficients, changing hourly but kept constant within each hour.

Several chemicals, characterised by different physical and chemical properties, and thus environmental behaviour, were simulated in both scenarios. The selected chemicals are not intended to be representative of the range of property variations, but rather to illustrate some specific real chemical property features that may change the model response and induce stiffness. Their physical-chemical properties are reported in Table 1. Dodine and benomyl, albeit present in ionised form,

Table 1: Properties of the simulated chemicals at 25 °C.[27–29]

Chemical		MW ($g\,mol^{-1}$)	$\log K_{ow}$	$\log K_{aw}$	$\log K_{oa}$	HL_{air} (days)	HL_{soil} (days)
PHE	phenantrene	178.2	4.57	-2.88	7.45	2	230
CHR	chrysene	228.3	5.86	-4.58	10.44	7	710
COR	coronene	300.4	6.75	-6.76	13.51	7	2300
HCB	hexachlorobenzene	284.8	5.5	-1.28	6.78	710	2300
CB	chlorobenzene	112.6	2.8	1.18	1.62	7	230
DCM	dichloromethane	89.94	1.25	-1.15	2.40	81	230
MET	metolachlor	283.8	3.13	-6.04	9.17	7	23
BEN	benomyl	290.6	2.3	-9.11	11.41	0.21	71
DOD	dodine	287.4	1.25	-9.00	10.25	7	13.25

were considered as non-ionic chemicals, characterised by the properties listed in Table 1. The chemicals can be grouped as follows:[30]

(a) scarcely intermedia mobile hydrophobic chemicals, with relatively high octanol-water partition coefficient ($\log K_{ow} > 4$) and air-water partition coefficients ($\log K_{aw}$) ranging between -6.76 and -2.88: phenanthrene (PHE), chrysene (CHR) and coronene (COR).

(b) multimedia semivolatile chemicals, with $\log K_{aw}$ between -1.28 and 1.18 and $\log K_{ow}$ between 1.25 and 5.5: hexachlorobenzene (HCB), dichloromethane (DCM), chlorobenzene (CB). These chemicals, given their range of properties, show a relative volatility.

(c) multimedia soluble chemicals, with $\log K_{aw}$ comprised between -9.00 and -6.00 and $\log K_{ow}$ in the $1.25 - -3.13$ range: metolachlor (MET), benomyl (BEN) and dodine (DOD). Such chemicals, given the range of their partition coefficients, are relatively involatile and tend to partition, to a large extent, to the water phase.

(d) half-life driven chemicals: in order to evaluate the stiffness introduced by biodegradation, MET was simulated using decreasing half-lives, progressively reducing the reference value provided in Table 1 (23 days) by one order of magnitude at a time (2.3, 0.2, 0.02 days).

The chemicals were homogeneously applied in the first 10 soil boxes (total application depth of 0.05 m) during the first hour of each simulation performed, with an application rate of 1 kg/ha.

However, series (a) chemicals, given their very scarce intermedia transfer, resulted in simulations where no rapid changes of the mass balance of the substance in the compartments were observed: this produced no appreciable differences in the performance of the numerical methods. On the other hand, when these chemicals were applied to air, significant and rapid transport processes towards the soil compartments (due to deposition) are shown; this, in turn, magnifies the different behaviour of the MFM with respect to the numerical integration. Therefore the results reported for series (a) refer to the application of chemicals in LA with a constant emission of 0.001 mol/h.

Results and discussion

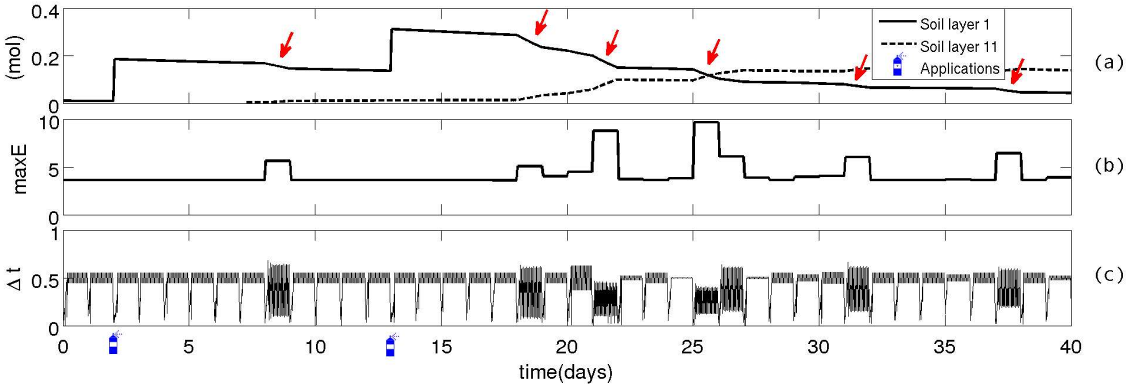

Figure 1: 40 days of a still-air MET simulation. Application days are marked on the bottom time line. Top: MET content (mol) in the first soil layer (arrows indicate rainy days). Middle: maximum eigenvalue. Bottom: timestep used by DOPRI54 (hour).

Environmental unsteadiness and automatic time stepping Figure 1 illustrates the relation between environmental events, eigenvalues and timesteps. In the simulation shown, performed in the still air scenario, an herbicide (MET) was applied twice (day 3 and 14) in the top 5 cm of soil (first 10 soil boxes in SoilPlus). For the sake of clarity, only the content of the first and the eleventh soil boxes is shown in Figure 1a. Rainy days cause a decrease in the MET amount in the first soil box due to leaching (arrows in Figure 1a). Mathematically, this shows up with `maxE` being 2-3 times higher than in dry days (Figure 1b). Correspondingly, during these events, the explicit method DOPRI5(4) had to take shorter timesteps for the whole day (Figure 1c). Analogously, shorter timesteps were used during the application hours. It is also clear that short timesteps are needed at the beginning of each simulation day, in response to the updated coefficients. Obviously, a fixed timestep algorithm would have needed short timesteps for all the simulation, increasing the computational cost.

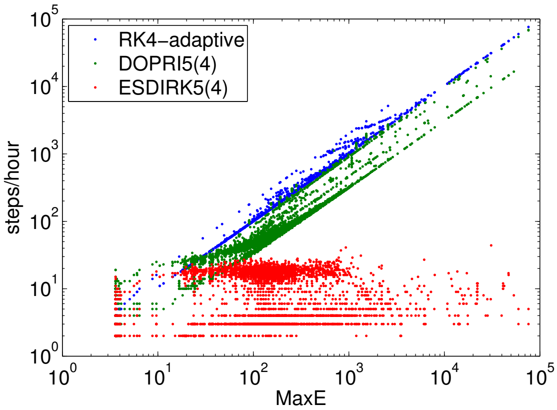

Figure 2: Correlation between `maxE` and number of steps per hour employed by RK4a and the two automatic algorithms. (Data from MET and DOD, still-air and dynamic-air scenarios)

Figure 2 is the scatter-plot of `maxE` and the number of timesteps taken by RK4a and the two automatic algorithms in the 2232 hours for four representative simulations. The linear relationship between `maxE` and the timesteps for the two explicit schemes is evident: up to 10^5 steps per hour are required, when `maxE` is of the same order of magnitude. On the other hand, ESDIRK5(4) never needed more than 100 steps per hour. This is due to the fact that ESDIRK5(4), being implicit, is not subject to the stability requirement and can take longer timesteps even when `maxE` is very high (see TextSI-1 and TextSI-4).

Eigenvalues dependence on environmental and physical-chemical properties Environmental properties may rule `maxE` in some circumstances. For example, when volatility and solubility are not extreme (e.g. unusually involatile or insoluble chemicals), `maxE` is mainly controlled by rain in the still-air scenario and by wind speed in the dynamic-air scenario. This is evident from Figures 3a–c, where the scatter plot of `maxE` in the MET simulations versus the hourly rainfall and the average of the UA and LA wind speed is shown. However, for highly volatile or soluble

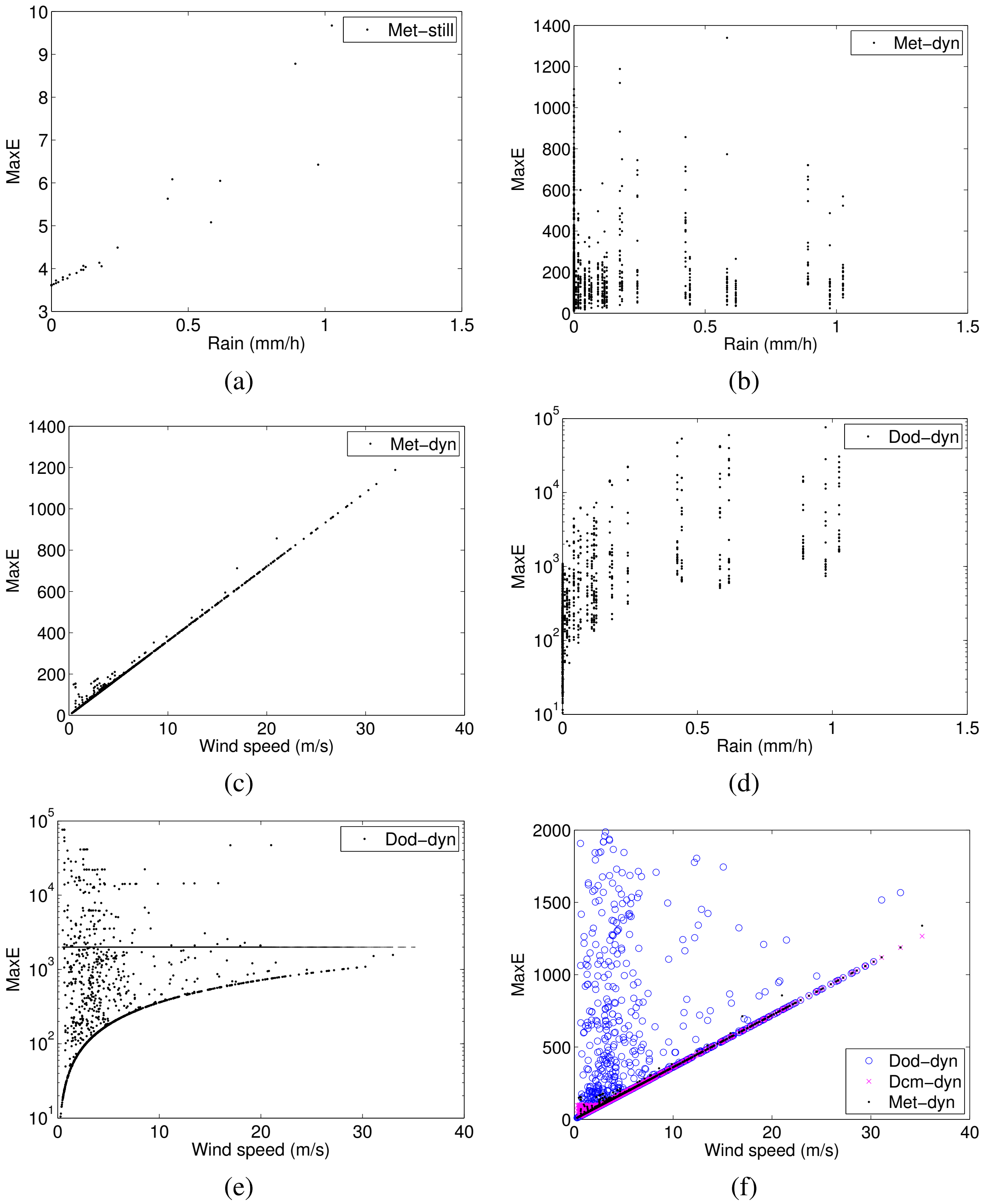

Figure 3: Correlation between rain rate, wind speed and `maxE`. a) MET in the still-air scenario. b,c) MET in the dynamic-air scenario. d,e) DOD in the dynamic-air scenario, with logarithmic scale on the vertical axis. f) A comparison of 3 chemicals (DOD values above the horizontal line in (b) are out of scale and not shown).

chemicals, this simple relationship is lost. For example, Figures 3d,e show the scatter plots of `maxE` versus the hourly rainfall and the average wind speed in the case of DOD. Figure 3d shows that increasing rainfall sharply enlarges `maxE`, even if there is not a strict correlation between `maxE` and rainfall (note the different vertical scale with respect to Figure 3b). The variability in `maxE` that is not explained by the rainfall amount could in principle be explained by the wind speed. However, Figure 3e clearly depicts that wind speed determines a minimum value for `maxE` without being able to explain all its variance. Such variability in `maxE` is not even explained by some linear combination of the two controlling variables: in fact, a multivariate linear regression fit (not shown) has an R^2 value of only 0.2, evidencing no correlation of `maxE` with rainfall and wind speed. Finally, Figure 3f shows that the lower bound on `maxE` determined by the wind speed is linear and largely independent on the molecule, whereas the upper limit of `maxE` strongly depends on the chemical under consideration.

This is a very strong indication that simulation efficiency can be greatly improved by using algorithms that are capable to automatically choose the timestep and adapt it during the simulation, since a good value of the timestep h cannot be easily predicted *a priori* by examining the properties of the chemicals and of the modelled environment.

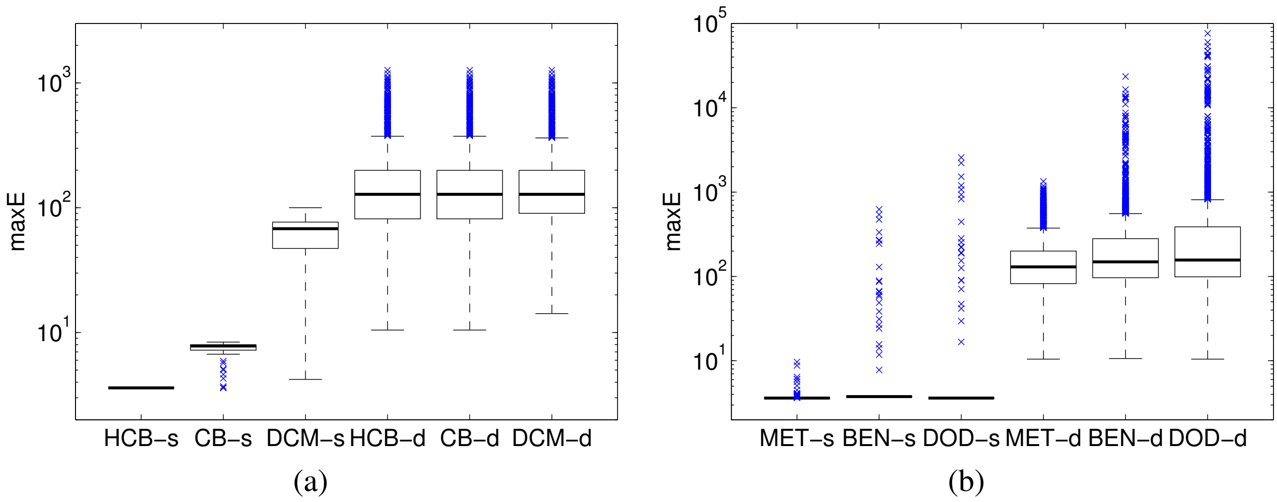

Figure 4: Box-plot of the maximum eigenvalue occurring during the 2232 hours of various simulations (whiskers indicate 3/2 of the interquartile range and the crosses represent outliers). (a): increasing volatility (b): increasing solubility. (Note the different vertical scale)

The effect of the chemical properties on `maxE` (and consequently on the number of timesteps per hour) was further examined by analysing the distribution of `maxE` during the 2232 simulation. In particular, one can observe that stiffness increases with very high or very low volatility and solubility, whereas $\log K_{ow}$ and the half-lives have a comparably smaller effect (complete results may be found in TableSI-1).

Figure 4 shows the box-plots of `maxE` occurring during each simulation. Increasing volatility in the still-air scenario causes an increase in the median of the maximum eigenvalue (4a): when $\log K_{aw}$ changes from -1.28 to 1.18, the median of `maxE` is increased by two orders of magnitude. Introducing a realistic wind greatly increases the eigenvalues, but reduces the above mentioned difference among the chemicals: this shows that wind is the dominant source of stiffness.

On the other hand, the increase of solubility (from MET to DOD) in the still-air scenario does not affect the median of `maxE`, but produces a wider distribution with very large outliers (4b), with `maxE` becoming as large as in the dynamic scenario for other series. This makes the employment of a fixed timestep method very unfeasible, whereas an automatic one is at its best.

When the dynamic-air scenario is considered, both median and outliers of the `maxE` distribution move up by a factor of 10^2, so that the spread of `maxE` in a simulation may now reach 4 orders of magnitude, with the higher values reaching 10^5. In these extreme cases, only an automatic algorithm based on an implicit method, like ESDIRK5(4), can perform the job of computing the solution efficiently.

CPU times, error and performance comparison In order to evaluate the numerical methods, their relative performance in terms of computational times and errors has to be taken into account. On the one hand, each timestep of an higher order method will require more CPU time than a step of a low order method; on the other hand, an higher order method can usually guarantee the same tolerance taking longer steps. Similarly, each timestep of an implicit method is more computationally demanding than an explicit one; this is counterbalanced by the fact that implicit methods can guarantee the same precision with longer timesteps and thus less steps are taken to complete the simulation. TableSI-1 collects the data on the performance of the algorithms on all the scenarios considered and the simulations performed.

The RK4a tests show that some degree of adaptivity in the timestep choice helps in lowering the computing time. However, setting a criterion for the choice of Δt based only on stability yields unpredictable results in terms of the relative error of the computed solution. For example, errors up to 10^{-2} in the nonstiff cases of CHR and COR in the still-air scenario were produced. Moreover, unexpected variability of the relative error (10^{-3}–10^{-7}) was observed for series (b) chemicals in the dynamic scenario. DOPRI5(4) and ESDIRK5(4) automatically choose the length of each timestep in order to satisfy a desired (set by the user) relative tolerance. `RTol` was set to 10^{-6}, which ensures that the numerical errors are at least 4–5 orders of magnitude smaller than those deriving from parameter uncertainties. All simulations reveal a relative error lower than 10^{-8} (see TableSI-2). The little computational overhead introduced by the adaption algorithm is well compensated by the reliability of the computed solution.

In evaluating the data on the number of steps, it has to be kept in mind that each simulation hour has a different matrix M and thus, no method can employ less than 2232 steps. The fastest CPU times of the implicit method reflect the lack of stability requirement: the algorithm is free to choose Δt based on `RTol` only. For example, in the case of PHE in still air, both methods use roughly 2 steps per hour to compute a numerical solution within tolerance while the explicit method is slightly faster, since its cost per timestep is lower. In dynamic air, the MFM system is much stiffer and stability forces DOPRI5(4) to take 10^5 steps, but ESDIRK5(4) can reach `RTol` using just one fifth of steps, being therefore much faster. Additionally, ESDIRK5(4) saves CPU time thanks to the very low ratio of rejected steps, since almost all of the CPU time is employed in computing accepted steps. No rejections due to negative values in the solution were observed, showing the robustness of the method. As a result, time savings can be striking: DOD in the dynamic-air scenario is simulated in 32 seconds by DOPRI5(4) and in 1.7 seconds by ESDIRK5(4) (see TableSI-1 for details).

Selection of methods for MFM applications A good method for general purpose MFM integration should yield results with predictable errors and in a reasonably short computational time, so that the modeller can concentrate on the MFM code, without worrying about the numerical part. Additionally, CPU time savings can become crucial for longer simulations or when spatially explicit models, such as those requiring a large discretization of an environmental scenario, are employed. Based on the framework shown, the following guidance can be provided

order The most convenient order for a numerical method is around 4 or 5. In fact lower order methods need too many timesteps to achieve the required accuracy and higher order ones usually cannot deploy their potential due to the non-smoothness of the solution. TableSI-2 contains a comparison with methods of order 2 and 8.

automatic timestep choice This is highly recommended, since it frees the modeller from having to worry about stability issues and the need to estimate the degree of stiffness of the MFM's equation. Given the degree of uncertainty of many input properties (physical and chemical properties, compartment composition, etc) the value suggested for `RTol` is 10^{-6}. This a reasonable choice to ensure that the errors introduced in the results by the numerical integration procedure are orders of magnitude smaller than those caused by the uncertainty in the parameters.

implicit methods They are a very convenient choice, since their extra cost is mitigated by the linearity of the ODE to be solved and they can achieve high accuracy while take comparatively long timesteps, even when rapid changes of the MFM variables occur in short times. In most of the tests, the implicit method outperformed the explicit ones. They are thus suggested for a general purpose MFM integrator. Only for low stiffness MFMs can explicit methods be considered.

implementation To the best of our knowledge, all ready-made implementations of implicit methods for ODEs assume that the ODE is non-linear. An ad-hoc implementation that takes into account the linearity of the MFM equation can lead to extra time saving.

Finally, it can be pointed out that, by examining a graph of the timestep length used by an explicit automatic algorithm during the simulation, the troublesome events for the numerical methods can be unveiled. As an example, these peculiarly short timesteps reveal when extreme changes in the value of a variable are present, which can be due to the normal variation of an environmental condition (e.g. a sudden rise or fall in temperature, high wind speeds or heavy rainfall as in Figure 1) or the wrong compilation of the environmental dataset (such as merging datasets of an environmental variable deriving by two non intercalibrated instruments).

Acknowledgement

This work was performed within the project “Modelling tools to evaluate the soil fate of agrochemicals, the mitigation potential of vegetated buffer strips and risk for aquatic organisms”, within the Italian Government funded PRIN 2008 project: “Ecotoxicological risk of plant protection products: ecological effects and mitigation measures”

Supporting Information Available

An overview of Runge-Kutta methods, pseudo-codes for the methods employed, complete data on the tests performed and a pictorial representation of the SoilPlus MFM are included in the Supporting Information. This material is available free of charge via the Internet at `http://pubs.acs.org/`.

A library with an implementation of the algorithms considered is available in source code format by contacting the corresponding author.

References

(1) Mackay, D. Finding fugacity feasible. *Environ. Sci. Technol.* **1979**, *13*, 1218–1223.

(2) Mackay, D.; Paterson, S. Calculating fugacity. *Environ. Sci. Technol.* **1981**, *5*, 1006–1014.

(3) Mackay, D.; Di Guardo, A.; Paterson, S.; Cowan, C. E. Evaluating the environmental fate of a variety of types of chemicals using the EQC model. *Environ. Toxicol. Chem.* **1996**, *15*, 1627–1637.

(4) Mackay, D.; Di Guardo, A.; Paterson, S.; Kicsi, G.; Cowan, C. E. Assessing the fate of new and existing chemicals: a five stage process. *Environ. Toxicol. Chem.* **1996**, *15*, 1618–1626.

(5) Mackay, D.; Di Guardo, A.; Paterson, S.; Kicsi, G.; Cowan, C.; Kane, D. M. Assessment of chemical fate in the environment using evaluative, regional and local-scale models: illustrative application to chlorobenzene and linear alkylbenzene sulfonates. *Environ. Toxicol. Chem.* **1996**, *15*, 1638–1648.

(6) Mackay, D. *Multimedia environmental models: the fugacity approach*, 2nd ed.; CRC Press: Boca Raton, 2001; p 261.

(7) Di Guardo, A.; Calamari, D.; Zanin, G.; Consalter, A.; Mackay, D. A fugacity model of pesticide runoff to surface water: development and validation. *Chemosphere* **1994**, *23*, 511–531.

(8) Wania, F.; Mackay, D. Global fractionation and cold condensation of low volatility organochlorine compounds in polar regions. *Ambio* **1993**, *22*, 10–18.

(9) Wania, F.; Mackay, D. The evolution of mass balance models of persistent pollutant fate in the environment. *Environ. Pollut.* **1999**, *100*, 223–240.

(10) Strand, A.; Hov, Ø. A model strategy for the simulation of chlorinated hydrocarbon distributions in the global environment. *Water Air Soil Pollut.* **1996**, *86*, 283–316.

(11) Suarez, L. A. *PRZM-3. A model for predicting pesticide and nitrogen fate in the crop root and unsaturated soil zones: user's manual for release 3.12.2*; 2006.

(12) Ghirardello, D.; Morselli, M.; Semplice, M.; Di Guardo, A. A dynamic model of the fate of

organic chemicals in a multilayered air/soil system: development and illustrative application. *Environ. Sci. & Technol.* **2010**, *44*, 9010–9017.

(13) Morselli, M.; Ghirardello, D.; Semplice, M.; Di Guardo, A. Modeling short-term variability of semivolatile organic chemicals in air at a local scale: an integrated modeling approach. *Environ. Poll.* **2011**, *159*, 1406–1412.

(14) Barra, R.; Vighi, M.; Maffioli, G.; Di Guardo, A.; Ferrario, P. Coupling SoilFug model and GIS for predicting pesticide pollution of surface water at watershed level. *Environ. Sci. Technol.* **2000**, *34*, 4425–4433.

(15) Hollander, A.; Huijbregts, M.; Ragas, A.; van de Meent, D. BasinBox: a generic multimedia fate model for predicting the fate of chemicals in river catchments. *Hydrobiologia* **2006**, *565*, 21–38.

(16) Pistocchi, A. A GIS-based approach for modeling the fate and transport of pollutants in Europe. *Environ. Sci. Technol.* **2008**, *42*, 3640–3647.

(17) Thibodeaux, L., Mackay, D., Eds. *Handbook of Chemical Mass Transport in the Environment*; CRC Press: Boca Raton, 2011.

(18) MacLeod, M.; Scheringer, M.; Podey, H.; Jones, K. C.; Hungerbühler, K. The origin and significance of short-term variability of semi-volatile contaminants in air. *Environ. Sci. Technol.* **2007**, *41*, 2827–2832.

(19) Hairer, E.; Nørsett, S. P.; Wanner, G. *Solving ordinary differential equations. I Nonstiff problems.*, 2nd ed.; Springer Series in Computational Mathematics; Springer-Verlag: Berlin, 1993; Vol. 8; p 528.

(20) Butcher, J. C. *Numerical methods for ordinary differential equations*, 2nd ed.; John Wiley & Sons Ltd.: Chichester, 2008; p 463.

(21) Lambert, J. D. *Numerical methods for ordinary differential systems. The initial value problem*; John Wiley & Sons Ltd.: Chichester, 1991; p 293.

(22) Kværnø, A. Singly diagonally implicit Runge-Kutta methods with an explicit first stage. *BIT* **2004**, *44*, 489–502.

(23) ODEPACK, Available at https://computation.llnl.gov/casc/odepack/odepack_home.html.

(24) GSL, Available at http://www.gnu.org/software/gsl.

(25) Golub, G. H.; Van Loan, C. F. *Matrix computations*, 3rd ed.; Johns Hopkins Studies in the Mathematical Sciences; Johns Hopkins University Press: Baltimore, MD, 1996; p 698.

(26) Rawls, W.; Ahuja, L.; Brakensiek, D.; Shirmohammadi, A. In *Handbook of hydrology*; Maidment, D., Ed.; 1993; pp 5.1–5.51.

(27) Mackay, D.; Shiu, W.; Ma, K. *Illustrated handbook of physical-chemical properties and environmental fate for organic chemicals*; V. Lewis Publisher, 1997; Vol. 1,2,3.

(28) Tomlin, *The pesticide manual*, 11th ed.; British Crop Protection Council: Farnham (UK), 1997.

(29) Footprint Database, Available at http://sitem.herts.ac.uk/aeru/iupac/index.htm.

(30) Gouin, T.; Mackay, D.; Webster, E.; Wania, F. Screening chemicals for persistence in the environment. *Environ. Sci. Technol.* **2000**, *34*, 881–884.

Graphical TOC Entry