Nationaal Lucht- en Ruimtevaartlaboratorium
National Aerospace Laboratory NLR

NLR-TP-2002-400

Control-fluid interaction in air-conditioned aircraft cabins

A demonstration of stability analysis for partitioned dynamical systems

J.M.A. Hofman

Nationaal Lucht- en Ruimtevaartlaboratorium
National Aerospace Laboratory NLR

NLR-TP-2002-400

Control-fluid interaction in air-conditioned aircraft cabins
A demonstration of stability analysis for partitioned dynamical systems

J.M.A. Hofman

This investigation has been carried out partly under a contract awarded by the EU, contract number G4RD-CT99-0056, and partly as a part of NLR's basic research programme, Workplan number I.1.A.1/I.1.A.3.

This report has been submitted for publication in Comput. Methods Appl. Mech. Engrg.

Customer:	National Aerospace Laboratory NLR
Working Plan number:	I.1.A.1/I.1.A.3
Owner:	National Aerospace Laboratory NLR
Division:	Information and Communication Technology
Distribution:	Unlimited
Classification title:	Unclassified
	August 2002

Summary

The numerical simulation of complicated dynamical systems is often realized by the coupling of existing subsystem models. In this partitioned treatment the interaction between the subsystems is effectuated through an algorithm for transmission and synchronization of coupled system variables. This coupling algorithm should be chosen carefully, as it affects the numerical stability and accuracy of the simulation results.

This report demonstrates how simplified models can be used to analyse the time stepping stability and accuracy of two common coupling algorithms: the staggered scheme, and the Jacobi scheme. This so-called a priori analysis is particularly useful when direct analysis of the coupled simulation models is prohibited by the complexity of (one of) these models, or in cases where only executable code is available. One of the primary ingredients of a priori analysis is the search for mathematical models that are simple, on the one hand, while still representing the basic physics, on the other hand.

A priori analysis has been applied to a partitioned simulation model named HEATPI. This simulation model is composed of a computational fluid dynamics tool that computes the time-dependent temperature at the location of a sensor in an aircraft cabin, and a simple temperature controller modelled in MATLAB Simulink. On basis of the sensor temperature the controller model computes the temperature at the air inlet of the aircraft cabin. By considering the conservation of heat a simple analytical model has been derived to determine the average cabin temperature as a function of time, for a given time dependent inlet temperature. This analytical model is called Thermal Cabin Model (TCM). The staggered scheme and the Jacobi scheme have been analysed for the coupling of the TCM and the controller model. This analysis is shown to be suitable to predict the time step sizes that are required for accuracy and stability of HEATPI.

In particular it is found that the two coupling algorithms have similar accuracies when the time steps are chosen sufficiently small. The Jacobi scheme yields the possibility to run both software models on different processors in a parallel fashion. This property can be exploited to increase the computational efficiency in cases where the models have similar computation times.

List of abbreviations

ACS	Air-conditioning system
CFD	Computational fluid dynamics
ETCM	Discrete thermal cabin model defined by Eq. (23)
HEAT	CFD program as used in this work to compute the time-dependent temperature field in the passenger cabin of a small commuter aircraft
HEATPI	Coupled simulation tool consisting of HEAT and the temperature controller which is modelled in MATLAB Simulink
JOR	Iterative scheme using the Jacobi method and relaxation (cf. Ref. 6)
SOR	Iterative scheme using the Gauss-Seidel method and relaxation (cf. Ref. 6)
TCM	Thermal cabin model defined by Eq. (12)

List of symbols

Symbol	Description	Unit
$A_S,\ A_J$	Iteration matrices for staggered scheme and Jacobi scheme	
$A_{SE},\ A_{JE}$	Iteration matrices defined by Eqs. (24) and (25)	
$\mathbf{b}$	Constant vector	
B	Iteration matrix	
$\mathbf{c}$	Constant vector	
$c_1,\ c_2$	Coefficients defined by Eq. (27)	°C
D	Subset of integers; $D = \{0, 1, 2, ..., T/\Delta t\}$	
$f,\ g,\ h$	Functions	
I	Integration part of controller model, as defined by Eq. (16)	°C
K_i	Integral action coefficient in controller model defined by Eqs. (15) and (16)	s^{-1}
K_p	Proportional action coefficient in controller model Eq. (15)	
m_a	Mass of air contained in the aircraft cabin	kg
p	Integer denoting the order of convergence	
r	Global discretization error of cabin temperature as defined by Eq. (35)	°C
t	Time	s
δt	Time step size in subsystem model	s
Δt	Time step size used in coupling scheme	s
T	Upper limit of simulation time interval; $T/\Delta t$ is a positive integer	s
T_c	Cabin temperature at sensor location, or average cabin temperature	°C
T_in	Temperature at air-conditioning inlet	°C
T_out	Temperature at air-conditioning outlet	°C
T_0	Inlet temperature at $t = 0$	°C
T_e	Inlet temperature at $t = \infty$	°C
T_r	Reference temperature in the controller model (15)	°C
u	Generic input variable	
x	Generic state variable	
y	Generic output variable	
$\mathbf{z}$	Vector of certain system variables	
ϵ	$\epsilon = \Delta t/\tau$	
ϕ_m	Mass flow rate of the air entering the aircraft cabin through the inlet opening	kg/s
$\lambda_+,\ \lambda_-$	Parameters defined by Eq. (27)	s^{-1}
ρ	Spectral radius of iteration matrix	
τ	Characteristic time of thermal cabin model; $\tau = m_\mathrm{a}/\phi_m$	s

Subscript/superscript	Description
co	Controller
cr	Upper limit of stability domain
f	Flow solver
i	Index designating a particular subsystem
$m,\ n$	Non-negative integers denoting discrete time stations
P	Predicted value
0	Initial value
¯	The exact solution of non-discrete equations is designated by $\bar{y}$ if y denotes the solution of the discretized equations.

Contents

(33 pages in total)

This page is intentionally left blank.

1 Introduction

Analysis of dynamical systems and multiphysics problems will often require the use of computer simulation models and numerical integration techniques. In many applications, computational feasibility and affordability is reached by breaking down the model of the entire physical system into models for several subsystems. In this partitioned approach the behaviour of the entire physical system is solved by advancing the solution for the separate subsystems in time. Interaction between the subsystems is effectuated through a numerical algorithm for transmission and synchronisation of coupled system variables. In the present report this algorithm will be referred to as the *coupling algorithm*.

The use of partitioned system models may have several advantages. For instance, individual software models can be easily exchanged and maintained. However, due to certain limitations, partitioning is not always preferable. For example, it is often found computationally inefficient to use partitioning for systems that involve interaction effects throughout a volume as is the case for electromagnetic fields. An overview of the use of partitioned analysis of coupled dynamical systems has recently been given in a tutorial article by Felippa *et al.* (Ref. 2).

It is well known that the partitioned approach requires a careful formulation of the coupling algorithm to avoid serious degradation in time stepping stability and accuracy (Refs. 2, 3, 4, 10). In many applications these aspects cannot be assessed analytically, e.g. when the subsystem models are very complex, or in situations where only executable code is available. In such cases it is worthwhile to search for analytical models that are simple, on the one hand, while still representing the basic physics, on the other hand. On basis of these analytical models the stability and accuracy of various coupling algorithms can then be analysed. Thus, basic physical insight can be used a priori to determine suitable coupling algorithms. This idea, that will be called *a priori analysis*, has proven very successful in the computer simulation of viscous-inviscid interaction (Ref. 9) and fluid-structure interaction (Refs. 3, 4).

A priori analysis can be applied in various other partitioned system applications, as will be demonstrated in this report for the case of an aircraft cabin that is coupled to a temperature controller. The aircraft cabin is modelled by a computational fluid dynamics (CFD) program, while the temperature controller is modelled in MATLAB Simulink. The time steps used in these simulation models are chosen to be very small with respect to the coupling time steps. The coupled simulation model, named HEATPI, can be used in the design and validation of the air-conditioning system (ACS) on board of aircraft. During the design process the time-dependent response of the cabin temperature is studied for different controller parameters. In practice, the controller parameters are chosen so that the cabin temperature approaches a desired value while satisfying certain comfort requirements concerning, e.g. the rate of temperature change. This problem has been treated in a partitioned fashion in order to efficiently combine the expertise in CFD and control problems. In this way simulations with HEATPI can be performed for various controller parameters without having to be experienced with the use of the CFD program.

In this report two common coupling algorithms will be applied to HEATPI: the staggered scheme and the Jacobi scheme. These algorithms are well known, e.g., in the realm of fluid-structure interaction (Refs. 2, 12). The numerical stability and accuracy of these coupling algorithms will be assessed, as applied to HEATPI. This assessment focuses in particular on the maximum size of the coupling time step for which the system is solved in a stable and sufficiently accurate way. It is interesting to note that the coupling time step is not only a parameter of the numerical simulation model, but also a design parameter for modern ACS systems. These systems are frequently equipped with digital temperature controllers using signals from temperature sensors that are sampled at a finite rate. The transfer of sampled data between different subsystems is similar to a particular coupling scheme as considered in this report. Therefore, the results in this report concerning the accuracy and stability limitations to the coupling time step may also be relevant to the design of ACS systems.

Concerning the time step sizes that can be used in HEATPI it has been observed that the restrictions as imposed by accuracy are found to be far greater than the stability limitations. Other systems exist for which the coupling schemes used in this report would only be absolutely stable for time steps that are unacceptably small for computational efficiency reasons. This applies especially for systems that are governed by stiff differential equations. In these cases it may be tried to solve the problem by combining the models into one model for the whole dynamical system. This method may call for intensive implementation effort. Moreover it is limited to situations where the subsystem models are explicitly known and where reusability of subsystem models is not essential. In cases where a partitioned procedure is preferred to solve the system dynamics it is worthwhile to consider one of the following methods

- Application of higher order accurate coupling schemes in order to extend the stability domain. Higher order schemes may be attained by finding appropriate predictor methods (cf. Ref. 2).
- Interfield iteration at each time station (cf. Section 2.3).
- Relaxation methods can be used to increase the stability domain. Examples are SOR and JOR (cf. Ref. 6).
- Semi-inverse method (cf. Ref. 1).

- Quasi-simultaneous treatment (cf. Ref. 8).

Each of these methods has its particular pros and cons which should be weighed against each other for the specific application to the dynamical system under consideration. For instance, higher order coupling schemes will generally require additional information on the subsystem models, e.g. the time derivatives of coupled system variables. Such information may be derived from simple, analytical models provided that these models are sufficiently accurate. Interfield iteration can be used to remove stability constraints. However, in many cases it is found that the computational cost can be reduced considerably by using predictor methods with smaller time steps.

The semi-inverse method is used for the coupling of two models that use the same input variable and the same output variable. This is different from HEATPI, where the input variable of the flow solver is an output variable of the controller model, and the other way round. Nevertheless, the semi-inverse method can be used in HEATPI, because the controller model can be inverted easily. The quasi-simultaneous treatment has proven to be successful in solving certain viscous-inviscid interaction problems. In this method two subsystem models are coupled through a coupling algorithm that is based on a simple model for one of the subsystems. The simple model should meet certain requirements that are given in Ref. 8. As in the semi-inverse method, the quasi-simultaneous treatment requires that one of the models is given in the ‘inverse’ form. In cases where simplified models can be derived for both subsystems, the above mentioned coupling schemes can be compared quantitatively on basis of a priori analysis.

This report is organised as follows. In Section 2 the problem of partitioned simulation of dynamical systems is explained by considering two generic dynamical models that have mutual interactions. A few common algorithms are presented to couple these models. Section 2 ends with a brief discussion on stability of the coupling algorithms. Simulation models used in HEATPI are shortly described in Section 3. In addition, simplified models are presented to analyse the time stepping stability of HEATPI for the two coupling algorithms mentioned above. As expounded in Section 4 this analysis leads to predictions of the time step sizes that are required for accuracy and stability. These predictions are verified in Section 5 by analysing the results computed by HEATPI.

2 The basic problem of model coupling

2.1 Coupling of subsystems

Consider a dynamical system that can be decomposed into two subsystems, F_1 and F_2. These subsystems are described by the mathematical models $\bar{M}_1$ and $\bar{M}_2$, respectively, where each model is defined by an operator that maps a given time-dependent input signal $\bar{u}_i(t)$ to an output signal, $\bar{y}_i(t)$ for $i = 1, 2$. Thus, the mathematical models can be denoted by

$$\bar{M}_i : \; \bar{u}_i(t) \to \bar{y}_i(t), \quad \text{for} \quad i = 1, 2. \tag{1}$$

The models are coupled through the identities

$$\begin{aligned} \bar{u}_1(t) &= \bar{y}_2(t), \\ \bar{u}_2(t) &= \bar{y}_1(t). \end{aligned} \tag{2}$$

The input and output signals are assumed to be scalar functions, for convenience. Vector signals can be treated analogously.

Following the identities (2) there is a two-way interaction between the models, as depicted in Figure 1. Two basic questions arise:

- Does the system described by Eqs. (1) and (2) have unique solutions for the input and output signals, given the initial conditions $\bar{u}_1(0)$, $\bar{u}_2(0)$.
- If there exist unique solutions, how can these be determined.

One can try to answer the above questions on the basis of specific information concerning both models, for instance the analytical prescriptions of the operators $\bar{M}_1$ and $\bar{M}_2$.

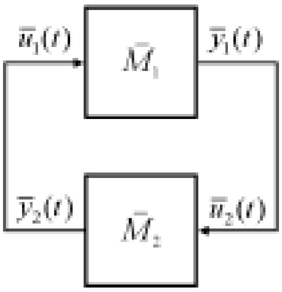

Fig. 1 Two-way interaction between $\bar{M}_1$ and $\bar{M}_2$.

If these descriptions are available it is recognized that the input and output signals satisfy implicit equations due to the mutual interactions between the models. In practice the operators' prescriptions are often too complicated to solve the implicit equations analytically, for instance if the operators are differential operators.

2.2 Simulation models

The present paper focuses on the computer simulation of two coupled subsystems as shown in Figure 1. It is assumed that software models M_1 and M_2 are available that represent numerical approximations of the mathematical models given by (1).

The numerical model M_i has a discrete input signal denoted by u_i^n, for $i = 1, 2$. Here $n = 0, 1, 2, ...$ is the discrete index of time $t^n = n\Delta t$, with constant time step Δt. Using this input signal the numerical model M_i computes its discrete output signal y_i^n which is a numerical approximation of the output signal $\bar{y}_i(t^n)$. The numerical models considered in this article use the following stationary one-step scheme:

$$\begin{aligned} x_i^{n+1} &= f_i(x_i^n, u_i^{n+1}), \\ y_i^{n+1} &= g_i(x_i^{n+1}, u_i^{n+1}). \end{aligned} \tag{3}$$

Here x_i^n is the discrete *state* variable of the model M_i at time t^n. Moreover, f_i and g_i are functions of which the function prescriptions are not necessarily known. The expression in (3) is chosen because it covers a wide class of models. For instance, the model M_i can be a computer programme to compute the temperature field x_i^{n+1}, at time t^{n+1}, in a certain three-dimensional geometry. This computation will usually be initialized by prescribing the temperature field x_i^n, at time t^n. The input u_i^{n+1} of the model may be the temperature prescribed at a part of the boundary of the geometry, while the output y_i^{n+1} can be the temperature evaluated at a specific location. In this example the function g_i in (3) is just a simple operation to extract the local temperature value from a given temperature field. The function f_i represents the part of the computer code that advances the temperature field from t^n to t^{n+1}. In practice, it will be found impossible to determine the function prescription of f_i, even when the source code of the computer program would be available.

2.3 Coupling of simulation models

There are various ways to couple numerical models of the form (3). In the present article two methods will be used. In order to explain these methods the state variables x_i^n can be neglected, so that the numerical models can be written as

$$M_i: \; y_i^{n+1} = h_i(u_i^{n+1}), \text{for } i = 1, 2. \tag{4}$$

Here the function prescriptions for h_1 and h_2 may be unknown.

The models can be coupled by choosing the following discrete counterpart of Eq. (2):

$$\begin{aligned} u_1^{n+1} &= y_2^{n+1}, \\ u_2^{n+1} &= y_1^{n+1}. \end{aligned} \tag{5}$$

Using these identities the input variables u_i^{n+1} in Eq. (4) can be eliminated, which yields

$$\begin{aligned} y_1^{n+1} &= h_1(y_2^{n+1}), \\ y_2^{n+1} &= h_2(y_1^{n+1}). \end{aligned} \tag{6}$$

It is assumed that Eqs. (5) and (6) have unique solutions for u_1^{n+1}, u_2^{n+1}, y_1^{n+1} and y_2^{n+1}, when sufficient initial conditions are prescribed, e.g. u_1^0, u_2^0. Solving the equations can be problematic, for two reasons. Firstly because the equations are implicit, and secondly because the function prescriptions of h_1 and h_2 can be too complicated or even unknown. This problem may be tackled by solving the equations in an iterative way, for instance by writing

$$\begin{aligned} y_1^{n+1,k+1} &= h_1(y_2^{n+1,k}), \\ y_2^{n+1,k+1} &= h_2(y_1^{n+1,k}). \end{aligned} \tag{7}$$

In the above equations the integer $k = 0, 1, 2, ...$ is increased until the variables $y_1^{n+1,k+1}$ and $y_2^{n+1,k+1}$ have converged within a certain tolerance, that is, *if* the process converges. Because the iterations in (7) should be performed for all time stations t^n this method is often found to be inefficient in most applications of (6). Alternatively, the index n can be used as an iteration index so that the processes of time-stepping and iteration are intertwined:

$$\begin{aligned} y_1^{n+1} &= h_1(y_2^{n+1,\mathrm{P}}), \\ y_2^{n+1} &= h_2(y_1^{n+1}), \end{aligned} \tag{8}$$

where $y_2^{n+1,\mathrm{P}}$ is a predictor for y_2^{n+1}. A few common choices for the predictor are $y_2^{n+1,\mathrm{P}} = y_2^n$ and $y_2^{n+1,\mathrm{P}} = y_2^n + \Delta t\, \dot{y}_2^n$. The latter choice requires the derivative $\dot{y}_2^n$ and there are several ways of finding an estimate value for it based on finite differences, cf. Ref. 6.

In this report two simple predictor methods will be considered. The first one uses the predictor $y_2^{n+1,\mathrm{P}} = y_2^n$. This method will be referred to as the *staggered* scheme. The second one, referred to as the Jacobi scheme, uses predictors for both variables, namely $y_1^{n+1,\mathrm{P}} = y_1^n$ and $y_2^{n+1,\mathrm{P}} = y_2^n$. The Jacobi scheme is given by

$$\begin{aligned} y_1^{n+1} &= h_1(y_2^n), \\ y_2^{n+1} &= h_2(y_1^n), \end{aligned} \tag{9}$$

By considering the staggered method (8), for fixed n, it is noticed that y_1^{n+1} is first computed by M_1 and this result is then substituted as an input value for M_2. Next, M_2 calculates the output value y_2^{n+1}. Thus, the models M_1 and M_2 are operated in a certain order. It is interesting to notice that computations have no specific order when using the Jacobi scheme (9). This property of the Jacobi scheme can be useful if two coupled solvers are required to compute in a parallel fashion for efficiency reasons.

2.4 Stability notions

Accuracy and well-posedness are important criteria for choosing a certain coupling method. In many applications the numerical results will be sufficiently accurate for sufficiently small time steps. Well-posedness, i.e. stability, may also impose a restriction to the size of the time step. When the maximum time step for stability is smaller than the time step that is required on basis of accuracy, the numerical method is found to be less efficient.

It is often difficult to analyse stability and accuracy for coupled systems of complicated simulation models. Therefore it is worthwhile to perform the analysis on basis of simplified models that represent the basic physics. These simplified models may be ordinary differential equations, which are then discretized and linearized. Application of the same coupling method as used to couple the original complicated simulation models yields a set of linear equations for the coupled system of simplified models. For stationary one-step schemes these linear equations can be compactly written as

$$\mathbf{z}^{n+1} = A\mathbf{z}^n + \mathbf{b}. \tag{10}$$

In this equation $\mathbf{z}^n$, for $n = 0, 1, 2, ...$, is the vector of variables that are stepped in time. For instance, $\mathbf{z}^n = (y_1^n, y_2^n)^\mathsf{T}$ for the Jacobi scheme given by (9). The matrix A is called the iteration matrix. It will be assumed that A is non-deficient. In physics and (numerical) mathematics the concept of stability strongly depends on the specific context. Generally speaking, the notion of numerical stability indicates the well-posedness of the time integration scheme with respect to the disturbances, as monitored by the chosen variable. The time integration scheme considered in this report is the coupling algorithm that is used to couple models of the form (3). The disturbances are variations in initial conditions, and the chosen variable is the numerical solution of the coupling algorithm in a conveniently chosen norm. In this report two stability concepts will be used, as defined below.

Zero-stability is considered when the numerical solution of the time integration scheme should approach the exact solution of the mathematical model(s) on a finite interval in time $[0, T]$ by choosing $\Delta t = T/n$ and taking the limit $n \to \infty$.

The process given by (10) is *zero-stable* if the spectral radius ρ of the iteration matrix satisfies $\rho \leq 1 + O(\Delta t)$.

Following the theorem of Lax zero stability of a time-integration scheme is equivalent with convergence if the discretization is consistent (cf. Ref. 5).

Absolute stability is relevant to solving the stationary problem by letting $t^n = n\Delta t \to \infty$. In this case the time step Δt is fixed while $n \to \infty$. Iterative methods of the form (10) are *absolutely stable* if the spectral radius ρ of the iteration matrix A satisfies $\rho < 1$.

Zero-stability and absolute stability will generally yield a restriction on the choice of the time step Δt. From the above stability definitions it follows that absolute stability is a stronger requirement than zero-stability, so that a coupling algorithm that is absolutely stable for a certain value of the time step, is also zero-stable.

3 Simulation models for temperature control of aircraft cabins

The two coupling algorithms introduced in the previous section have been applied to the partitioned simulation model HEATPI. The subsystems that are coupled in HEATPI will be shortly described in this section. Moreover, simplified models will be presented in order to analyse the numerical stability and accuracy of HEATPI.

3.1 Simulation system HEATPI

HEATPI is a numerical simulation system to study the temperature control of aircraft cabins. This system integrates a MATLAB Simulink model of the temperature controller and a CFD program to compute the fields of velocity and temperature of the air in an aircraft cabin configuration. The CFD program solves the time-dependent incompressible Navier-Stokes equations, including heat transfer using the Boussinesq approximation (cf. Ref. 11). For reasons of computational efficiency it is assumed that the velocities in the direction of the length of the cabin can be neglected, so that the cabin configuration can be modelled in two dimensions. Simulations using a three-dimensional cabin model indicate that this assumption is reasonable. Moreover, it is assumed that the air flow in the cabin is symmetric about the vertical plane that bisects the cabin geometry in the length direction.

On basis of the above assumptions a two-dimensional model has been developed for the right half of a small commuter aircraft cabin, as depicted in Figure 2. This model includes an inlet opening above the stowage bins where the velocity and the temperature of the incoming air is prescribed, and an outlet opening with a boundary condition for the pressure. Two seats including seated passengers have been modelled in two dimensions by considering conservation of volume and conservation of surface in different planes of cross-section. The resulting model is shown in Figure 2. The cabin geometry and the fluid region are supplied with a computational grid that has a carefully chosen refinement near the inlet opening (the grid for the fluid region is not shown in Figure 2). One of the grid cells serves as a sensor that measures the local air temperature (see Figure 2). This sensor temperature is used as a time-dependent input signal for the temperature controller model. The controller model will be described in Section 3.3.

3.2 Thermal cabin model

A simple dynamical model for the average temperature in the aircraft cabin can be obtained by using the conservation of heat energy. This yields

$$m_{\mathrm{a}} \frac{dT_{\mathrm{c}}(t)}{dt} = \phi_m T_{\mathrm{in}}(t) - \phi_m T_{\mathrm{out}}(t), \tag{11}$$

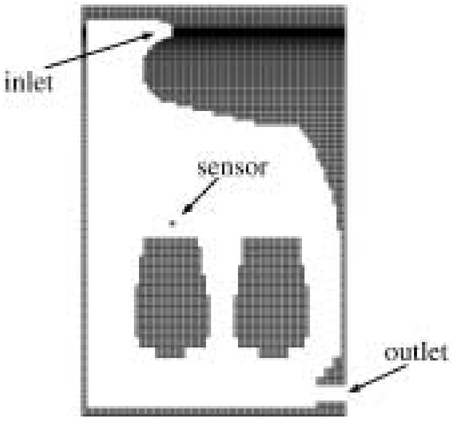

Fig. 2 Two-dimensional cabin geometry used in HEATPI, including two occupied seats, air-conditioning inlet and outlet openings, and stowage bins. The cabin temperature is measured at the indicated sensor location.

where m_{a} and T_{c} are the mass and the average temperature of the air inside the cabin, respectively. Moreover ϕ_m is the mass flux; T_{in} and T_{out} are the temperature of the air at the inlet and outlet, respectively. The above equation expresses that the rate of change of the heat stored in the cabin air is given by the difference between the heat flux coming in and the heat flux going out.

In order to be able to compute $T_{\mathrm{c}}(t)$ for given $T_{\mathrm{in}}(t)$ it is assumed that $T_{\mathrm{out}}(t) = T_{\mathrm{c}}(t)$. This assumption yields a good approximation when the rate of change of the inlet temperature is relatively small, so that temperature differences throughout the cabin volume remain small. Using this assumption Eq. (11) can be rewritten as

$$\tau \frac{dT_{\mathrm{c}}(t)}{dt} + T_{\mathrm{c}}(t) - T_{\mathrm{in}}(t) = 0, \tag{12}$$

where $\tau = m_{\mathrm{a}}/\phi_m$. The simple model given by (12) will be referred to as the TCM (Thermal Cabin Model).

In order to judge the validity of the TCM as a substitute model for the flow solver, both models have been used to compute the temperature $T_{\mathrm{c}}(t)$ for a time-dependent inlet temperature given by

$$T_{\mathrm{in}}(t) = \begin{cases} T_0, & \text{for } t < 0; \\ T_{\mathrm{e}}, & \text{for } t \geq 0. \end{cases} \tag{13}$$

The initial cabin temperature is set to $18\,^\circ\mathrm{C}$, independent of the position in the cabin geometry. Moreover, $T_0 = 18\,^\circ\mathrm{C}$, $T_{\mathrm{e}} = 23\,^\circ\mathrm{C}$, $m_{\mathrm{a}} = 56$ kg and $\phi_m = 0.79$ kg/s. In Figure 3 the time-dependent sensor temperature computed by the CFD program (solid line) is compared to the average cabin temperature as determined by the TCM (dashed line). The TCM yields an exponen-

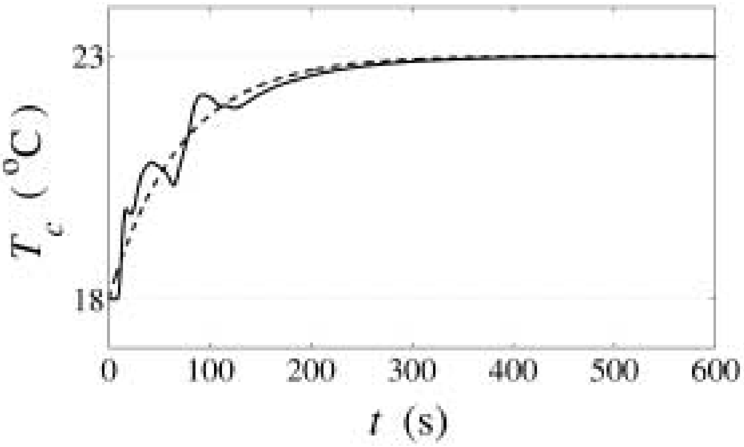

Fig. 3 $T_{\rm c}$ *versus time as computed by the CFD program (solid line) and the TCM (dashed line).*

tial solution for $T_{\rm c}(t)$ with a time constant given by $\tau = 71$ s. Obviously, the TCM does not take into account that the temperature depends on the position in the cabin. This position dependence produces fluctuations of the sensor temperature computed by the CFD program, as can be seen in Figure 3. However, considering the behaviour of the cabin temperature on time scales of the order of τ, the results computed by the TCM agree reasonably with those found by the CFD program.

In order to couple the TCM with the controller model, the TCM introduced in the previous section will be discretized. Discretization of the TCM can be done in several ways, for instance by assuming that its input signal $T_{\rm in}(t) = T_{\rm in}^{n+1}$, for $t^n < t \leq t^{n+1}$ and constant value $T_{\rm in}^{n+1}$. Defining $T_{\rm c}^n = T_{\rm c}(t^n)$ it follows from Eq. (12) that

$$T_{\rm c}^{n+1} = (T_{\rm c}^n - T_{\rm in}^{n+1})e^{-\Delta t/\tau} + T_{\rm in}^{n+1}. \qquad (14)$$

Using this expression, which will be referred to as the *discrete TCM*, the ‘new’ value of the output variable, $T_{\rm c}^{n+1}$, can be determined from the old value, $T_{\rm c}^n$, and the new input value, $T_{\rm in}^{n+1}$. It is noticed that the discrete TCM is an instance of the discrete model expression (3) for the special case of $x_i^n = y_i^n$.

3.3 Controller model

The controller is used to control the cabin inlet temperature $T_{\rm in}(t)$, using the cabin temperature $T_{\rm c}(t)$ as an input signal. The time-continuous controller model is given by

$$T_{\rm in}(t) = K_{\rm p}(T_{\rm r} - T_{\rm c}(t)) + I(t), \qquad (15)$$

with the shorthand notation

$$I(t) = K_{\rm i}\int_0^t (T_{\rm r} - T_{\rm c}(t'))dt'. \qquad (16)$$

A discrete model for the controller can be easily derived from (15) by assuming that $T_{\mathrm{c}}(t) = T_{\mathrm{c}}^{n+1}$, for $t^n < t \leq t^{n+1}$ and constant value T_{c}^{n+1}. Defining $I^n = I(t^n)$ the following discrete controller model is obtained:

$$\begin{aligned} I^{n+1} &= I^n + K_{\mathrm{i}}\Delta t(T_{\mathrm{r}} - T_{\mathrm{c}}^{n+1}), \\ T_{\mathrm{in}}^{n+1} &= I^{n+1} + K_{\mathrm{p}}(T_{\mathrm{r}} - T_{\mathrm{c}}^{n+1}). \end{aligned} \tag{17}$$

It is seen that this model is an instance of (3) where $x_i^n = I^n$, $u_i^n = T_{\mathrm{c}}^n$, and $y_i^n = T_{\mathrm{in}}^n$. In the previous section two coupling schemes have been introduced. These schemes will be applied to the coupling of the discrete models for the controller and the TCM.

4 Analysis of coupling methods using simple models

4.1 Staggered scheme

The staggered method is given by Eq. (8), using the predictor $y_2^{n+1,\mathrm{P}} = y_2^n$. This method can be applied to the controller and the TCM, by replacing the 'new' input value T_{c}^{n+1} in (17) by the 'old' value T_{c}^n. Then, the time-evolution of T_{in}^n and T_{c}^n, for $n = 0, 1, 2, ...$, can be obtained by the linear equations (14) and (17), for given initial conditions T_{in}^0 and T_{c}^0. After some manipulations the time-evolution of the vector $\mathbf{z}^n = (I^n, T_{\text{c}}^n)^{\mathsf{T}}$ can be written as

$$\mathbf{z}^{n+1} = A_S \mathbf{z}^n + \mathbf{b}, \tag{18}$$

where the iteration matrix is

$$A_S = \begin{pmatrix} 1 & -K_{\text{i}}\Delta t \\ \left[1 - e^{-\Delta t/\tau}\right] & \left[e^{-\Delta t/\tau} + (K_{\text{p}} + K_{\text{i}}\Delta t)\left(e^{-\Delta t/\tau} - 1\right)\right] \end{pmatrix}. \tag{19}$$

The constant terms in the set of linear equations are contained in the vector $\mathbf{b} = \left(T_{\text{r}}K_{\text{i}}\Delta t,\ T_{\text{r}}(1 - e^{-\Delta t/\tau})(K_{\text{p}} + K_{\text{i}}\Delta t)\right)^{\mathsf{T}}$. The time-evolution of T_{in}^n is computed afterwards by

$$T_{\text{in}}^{n+1} = I^n + (K_{\text{i}}\Delta t + K_{\text{p}})(T_{\text{r}} - T_{\text{c}}^n). \tag{20}$$

Another staggered scheme can be obtained by using the predicted value T_{in}^n in (14) while retaining T_{c}^{n+1} in (17). In that case the TCM and the PI controller exchange their input-output information in the reverse order, as compared to the previous staggered scheme. This order will generally affect the stability and the accuracy of the staggered scheme. However, for the models considered here it can be shown that the two staggered schemes are absolutely stable for the same values of the time-step Δt. Moreover, for a given time step size, the time-accuracies of the two staggered schemes are similar. Stability and accuracy of the staggered scheme given by Eqs. (18) and (19) will be discussed in Sections 4.3 and 4.4, respectively. For the other staggered scheme stability and accuracy can be assessed analogously.

4.2 Jacobi scheme

The Jacobi scheme is defined by Eq. (9). As in the staggered scheme above, the Jacobi scheme can be applied to the controller and the TCM, by replacing T_{c}^{n+1} in (17) by T_{c}^n. In addition T_{in}^{n+1} in (14) is replaced by the predicted value T_{in}^n. Thus, predictors are being used both for the input of the controller and the input of the TCM. Defining $\mathbf{z}^n = (I^n, T_{\text{in}}^n, T_{\text{c}}^n)^{\mathsf{T}}$ the time-evolution of the coupled system is given by

$$\mathbf{z}^{n+1} = A_J \mathbf{z}^n + \mathbf{c}, \tag{21}$$

where

$$A_J = \begin{pmatrix} 1 & 0 & -K_{\mathrm{i}}\Delta t \\ 1 & 0 & -(K_{\mathrm{p}} + K_{\mathrm{i}}\Delta t) \\ 0 & \left(1 - e^{-\Delta t/\tau}\right) & e^{-\Delta t/\tau} \end{pmatrix} \tag{22}$$

and $\mathbf{c} = \left(T_{\mathrm{r}}K_{\mathrm{i}}\Delta t,\ T_{\mathrm{r}}K_{\mathrm{p}},\ T_{\mathrm{r}}(K_{\mathrm{p}} + K_{\mathrm{i}}\Delta t)\right)^{\mathsf{T}}$.

The iteration matrices in (19) and (22) have been determined on basis of the discrete TCM model given by (14). This model has been derived from the exact solution of the differential equation (12). Finding the exact solution would have been difficult in cases where the system equations have more complexity. In such cases it is convenient to discretize the system equations directly. This is illustrated by discretization of the differential equation for the TCM as given in (12). For instance, backward Euler discretization may yield the expression

$$T_{\mathrm{c}}^{n+1} = \tfrac{1}{1+\epsilon}T_{\mathrm{c}}^{n} + \tfrac{\epsilon}{1+\epsilon}T_{\mathrm{in}}^{n+1}. \tag{23}$$

where $\epsilon = \Delta t/\tau$. The discrete TCM model (23) will be called ETCM. This model can be coupled with the discrete controller model (17), for instance by using the staggered method described in the previous section. In that case the time-evolution of $\mathbf{z}^n = (I^n, T_{\mathrm{c}}^n)^{\mathsf{T}}$ is as in (18), but with a different iteration matrix given by

$$A_{SE} = \begin{pmatrix} 1 & -K_{\mathrm{i}}\tau\epsilon \\ \frac{\epsilon}{1+\epsilon} & \frac{1-\epsilon(K_{\mathrm{p}}+K_{\mathrm{i}}\tau\epsilon)}{1+\epsilon} \end{pmatrix}. \tag{24}$$

Euler's method has an accuracy of $O(\Delta t)$, so that $A_{SE} = A_S + O(\Delta t/\tau)$. Thus, if the discrete TCM model is replaced by the ETCM model it is expected that the behaviour of the coupled system of discrete models is almost unchanged, at least for sufficiently small time steps.

If the discrete model ETCM is coupled with the controller according to Jacobi's method, the following iteration matrix is found:

$$A_{JE} = \begin{pmatrix} 1 & 0 & -K_{\mathrm{i}}\tau\epsilon \\ 1 & 0 & -(K_{\mathrm{p}} + K_{\mathrm{i}}\tau\epsilon) \\ 0 & \frac{\epsilon}{1+\epsilon} & \frac{1}{1+\epsilon} \end{pmatrix}. \tag{25}$$

It can be straightforwardly verified that $A_{JE} = A_J + O(\Delta t/\tau)$ where A_J is given by (22).

4.3 Stability of the coupling schemes

Discrete models for the controller and the TCM have been introduced in the previous sections. Two methods have been shown to couple these discrete models. In the present section the stability of these methods will be examined. The following system parameters will be used:

$$\begin{aligned}
&m_{\mathrm{a}} = 56\,\mathrm{kg},\\
&\phi_m = 0.79\,\mathrm{kg/s},\\
&K_{\mathrm{p}} = 0.8,\\
&K_{\mathrm{i}} = 0.05\,\mathrm{s}^{-1},\\
&T_{\mathrm{r}} = 23\,{}^\circ\mathrm{C},\\
&T_{\mathrm{c}}(0) = 18\,{}^\circ\mathrm{C},\\
&T_{\mathrm{in}}(0) = T_{\mathrm{r}} - (1-K_{\mathrm{p}})[T_{\mathrm{r}} - T_{\mathrm{c}}(0)].
\end{aligned}$$

The stability of a particular coupling scheme is determined by its iteration matrix. It is seen from Eqs. (19) and (22) that the iteration matrices for the staggered scheme and the Jacobi scheme can be expressed solely in terms of three dimensionless quantities, namely K_{p}, $K_{\mathrm{i}}\Delta t$ and $\phi_m\Delta t/m_{\mathrm{a}}$. Before examining the stability of the coupling schemes it is important to know the behaviour of the solution of the coupled models of the TCM and the controller, as given by Eqs. (12) and (15). These equations have a unique solution for the cabin temperature that can be determined analytically. This analytical solution, which will be denoted by $\bar{T}_c(t)$, is given by

$$\bar{T}_c(t) = c_1 e^{\lambda_+ t} + c_2 e^{\lambda_- t} + T_{\mathrm{r}}, \tag{26}$$

where

$$\begin{aligned}
\lambda_\pm &= \frac{-(1+K_{\mathrm{p}}) \pm i\sqrt{4\tau K_{\mathrm{i}} - (1+K_{\mathrm{p}})^2}}{2\tau},\\
c_1 &= \left(\frac{\lambda_- + 1/\tau}{\lambda_+ - \lambda_-}\right)\bigl(T_{\mathrm{r}} - T_{\mathrm{c}}(0)\bigr),\\
c_2 &= T_{\mathrm{c}}(0) - T_{\mathrm{r}} - c_1.
\end{aligned} \tag{27}$$

Since $\mathrm{Re}(\tau\lambda_\pm) < 0$ the exponentials in (26) are decaying, so that the coupled system of the controller and the TCM is *physically* stable. Therefore, it is reasonable to require that any numerical method to solve the coupled system should be *absolutely* stable (cf. Ref. 10). When using the staggered scheme given by (19) absolute stability is guaranteed when the spectral radius of the matrix A_S in (19) is smaller than unity. This condition is satisfied for time steps $\Delta t \lesssim 63$ s as can be seen in Figure 4, where the curve labelled by *S* gives the spectral radius of A_S as a function of Δt.

For Jacobi's method absolute stability conditions follow by determining the spectral radius of the

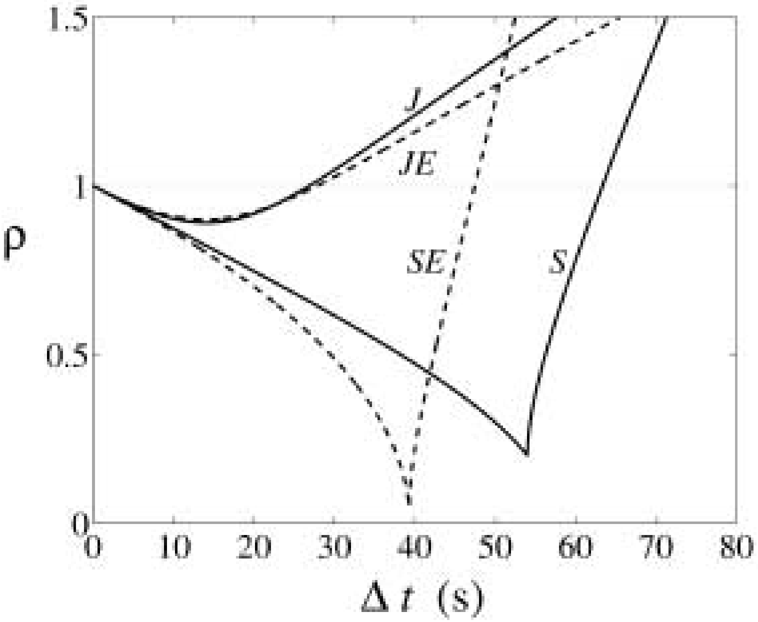

Fig. 4 Spectral radius ρ versus time step Δt for the iteration matrices A_S (S), A_J (J), A_{SE} (SE), and A_{JE} (JE).

iteration matrix A_J in (22). The result is indicated by J in Figure 4. It is found that the Jacobi scheme is absolutely stable for $\Delta t \lesssim 27$ s.

The curves labelled by SE and JE give the spectral radius for the matrices A_{SE} (staggered) and A_{JE} (Jacobi), respectively. As mentioned in the previous section $A_{SE} = A_S + O(\Delta t/\tau)$ and $A_{JE} = A_J + O(\Delta t/\tau)$. For the system parameters used here the time scale $\tau = 71$ s, which is of the same order of magnitude as the stability limit $\Delta t = 63$ as found for A_S. This explains the significant difference in the stability domains determined for A_S and A_{SE}.

It is recalled that A_S is derived from the exact solution of the differential equation given in (12). This exact solution is discretized by setting $T_{\text{in}}(t) = T_{\text{in}}^{n+1}$, for $t^n < t \leq t^{n+1}$ and a constant value T_{in}^{n+1}. The same discretization will be used if the flow solver is coupled to the controller model. Therefore, it is expected that the stability domain $[0\text{ s}, 63\text{ s}]$ is a reasonable estimate for the stability domain that would be found when the staggered scheme is used to couple the flow solver and the controller model. This expectation will be verified in Section 5.

4.4 Accuracy of the coupling schemes

The staggered scheme (18) is used to determine a numerical approximation $\mathbf{z}$ of the exact solution $\bar{\mathbf{z}} = (I, T_{\text{c}})^{\mathsf{T}}$ of a set of ordinary differential equations (ODE). This ODE is given by

$$\frac{d\bar{\mathbf{z}}}{dt} + A\bar{\mathbf{z}} = \mathbf{0}, \tag{28}$$

where

$$A = \begin{pmatrix} 0 & K_\mathrm{i} \\ -1/\tau & (1+K_\mathrm{p})/\tau \end{pmatrix}. \tag{29}$$

The equation (28) can be derived from Eqs. (12) and (15) by setting $T_\mathrm{r} = 0\,^\circ\mathrm{C}$. The latter choice renders the ODE homogeneous. In the accuracy analysis considered here inhomogeneous terms can be neglected, for convenience. In this case the staggered scheme is

$$\mathbf{z}^{n+1} = A_S \mathbf{z}^n, \tag{30}$$

where the iteration matrix A_S is defined by (19).

The accuracy of the staggered scheme can be assessed by using the so-called *modified ODE* which is given by

$$\frac{d\mathbf{z}}{dt} + B\mathbf{z} = \mathbf{0}. \tag{31}$$

The matrix B is chosen such that $\mathbf{z}(t^n) = \mathbf{z}^n$, where $\mathbf{z}^n$ is computed by using the staggered scheme (30). It follows from (31) that

$$\mathbf{z}(t^n) = e^{-Bn\Delta t}\mathbf{z}_0, \tag{32}$$

for a given initial value $\mathbf{z}_0$. Combination of (30) with (32) yields

$$B = \frac{-\log(A_S)}{\Delta t}. \tag{33}$$

The staggered scheme (30) is globally $p-$ order time-accurate if the difference between the modified ODE and the original ODE in (28) is of $O(\Delta t^p)$, cf. Ref. 7.

By expanding the matrix B as

$$B = A_0 + A_1\Delta t + O(\Delta t^2), \tag{34}$$

where A_0 and A_1 are independent of Δt, it is straightforward to show that $A_0 = A$. This means that the staggered scheme is consistent. It follows by direct inference of Lax's theorem that the numerical solution $\mathbf{z}^n$ as determined by the staggered scheme will converge to the exact solution $\bar{\mathbf{z}}(t^n)$ of (28) if Δt approaches zero. In addition it can be easily derived that $A_1 \neq 0$, for any set of system parameters K_p, K_i and τ, except for the trivial case where K_p and K_i are both zero. This observation implies that the staggered scheme (30) is globally first order time-accurate. In a similar fashion it can be shown that the Jacobi scheme in Eq. (21) is globally first order time-accurate also.

The modified equation method is useful to determine the order of time-accuracy, as has been shown above. In the present case the accuracy of the cabin temperature T_c^n can be assessed directly by computing the global discretization error r for a given set of system parameters. The global discretization error is defined by

$$r = \max_{n\in D}\bigl(T_\mathrm{c}^n - \bar{T}_c(t^n)\bigr), \tag{35}$$

where $\bar{T}_c(t^n)$ is the exact solution given by (26), evaluated at time station $t^n = n\Delta t$. Moreover $n \in D$ where D is a set of integers defined by $D = \{0, 1, 2, ..., T/\Delta t\}$. Here T is the maximum time value, and $T/\Delta t$ is a positive integer. In Figure 5 the global discretization error is plotted versus the time step Δt for the staggered scheme (squares) and the Jacobi scheme (circles). The system parameters are as given in Section 4.3. For both numerical schemes the results in Figure 5 show that $r = O(\Delta t)$, as expected. The maximum time is $T = 1000$ s. This particular choice of T does not affect the global error r as it appears that the difference $\bigl(T_\mathrm{c}^n - \bar{T}_c(t^n)\bigr)$ reaches a maximum for a time value $t^n \approx$ 100 s. It is interesting to note that the solutions T_c^n and $\bar{T}_c(t^n)$ also have maximum values for approximately the same moment in time.

If the time step increases past the value where absolute stability is lost, the global error diverges. As was found above this happens for $\Delta t > 63$ s for the staggered scheme, and for $\Delta t > 27$ s for the Jacobi scheme. When a global error of $r = 0.1\,^\circ$C is tolerated, time steps of $\Delta t = 2$ s are found to be sufficiently small. For smaller time steps there is no significant difference between the accuracies of the two numerical schemes.

When the numerical schemes are used in HEATPI to couple the flow solver with the controller for time steps $\Delta t = 2$ s, it is important to realize that the results can be less accurate then what is indicated by the global error given in Figure 5. This is due to the different behaviour of the TCM and the flow solver for time scales of the order of 1 s (see Figure 3). In order to compute this behaviour sufficiently accurate it is relevant to consider the chosen predictor value which introduces an error that is proportional to the magnitude of the derivative dT_c/dt (cf. Section 2.3). It can be derived from the results in Figure 3 that the data computed by the flow solver yield a magnitude of dT_c/dt that is about four times greater than the derivative determined by the TCM at certain moments in time. Thus, for the system parameters used here, an accuracy of HEATPI of $\pm 0.1\,^\circ$C requires a time step that is at least four times smaller than that for the coupled system of the controller and the TCM. This yields a maximum time step $\Delta t \approx 0.5$ s for simulations with HEATPI.

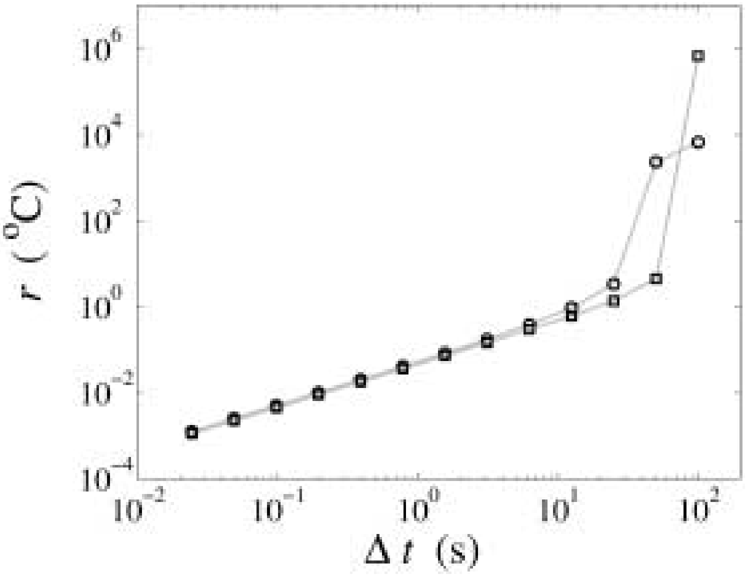

Fig. 5 Global error r plotted against time step size Δt for the staggered scheme (squares) and the Jacobi scheme (circles).

5 Application of coupling methods to HEATPI

In the previous sections simple models have been used in order to determine whether certain numerical schemes can be used in HEATPI to couple the CFD program with the controller. Important criteria for the suitability of these schemes are stability and accuracy. In this section simulation results found by HEATPI will be analysed in order to verify if the simple models are useful to predict the accuracy and the maximal time step Δt for which the coupling algorithm is absolutely stable.

Absolute stability of HEATPI is assessed by simulation of the time-dependent behaviour of the cabin temperature T_{c} as measured at the location of the sensor (Figure 2). During the simulation the flow solver computes the values of T_{c} at discrete times $t_{\text{f}}^{m} = m\,\delta t_{\text{f}}$, with a time step size δt_{f} and $m = 0, 1, 2, ...$ This time step is chosen to be much smaller than the coupling time step: $\delta t_{\text{f}} \ll \Delta t$. Analogously, the time step δt_{co} as used by the controller model to compute the inlet temperature T_{in}, satisfies $\delta t_{\text{co}} \ll \Delta t$. In HEATPI the flow solver is coupled to the controller model by transfer of the values T_{c}^{n} and T_{in}^{n}, for $n = 0, 1, 2, ...$ according to a certain coupling algorithm as explained in Section 4. Here T_{c}^{n} and T_{in}^{n} are computed at the coupling times $t^{n} = n\Delta t$.

The temperature T_{c} has been calculated for the same system parameters as used in the analysis of the coupling of the TCM and the controller. These parameters are listed in Section 4.3. The results computed for the staggered scheme are plotted in Figures 6 and 7, for coupling time steps $\Delta t \in \{1\text{ s}, 50\text{ s}, 57\text{ s}, 60\text{ s}\}$. The lines give T_{c} as computed at time stations t_{f}^{m}, and the circles indicate the values T_{c}^{n}. From the results in Figures 6 and 7 it is seen that the signal T_{c} starts to oscillate severely as the time step Δt increases past a certain critical value of the time step, denoted by Δt_{cr}. For $\Delta t < \Delta t_{\text{cr}}$ the signal converges to the reference temperature $T_{\text{r}} = 23\,^{\circ}\text{C}$ as the time increases. This implies that the coupling algorithm is absolutely stable for these values of the time step (cf. the definition of absolute stability in Section 2.4). The critical time step, defining the upper limit of the stability domain, is $\Delta t_{\text{cr}} = 58$ s with an accuracy of ± 2 s. This agrees very well with the value that has been derived for the coupling of the TCM and the controller, namely $\Delta t_{\text{cr}} = 63$ s.

In a similar fashion the absolute stability has been assessed for the case where the Jacobi scheme is used in HEATPI. In this case Figure 8 gives the results for T_{c} as a function of time, for time steps $\Delta t = 1$ s (solid line), $\Delta t = 15$ s (dotted line) and $\Delta t = 20$ s (dashed line). These results show that the critical time step is $\Delta t_{\text{cr}} = 15 \pm 5$ s. The critical time step as found by analysis of the simplified models is $\Delta t_{\text{cr}} = 27$ s which appears to be reasonable prediction.

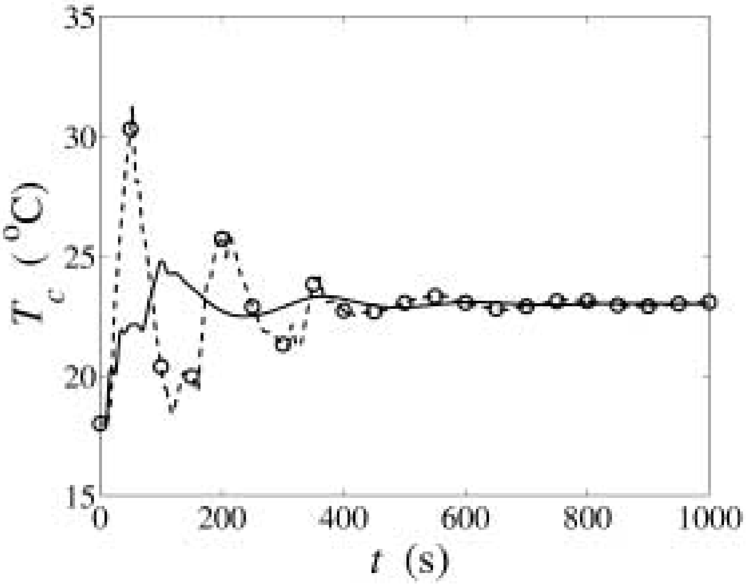

Fig. 6 Time dependent behaviour of the temperature T_c as computed by HEATPI using the staggered scheme for different time step sizes: $\Delta t = 1$ s (solid line) and $\Delta t = 50$ s (dashed line). Circles indicate the discrete values T_c^n for $\Delta t = 50$ s.

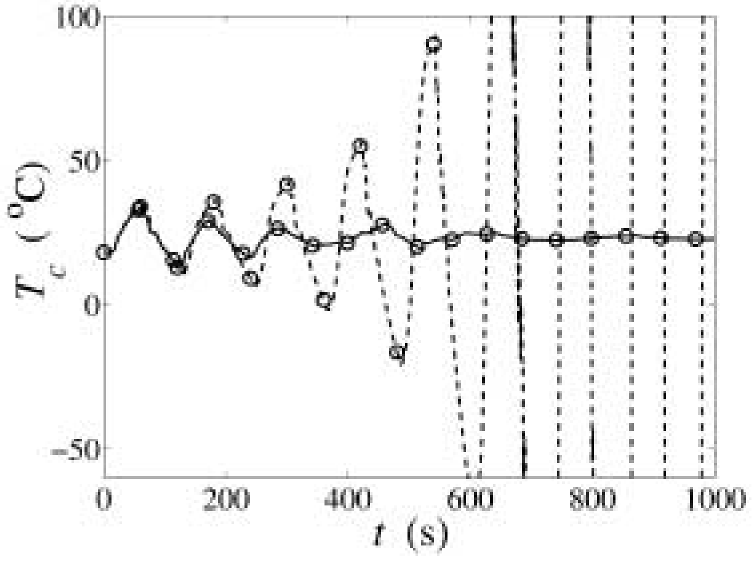

Fig. 7 T_c versus time as calculated by HEATPI using the staggered scheme for different time step sizes: $\Delta t = 57$ s (solid line) and $\Delta t = 60$ s (dashed line). Circles indicate the discrete values T_c^n.

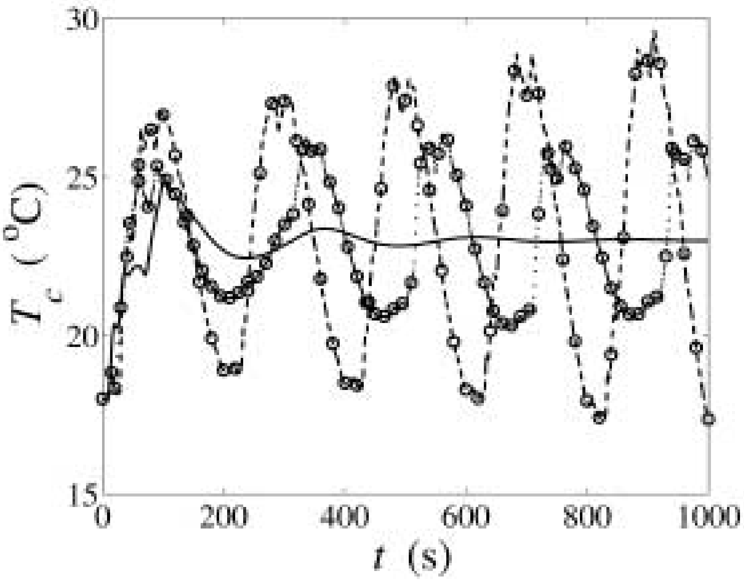

Fig. 8 T_c *plotted against time, as computed by HEATPI using the Jacobi scheme for different time step sizes:* $\Delta t = 1$ *s (solid line),* $\Delta t = 15$ *s (dotted line) and* $\Delta t = 20$ *s (dashed line). Circles indicate the discrete values* T_c^n*.*

Besides the stability of HEATPI, the accuracy has been studied. It is evident that for sufficiently accurate results the time step should be well below the critical value Δt_{cr}. For the example problem considered here the values of Δt_{cr} as found for both coupling schemes are far greater than the time steps that are required for accuracy. A simple comparison of the TCM and the flow solver in Section 4 has indicated that HEATPI is able to compute T_{c} with an accuracy of $\pm 0.1\,^\circ\mathrm{C}$ for time steps that satisfy $\Delta t \lesssim 0.5$ s. This can be verified by examining the numerical results of the cabin temperature, as computed by HEATPI for times $t \in [0\text{ s}, 1000\text{ s}]$ and for time steps Δt, $\Delta t/2$, $\Delta t/4$, etcetera. The accuracies found for the staggered scheme and the Jacobi scheme are of $O(\Delta t)$, as expected. For a value of the time step below $\Delta t = 0.2$ s the result has converged within a margin of $\pm 0.1\,^\circ\mathrm{C}$, for both coupling schemes. On basis of this result the predicted value of $\Delta t = 0.5$ s appears to be remarkably good.

6 Concluding remarks

This report demonstrates how simplified models can be used to analyse the time stepping stability and accuracy of two common algorithms for the coupling of two dynamical simulation models. This 'a priori analysis' is particularly useful when direct analysis of the coupled simulation models is prohibited by the complexity of (one of) these models, or in cases where only executable code is available. One of the primary ingredients of a priori analysis is the search for mathematical models that are simple, on the one hand, while still representing the basic physics, on the other hand.

For the example system HEATPI considered in this report, a priori analysis shows to be useful to predict the stability domain for both coupling algorithms. The accuracy of HEATPI could also be predicted very well, on basis of two observations:

- The accuracy that has been determined analytically for the coupled system consisting of the TCM and the controller model.
- A quantitative comparison of the time-dependent response of the flow solver and the TCM for a sudden change in the input signal $T_{\text{in}}(t)$. This comparison concerns in particular the magnitude of the derivative dT_{c}/dt.

Both observations have been made a priori, i.e. without the performance of computer simulations with HEATPI.

For HEATPI it has been found that the staggered scheme and the Jacobi scheme have similar accuracies for the same time step size, provided that the time step is chosen sufficiently small. For both schemes the CFD program and the controller model may be executed on different computers that are communicating through a network. The Jacobi scheme yields the possibility to run both software models in parallel. This is particularly useful in cases where the models have similar computation times.

It is finally noted that the analysis given in this report can be extended to systems of more than two partitions, and for simulation models that have multiple input and output signals. In these cases it is possible to use combinations of the coupling algorithms mentioned in this report. The effort engaged with the stability analysis of various coupling schemes for multiple coupled systems can be reduced by focusing on systems that have strong interactions.

Acknowledgements

The author would like to acknowledge Prof.Dr. A.E.P Veldman and Ir. G.A. Kremer of the Rijksuniversiteit Groningen for numerous technical discussions on the present work. The research has partly been funded by the European Commission (EC) through the ASICA project in the Growth programme. This report does not represent the view of the EC, and the author is solely responsible for the contents of this report.

7 References

1. J.E. Carter, Viscous-inviscid interaction analysis of transonic turbulent separated flow, AIAA paper No. 81-1241, in: Proceedings of the 14th AIAA Fluid and Plasma Dynamics Conference (1981).
2. C.A. Felippa, K.C. Park, and C. Farhat. Partitioned analysis of coupled mechanical systems, *Comput. Methods Appl. Mech. Engrg.* **190**, pp. 3247 - 3270 (2001).
3. S. Piperno, C. Farhat, and B. Larrouturou, Partitioned procedures for the transient solution of coupled aeroelastic problems - Part I: model problem, theory, and two-dimensional application, *Comput. Methods Appl. Mech. Engrg.* **124**, pp. 97 - 112 (1995).
4. S. Piperno and C. Farhat, Partitioned procedures for the transient solution of coupled aeroelastic problems - Part II: energy transfer analysis and three-dimensional applications, *Comput. Methods Appl. Mech. Engrg.* **190**, pp. 3147 - 3170 (2001).
5. R.D. Richtmeyer and K.W. Morton, *Difference methods for initial value problems*, John Wiley (1967).
6. J. Stoer and R. Bulirsch, *Introduction to numerical analysis*, Springer, New York (1980).
7. A.M. Stuart and A.R. Humphries, *Dynamic systems and numerical analysis*, Cambridge university press, Cambridge (1996).
8. A.E.P. Veldman, New, quasi-simultaneous method to calculate interacting boundary layers, *AIAA J.* **19**, pp. 79 - 85 (1981).
9. A.E.P. Veldman, A numerical view on strong viscous-inviscid interaction, in: *Viscous flow calculation methods*, W.G. Habashi (ed.), Pineridge Press, pp 343 - 363 (1984).
10. A.E.P. Veldman, Viscous-inviscid interaction, partitioned dynamical systems and i(n)te(g)ration, Technical publication TP 89232, National Aerospace Laboratory NLR, Amsterdam (1989).
11. P. Wesseling, *Principles of computational fluid dynamics*, Springer, Berlin (2001).
12. S. van Zuijlen, H. Bijl and S. Hulshoff, Accuracy and efficiency of partitioning algorithms in fluid-structure interaction computations, in: H.A. Mang, F.G. Rammerstorfer, and J. Eberhardtsteiner (eds.), *Proceedings of WCCM V*, Vienna, Austria (2002).