



UNICADO: multidisciplinary analysis in conceptual aircraft design

Maurice Zimmnu¹ · Florian Schültke¹ · Eike Stumpf¹

Received: 31 March 2022 / Revised: 16 September 2022 / Accepted: 12 October 2022
© The Author(s) 2022

Abstract

In this paper, different iterative methods, so-called architectures, within the multidisciplinary analysis in conceptual aircraft design of the UNICADO software are elaborated, applied and analyzed. Possible execution sequences in the sequential method (Gauss-Seidel architecture) are derived via a graph-based algorithm in combination with expert knowledge. Sensitivities of the design disciplines are analyzed and a permitted residual for stable convergence characteristic for the aircraft design with UNICADO is derived. Prerequisites for the application of a parallel iterative method, the Jacobi architecture are conducted. Runtime and convergence characteristics of the Gauss-Seidel architecture and the Jacobi architecture are evaluated. A damping method is applied to the Jacobi architecture to enhance the convergence characteristics. The Gauss-Seidel and Jacobi architectures are used to design two different aircraft, the CSR and the CSMR, with an iteration accuracy of $2.5\text{e}-3$. For these use cases studied, the Gauss-Seidel architecture converges more stably and faster than the Jacobi architecture and is, therefore, the more favorable. The aircraft design with the implemented Jacobi architecture oscillates and does not converge. Only with an implemented damping method, convergence is achieved. If the iteration time of the design loop increases, e.g., when using higher fidelity methods for aircraft design, the choice of architecture must be re-evaluated.

Keywords Aircraft design · Multidisciplinary analysis · MDA · UNICADO · MICADO

Abbreviations

CeRAS	Central reference aircraft data system
CSMR	CeRAS short-medium range
CSR	CeRAS short range
MDA	Multidisciplinary analysis
MDO	Multidisciplinary design optimization
MICADO	Multidisciplinary integrated conceptual aircraft design and optimization
RCE	Remote component environment
UNICADO	University conceptual aircraft design and optimization
VISTOMS	Visualization tool for large MDO systems
XDSM	Extended design structure matrix
ϵ	Residual
\mathbf{y}	Vector of objective variables

CoG	Center of gravity
Fuel _{max}	Maximum fuel mass
h	Step size
m_{fuel}	Mission fuel mass
MTOM	Maximum take-off mass
MZFM	Maximum zero fuel mass
OME	Operating mass empty
Payload _{max}	Maximum payload mass
psn	Process step number

1 Introduction

In multidisciplinary analysis and optimization (MDAO) within the UNICADO software [1], past research was mainly focused on the optimization part. The optimization part uses each multidisciplinary analysis (MDA) as a sample point for deriving a potential optimum for a given objective. In our case, the MDA is a whole conceptual aircraft design process with UNICADO for a given variation in specified design variables. As the aircraft design process is expensive in terms of computation time, it is also propagated in the cost of the optimization process. Therefore, the aim is to minimize the computation time of the aircraft design process, not with a reduction of the

✉ Maurice Zimmnu
maurice.zimmnu@ilr.rwth-aachen.de
Florian Schültke
florian.schuelcke@ilr.rwth-aachen.de
Eike Stumpf
eike.stumpf@ilr.rwth-aachen.de

¹ Institute of Aerospace Systems (ILR), RWTH Aachen University, Wuellnerstr. 7, 52062 Aachen, Germany

computational accuracy, but with an enhanced MDA process. In this paper, we investigate different execution sequences of aircraft design disciplines of UNICADO and the parallelization of all design disciplines to derive an optimal execution sequence and type in terms of a minimum computation time.

This document is structured as follows: In Sect. 2 the fundamentals of MDA are outlined, the eXtended Design Structure Matrix (XDSM) is explained and University Conceptual Aircraft Design and Optimization (UNICADO) is briefly introduced as well as its connection to MDA. In Sect. 3, the current MDA in UNICADO is analyzed and different approaches for its optimization are presented and applied. In Sect. 4, results of the different optimization strategies are compared to the initial architecture. Finally in Sect. 5, the results are discussed and concluded.

2 Multidisciplinary analysis

In this section, we explain the term MDA, followed by its application to conceptual aircraft design in general concluding with an introduction of UNICADO and its connection to MDA.

2.1 Fundamentals

Aircraft design in general consist of several disciplines, e.g. aerodynamics, propulsion and performance. Those disciplines are coupled by variables e.g. the performance estimation module needs aerodynamic data to calculate the necessary fuel for a mission. This kind of dependence is called forward coupling. If the disciplines are not only forward, but also backward coupled—i.e. the mass estimation provides a value for OME which is used by the mission analysis to calculate a m_{fuel} and a new MTOM, which are then in turn used by the mass estimation to calculate the next updated OME—they have to be executed iteratively. The gathering of the disciplines into a single analysis is called MDA. In MDA, the disciplines are executed iteratively until each target vector of objective variables \mathbf{y}_i^t of discipline i is equal to the response vector of objective variables \mathbf{y}_i^r with respect to a permitted absolute residual ρ . The analysis is then considered converged:

$$\mathbf{y}_i^t = \mathbf{y}_i^r + \rho. \quad (1)$$

In this paper, we will use the following relative residual ϵ formulation:

$$\epsilon = \left| \frac{\mathbf{y}_i^t - \mathbf{y}_i^r}{\mathbf{y}_i^t} \right| \quad (2)$$

An extensive survey of architectures for multidisciplinary analysis and optimization can be found in Martins [2]. For the iterative solving of an MDA there exist several methods,

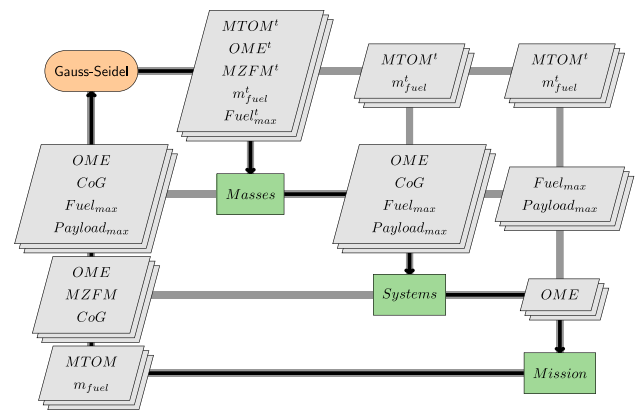


Fig. 1 Gauss-Seidel architecture in XDSM view

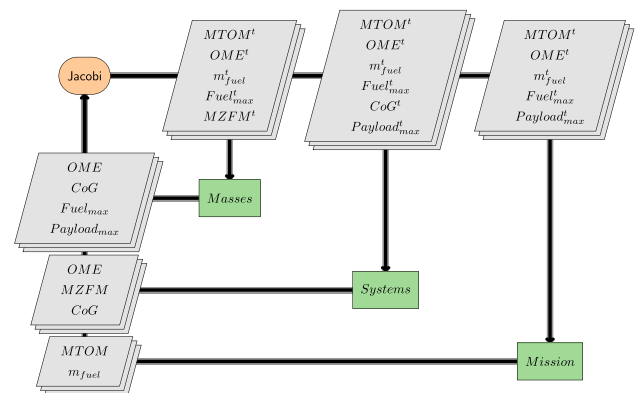


Fig. 2 Jacobi architecture in XDSM view

where common methods are Gauss-Seidel (Fig. 1), Jacobi (Fig. 2) and Newton.

For the final MDA architecture, each aircraft design module is considered as a black box, so that only the inputs and outputs are known, but not the system of equations inside. Therefore, the Newton method cannot be applied here, because it solves systems of equations of the different disciplines simultaneously. The other two, Gauss-Seidel and Jacobi are so called fixed-point approaches, where each discipline is solved individually holding the others fixed. Note that for the convergence of both iterative methods in mathematical sense, the system of equations must fulfill certain conditions, among others that the system's matrix has to be symmetric positive definite. However, it is unclear how these convergence criteria can be verified for a system of black boxes, where no equations are accessible. The mathematical theory behind the iterative methods Gauss-Seidel and Jacobi is given in [3]. The architectures in Figs. 1 and 2 are depicted as an extended design structure matrix (XDSM) [2, 4]. The respective iterative method is depicted as an orange, rounded rectangle. Each design discipline is presented as a

green rectangle. In the rhomboids, input and output variables from or to design disciplines or to the iterative method are listed. To illustrate that a given input consists of more than the listed variables, the rhomboids are stacked. The data flow is presented as grey bars, while the process flow is described by black lines with arrows as endings. The data inflow is presented by the vertical grey bars above and below each discipline. The data outflow is described by the horizontal grey bars to the left and to the right of each discipline. As depicted in Fig. 1, within the Gauss-Seidel architecture each discipline is executed sequentially (following the black line). Consequently each discipline is updated with the results of the previous discipline within the same iteration. E.g. the discipline Masses gets a target operating mass empty OME' as an input, recalculates an update of OME and provides the results as input for the discipline Systems. Systems in turn recalculates an update of OME and provides it as an input for Mission. Within the Jacobi architecture in Fig. 2, the disciplines are executed in parallel. Inside one iteration, there is no exchange of calculated variable updates. All updated values are provided as an input in the next iteration. Therefore, the Jacobi architecture needs more iterations than the Gauss-Seidel method to converge [5, 6]. The architectures depicted in Figs. 1 and 2 are schematic to clarify the characteristics of the individual architecture and are simplified without backward couplings between each discipline.

2.2 UNICADO

UNICADO stands for university conceptual aircraft design and optimization and is on the one hand a software for conceptual aircraft design [7] and on the other hand a project funded by LuFo, the German Federal Aviation Research Programme [8]. The software is derived from the well established software multidisciplinary integrated conceptual aircraft design and optimization (MICADO), developed from the institute of aerospace systems ILR at RWTH Aachen University [9, 10]. The software is developed in C++ and is designed modular i.e. each aircraft design discipline has its own module which is present as

an own executable. The modules are integrated and the execution sequence is controlled via the Remote Component Environment (RCE) [11, 12]. The aircraft parameters are exchanged via one central aircraft exchange file, which has a human readable eXtensible Markup Language (XML) format. The execution sequence of the UNICADO design disciplines is depicted in Fig. 12 and consists of two pre-sizing modules, followed by a design-loop, continued by a loop where a study mission is simulated leading to some post-processing modules. The initial Gauss-Seidel architecture implementation of the design-loop of UNICADO in RCE is depicted in Fig. 3.

Within this paper, we focus on the MDA of the design-loop only. Note that methods used in the disciplines are of varying fidelity, therefore, no system of equations exists nor gradient information is provided. Hence, the respective MDA treats the disciplines as black boxes where only input and output information is available. The currently implemented architecture is Gauss-Seidel (Fig. 12). The aircraft design with UNICADO is assumed converged, if the relative change of the objective variables Maximum Take-Off Mass (MTOM), Operating Mass Empty (OME), mission fuel (m_{fuel}) and Center of Gravity (CoG) in x -Direction are below a permitted relative residual ($\epsilon \leq 2.5e-3$) as stated in Eq. 1. Furthermore, in default setting, the following boundary condition on overall aircraft design level must be fulfilled:

$$C_M = 0 + \xi, \quad \text{with } \xi < 1e-4 \quad (3)$$

i.e. the angle of incidence of the horizontal tailplane (HTP) is recalculated after each iteration in order to satisfy that the moment derivative in mid-cruise condition is almost 0 with a permitted residual. This condition states that the aircraft is trimmed during cruise flight. Note that on discipline level further boundary conditions must be fulfilled, e.g., the force equilibrium for mission calculation, which are not explicitly stated here in detail.

For the studies within this work, two aircraft, designed with UNICADO are used, namely the CeRAS Short Range

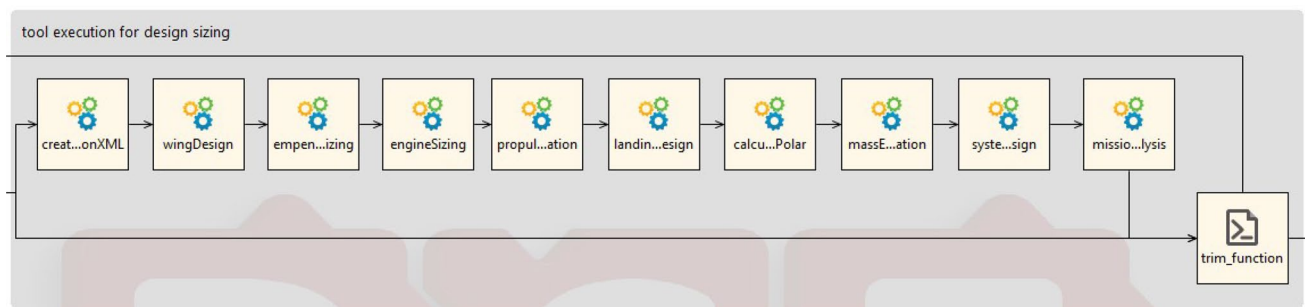


Fig. 3 Implementation of design loop of UNICADO in RCE as Gauss-Seidel architecture

(CSR) and the CeRAS Short-Medium Range (CSMR). The former is also public available in the Central Reference Aircraft Data System (CeRAS) [13, 14].

3 Analysis of architectures

In this section, we analyze the application of Gauss-Seidel and Jacobi architecture for aircraft design with the UNICADO software.

3.1 Gauss-Seidel architecture

In the following, an optimal execution sequence within the Gauss-Seidel architecture is derived for a minimized iteration number for the aircraft design to converge. Furthermore, the discipline sensitivities are analyzed in order to elaborate a permitted residual for robust convergence of the aircraft design with UNICADO.

3.1.1 Execution sequences

A subset of the Gauss-Seidel architecture for UNICADO is depicted in Fig. 4, where also the backward couplings are plotted.

E.g. On one hand, Systems reads OME as an input passed by Masses and on the other hand, Systems in turn calculates OME and provides it backward as an input to Masses. The convergence rate of a MDA is in general influenced by the coupling, on the one hand by the number of feedback variables [5] and on the other hand by the sensitivity of each discipline on those variables. This sensitivity is also called strength of coupling [15]. At first, the disciplines of the UNICADO architecture are analyzed with a visualization tool for large MDO systems, called VISTOMS [16]. In the backend of this visualization tool, graph-based algorithms

(KADMOS [17]) are implemented, among others, to reduce the backward coupling of the MDA problem. The brute-force approach is applied to determine the discipline sequence with a minimized backward coupling. In Fig. 12, the initial architecture of the UNICADO software is depicted. With the application of KADMOS brute-force approach, the execution sequence and the backward couplings are optimized (from 70 to 66 feedback variables (Fig. 13)). Note that only the disciplines in the design-loop are illustrated, because we focus on the MDA part of the design steps and neglect the studies and post-processing steps.

Each discipline needs a certain set of input variables and initial values (guesses) for these variables. For a proper provision of the initial values, there are several boundary conditions to the permitted execution sequence. At least the first execution has to be in a sequence, where all disciplines are provided with the necessary inputs i.e. not every discipline is capable to calculate its results with 0 as start value. Alternatively, the aircraft exchange file has to be filled with initial values. The necessary variables are depicted in Fig. 14. We assume, with regard to the number of coupling variables that the main driving disciplines within the design-loop are

- landingGearDesign
- calculatePolar
- massEstimation
- systemsDesign
- missionAnalysis.

Furthermore, except for landingGearDesign, the listed disciplines are analysis disciplines, which compute the objective variables, i.e. they drive convergence of the aircraft design. Let us denote the permitted process step number of each discipline with $\text{psn}()$, then the following rules apply for these remaining disciplines according to Fig. 14:

1. $\text{psn}(\text{systemsDesign}) > \text{psn}(\text{landingGearDesign})$
2. $\text{psn}(\text{systemsDesign}) > \text{psn}(\text{massEstimation})$
3. $\text{psn}(\text{missionAnalysis}) > \text{psn}(\text{calculatePolar})$
4. $\text{psn}(\text{missionAnalysis}) > \text{psn}(\text{massEstimation})$

As stated before, the convergence rate is not only influenced by the number of feedback variables, but also by the sensitivities of each discipline on these feedback variables. That is the reason why in the next step we examine if the execution sequence proposed by the application of VISTOMS leads to a minimum number of necessary iterations until the aircraft design is converged. If we permute the possible execution sequences of all disciplines in the design-loop, there would be $10! = 3628800$ possible execution sequences to test. By only examining the main driving disciplines there are $5! = 120$ possible executions sequences left. By applying rules 1. – 4., the resulting number of possible execution

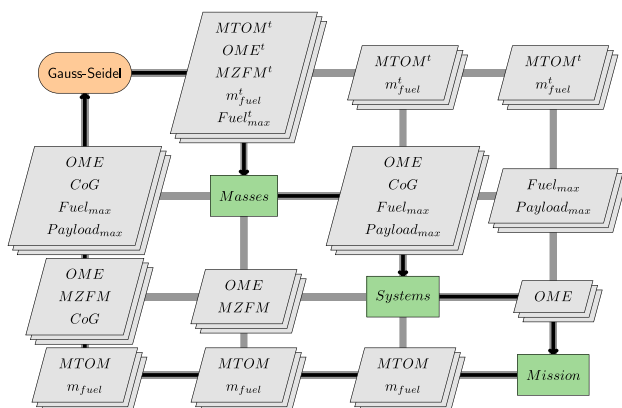


Fig. 4 Gauss-Seidel architecture in XDSM view with backward coupling

sequences equals 16. In these 16 sequences, the initial and the proposed sequence are included. As an initial comparison of the iteration numbers, we execute UNICADO to design the CSR aircraft with default settings for convergence as stated in Sect. 2.2, i.e. the trim condition is activated and the residual is $\epsilon \leq 2.5e-3$. In Table 1 for each sequence the necessary number of iterations until convergence of the objective variables, the number of iterations until the trim boundary condition is fulfilled as well as the number of feedback variables are compared.

Note that the number of feedback variables are only those, which are fed back to the above listed main driving disciplines. Because of that it only seems that study 16 has fewer feedback variables, than the proposed study 14 (Table 1). The number of iterations until the objective variables are converged for the first time are similar for all execution sequences independent of the number of feedback variables (5–6 iterations). The major differences lie in the number of iterations until the trim boundary condition is fulfilled. The trim boundary condition needs more iterations to be fulfilled, as the moment derivative is influenced, among others, by the changes of masses, which induce changes in the drag derivative and the CoG. Therefore, when masses are not kept constant, the moment derivative is not a linear function of the angle of incidence of the HTP. In conclusion, for the permitted residual and the trim boundary setting, there is no impact of a minimized backward coupling (proposed study 14) or a reordering of the execution sequence in general.

Table 1 Study of execution sequences for a CSR aircraft design with permitted residual $\epsilon \leq 2.5e-3$, with it 1 = number of iterations until objective variables are converged; it 2 = number of iterations until objective variables are converged and trim boundary condition is fulfilled; feedback = number of feedback variables

study	it 1	it 2	feedback
1	6	28	301
2	6	18	72
3	6	28	301
4	6	18	72
5	6	24	303
6	6	17	74
7	5	12	299
8	5	21	70
9	6	21	297
10	6	31	68
11	5	12	299
12 (initial)	5	21	70
13	5	18	300
14 (proposed)	6	18	66
15	6	29	302
16	6	31	62

To measure the influence of the execution sequence on the convergence of the objective variables, we neglect the trim boundary condition, decrease the residual to $\epsilon \leq 1e-6$ and restrict the allowed maximum number of iterations to 200 (the latter, to limit the computational costs). The results are depicted in Table 2. The majority of sequences do not converge within the permitted maximum number of iterations. The studies 8, 10, 12 and 16 (Figs. 15, 16, 17, 18) converge within 188 iterations. The only characteristic which these execution sequences have in common is that the missionAnalysis discipline is executed at the end of each sequence. The aircraft design results for the four converged designs are identical. For a similar study, only with the permitted residual set to $\epsilon \leq 1e-5$ all studies converged, while again execution sequences 8, 10, 12, 16 converged in a minimum (and equal) number of iterations.

The convergence of the objective variables of randomly picked study 16 is depicted in Fig. 5.

After a steep decrease of the residuals to $\epsilon \approx 1e-4$ for MTOM, OME and m_{fuel} , resp. $\epsilon \approx 1e-7$ for CoG, the residuals are oscillating without further continuous decrease. These oscillations can have multiple reasons. The most obvious reason is the coupling of the objective variables:

$$\begin{aligned} \text{MTOM} &= \text{OME} + m_{\text{fuel}} + \text{Payload}, \\ m_{\text{fuel}} &\leq \text{Fuelmax}, \\ \text{Payload} &\leq \text{Payload}_{\text{max}} \end{aligned} \quad (4)$$

The other reason is that some disciplines might have implicit functional relations e.g. missionAnalysis reads MTOM and m_{fuel} as an input and calculates among others the same variables as an output, i.e.

Table 2 Study of execution sequences for a CSR aircraft with permitted residual $\epsilon \leq 1e-6$ and switched off trim boundary condition; it 1 = number of iterations until objective variables are converged; feedback = number of feedback variables

study	it 1	feedback
1	–	301
2	–	72
3	–	301
4	–	72
5	–	303
6	–	74
7	–	299
8	188	70
9	–	297
10	188	68
11	–	299
12 (initial)	188	70
13	–	300
14 (proposed)	–	66
15	–	302
16	188	62

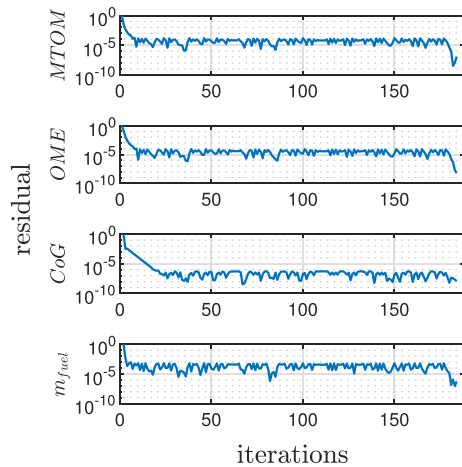


Fig. 5 Convergence of the objective variables in UNICADO for a CSR design with module execution sequence of study 16 and a permitted residual of $\epsilon \leq 1e-6$; switched off trim boundary condition

$$\begin{bmatrix} \text{MTOM} \\ m_{\text{fuel}} \\ \vdots \end{bmatrix} = f_{\text{missionAnalysis}} \left(\begin{bmatrix} \text{MTOM} \\ m_{\text{fuel}} \\ \vdots \end{bmatrix} \right) \quad (5)$$

Furthermore, it is not necessarily given that the discipline itself converges when it is repeatedly executed.

3.1.2 Discipline sensitivities

To derive the behavior of each discipline, we analyze the sensitivities of the disciplines massEstimation, systemsDesign and missionAnalysis. A comprehensive sensitivity analysis would be performed with principle component analysis (PCA) [18]. Because of the large number of input variables of each discipline and the computational cost of computing the sensitivities of all variables, we narrow down to the sensitivity analysis of manually chosen variables, which are the assumed drivers of the convergence rate of the overall aircraft design with UNICADO. For a sensitivity analysis, the second-order central-differences formula [3] is used to calculate the numerical approximations of the derivatives:

$$\frac{dy}{dx} = \frac{y(\mathbf{x} + h) - y(\mathbf{x} - h)}{2h} + \mathcal{O}(h^2) \quad (6)$$

where h denotes the step size, \mathbf{y} are the objectives, \mathbf{x} are the state variables and $\mathcal{O}(h^2)$ denotes the truncation error of the Taylor series expansion. Note that we assume, as a common practice, a continuous relation between input and output variables for small deviations in the inputs and additionally differentiability, even though we know that the discipline behavior can be discontinuous because of the presence of conditionals within the code of each discipline. We analyze the sensitivities for each of the three named disciplines of

the seven input variables, also depicted in Fig. 1 and the same seven outputs, namely: MTOM, OME, m_{fuel} , CoG, MZFM, Fuelmax, $\text{Payload}_{\text{max}}$. An input variable can also be the output variable of a discipline because of an implicit functional relation (cf. Eq. 5). For each discipline, a Jacobian matrix is constructed, containing the numerical transposed gradients of each output variable [19]:

$$\mathbf{J} = \begin{bmatrix} \nabla^T \mathbf{y}_1 \\ \vdots \\ \nabla^T \mathbf{y}_p \end{bmatrix} = \begin{bmatrix} \nabla^T \text{MTOM} \\ \nabla^T \text{OME} \\ \nabla^T m_{\text{fuel}} \\ \nabla^T \text{CoG} \\ \nabla^T \text{MZFM} \\ \nabla^T \text{Fuelmax} \\ \nabla^T \text{Payload}_{\text{max}} \end{bmatrix}_{D,h} \quad (7)$$

where $\mathbf{y}_{j=1,\dots,p}$ is the respective output e.g. MTOM. p denotes the number of output variables, in our case $p = 7$. D stands for the respective discipline and h denotes the step size. The numerical gradient is depicted as follows:

$$\nabla \text{MTOM}_{D,h} = \begin{bmatrix} \partial \text{MTOM} / \partial \text{MTOM} \\ \partial \text{MTOM} / \partial \text{OME} \\ \partial \text{MTOM} / \partial m_{\text{fuel}} \\ \partial \text{MTOM} / \partial \text{CoG} \\ \partial \text{MTOM} / \partial \text{MZFM} \\ \partial \text{MTOM} / \partial \text{Fuelmax} \\ \partial \text{MTOM} / \partial \text{Payload}_{\text{max}} \end{bmatrix} \quad (8)$$

where each partial derivative is calculated by Eq. 6. The sensitivity analysis is performed for each discipline for different step sizes, $h \in \{1e-9, 1e-8, \dots, 1e-1\}$. As first observations, we list the input variables which have no impact on a respective discipline.

- Changes in OME, CoG, Fuelmax and $\text{Payload}_{\text{max}}$ have no influence on the outputs of massEstimation (e.g. $\partial \text{OME} / \partial \text{CoG} = 0$)
- Changes in MTOM and m_{fuel} have no influence on the outputs of missionAnalysis (e.g. $\partial \text{MTOM} / \partial \text{MTOM} = 0$; note that the assumption of Eq. 5 does not hold)
- Changes in MZFM have no influence on the outputs of systemsDesign (e.g. $\partial \text{OME} / \partial \text{MZFM} = 0$)

As the disciplines are executed with default settings, the use and impact of input variables can change if settings are modified and therefore, other calculation and estimation methods are used. Next, we sum up the variables which are only bypassed by a discipline, i.e. the relation is not an implicit function and is characterized as a partial derivative value of 1, e.g. $\partial \text{CoG} / \partial \text{CoG} = 1$.

- In massEstimation: m_{fuel} and MTOM are bypassed (Fig. 6, upper plot)

- In missionAnalysis: OME, Fuelmax Payload_{max} and CoG are bypassed (Fig. 7, upper plot)
- In systemsDesign: CoG is bypassed (Fig. 8, upper plot)

Exemplary some partial derivatives for massEstimation, missionAnalysis and systemsDesign are depicted in Figs. 6, 7 and 8. The bypassed variables in theory do not change their value independent of the step size. Therefore, it can be derived that due to output precision of the disciplines or due to rounding errors [3] in the central-differences method the maximum precision which can be analyzed for the disciplines is $1e-6$. For step sizes below $1e-6$ even the bypassed variables seem sensitive to a variation of themselves, which can only occur due to numerical rounding errors. As depicted in Fig. 7, the sensitivities of missionAnalysis, $\partial MTOM/\partial OME$ and $\partial m_{fuel}/\partial OME$ start to vary from a step size of $< 1e-3$ in magnitude and in sign of the gradient. These characteristics are indicators or at least facilitate oscillating convergence behavior of the objective variables. For decreasing step sizes from $1e-4$, the partial derivatives change the sign of the gradient, i.e. the behavior changes from

for increasing OME, also m_{fuel} increases
to

for increasing OME, m_{fuel} decreases.

This behaviour occurs due to a reserve fuel calculation method, which does not use the most recent trip fuel mass and is part of active maintenance. Compared to Fig. 5, there exists a correlation between the sensitivity and the oscillating behavior of convergence. E.g. MTOM oscillates for residuals in range $1e-4 \leq \epsilon \leq 1e-6$ and m_{fuel} oscillates for residuals in range $5e-4 \leq \epsilon \leq 1e-6$. In conclusion, we cannot expect a stable convergence behavior of the objective variables for permitted residuals below $1e-4$ due to varying sensitivities in slope and sign of the gradients of the discipline missionAnalysis for step sizes below $1e-4$. As UNICADO is actively maintained, we are analyzing and improving the convergence behavior of missionAnalysis.

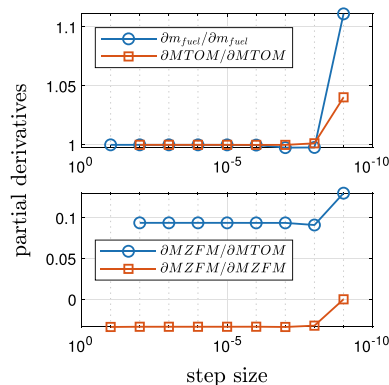


Fig. 6 Numerical partial derivatives for massEstimation for decreasing step sizes from $1e-1$ to $1e-9$

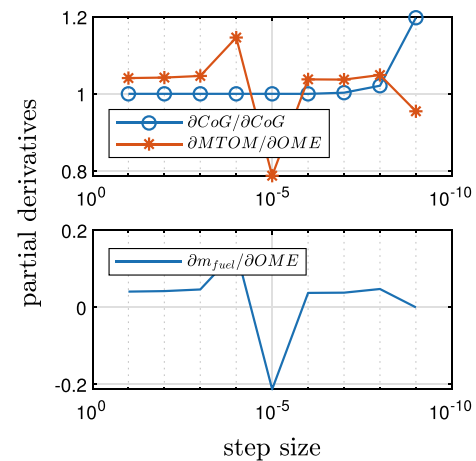


Fig. 7 Numerical partial derivatives for missionAnalysis for decreasing step sizes from $1e-1$ to $1e-9$

3.2 Jacobi architecture

In this section, we collect prerequisites for the implementation of the Jacobi architecture within UNICADO, describe the implementation and the application of the architecture, and finally add a damping method to impose convergence of the aircraft design.

3.2.1 Prerequisites for parallel execution of design disciplines

The previous studies are applied on a sequential execution sequence within the Gauss-Seidel architecture. In this section, prerequisites for a parallel execution of the UNICADO design disciplines for an application of the Jacobi architecture are collected. As depicted in Fig. 14, each discipline has necessary initial input variables, which have to be provided.

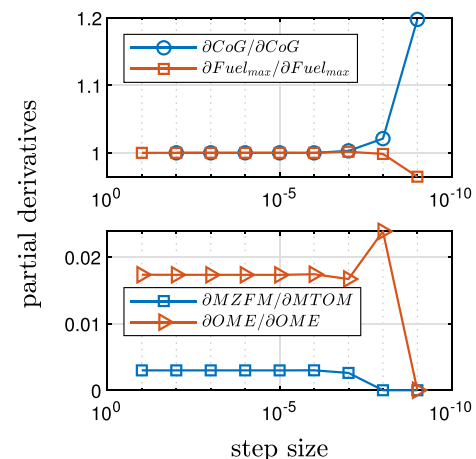


Fig. 8 Numerical partial derivatives for systemsDesign for decreasing step sizes from $1e-1$ to $1e-9$

If the disciplines are executed in parallel, the initial values would be missing. Therefore, the disciplines either have to be executed one time in sequential sequence, or the aircraft exchange file must provide those initial values.

As different disciplines may write the same variable, there has to be a merge logic, which ensures that the respective variable must be written, by the latest discipline from the sequential execution sequence to ensure comparability between sequential and parallel execution. Note that within a parallel execution, calculated output variables from a discipline will be provided to the other disciplines in the next iteration of the Jacobi architecture, i.e. there is no variable update within one iteration (Fig. 2).

When different disciplines access (read / write) the same variable, while they are executed in parallel, there are two options, how to handle the parallel access. Either one central file is used—then, the access of the file has to be locked during the write process of a respective discipline to avoid another discipline reading an outdated variable—or local copies of the aircraft exchange file have to be created for each discipline, which have to be merged at the end of each iteration within parallel execution.

The first approach with one central aircraft exchange file leads to overwriting and processing an incomplete file, because during the creation of the lock file, data are already processed. Therefore, we chose the latter approach.

3.2.2 Implementation of Jacobi architecture

In this section, we describe, how we implement the Jacobi architecture for the UNICADO design disciplines. As stated in Sect. 2.2, we use RCE for the integration of the design disciplines. As further stated in Sect. 3.2.1, the design disciplines first are executed sequentially for one iteration to provide necessary initial values for all disciplines. Next

local copies of the aircraft exchange file are provided to each discipline. All disciplines of the design loop are executed in parallel. Over a signal handling within RCE, the merge script waits until all disciplines are executed and merges each local copy of the aircraft exchange file back to a central aircraft exchange file. Each local copy is merged in the initial sequential execution sequence of the disciplines into the central aircraft exchange file. For merging, the module execution order of the Gauss-Seidel architecture is used to ensure comparability with it. Note that also the merge time at the end of each iteration has to be taken into account for a comparison of the MDA architectures. The implementation of the Jacobi architecture is depicted in Fig. 9.

3.2.3 Application of Jacobi architecture

The convergence progression of an aircraft design of the CSMR with UNICADO with an applied Jacobi architecture within RCE is depicted in Fig. 10a. The aircraft design in this case does not converge. The objective variables m_{fuel} and MTOM oscillate each around a fixed residual value. The oscillations of m_{fuel} have a magnitude of $4\text{e}-2$ which means 4% of mission fuel mass. In the case of the aircraft design of CSMR, this means a variation of approx. 678 kg in fuel mass. The oscillating residuals of MTOM have a magnitude of $< 1\text{e}-1$, which results for the CSMR in approx. 790 kg.

The oscillations can be caused by different circumstances. As evaluated in Sect. 3.1.2, oscillations of the objective variables m_{fuel} and MTOM occur in the magnitude of $1\text{e}-4$ due to discipline sensitivities of missionAnalysis. This might—but not necessarily—be an explanation of the oscillations, but cannot be the only explanation for the magnitude of the oscillations in the application of the Jacobi architecture. From the MDA representations in Figs. 1 and 2 we derive that MTOM and m_{fuel} are calculated by missionAnalysis.

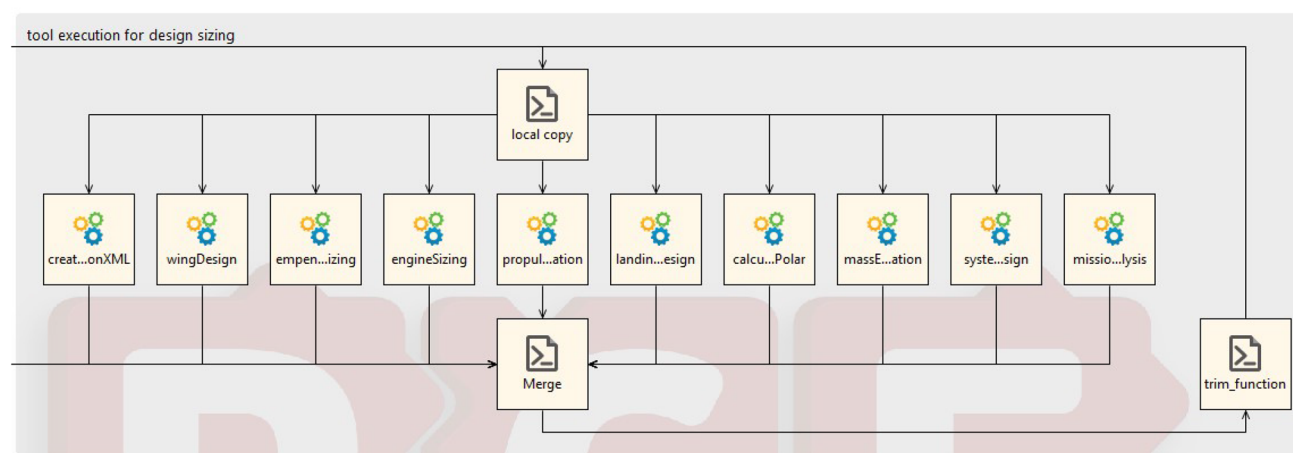


Fig. 9 Implementation of design loop of UNICADO in RCE as Jacobi architecture

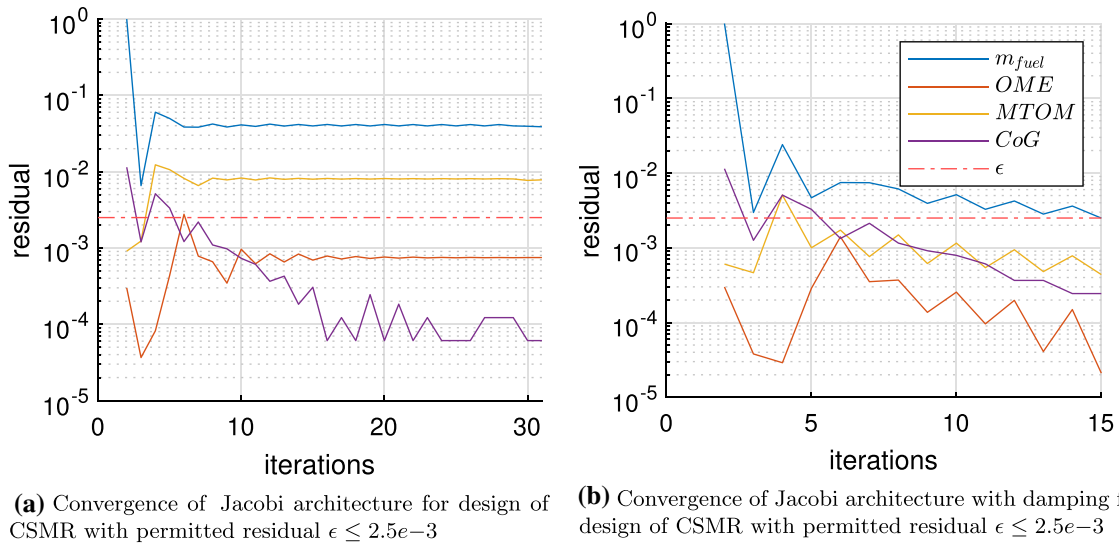


Fig. 10 **a** Convergence of Jacobi architecture for design of CSMR with permitted residual $\epsilon \leq 2.5e-3$. **b** Convergence of Jacobi architecture with damping for design of CSMR with permitted residual $\epsilon \leq 2.5e-3$

From the sensitivities of missionAnalysis in Fig. 7, we conduct that MTOM and m_{fuel} are mainly sensitive to changes in OME. Where OME is on the one hand calculated by massEstimation and on the other hand by systemsDesign. OME calculated from massEstimation is in turn sensitive to changes in MTOM and $MZFM$. OME calculated by systemsDesign is again sensitive to changes in MTOM, OME and Fuelmax.

In Gauss-Seidel architecture systemsDesign is fed with an update of Fuelmax and OME which is calculated from massEstimation, within the same iteration in order systemsDesign to calculate an updated OME and $MZFM$.

In Jacobi architecture, the OME calculated from massEstimation is never used, because it is calculated in parallel from systemsDesign and overwritten by systemsDesign, i.e. systemsDesign receives its own calculated OME from the last iteration. In addition, systemsDesign uses the Fuelmax value from the last iteration to calculate a new OME.

In conclusion, the oscillations of MTOM and m_{fuel} calculated by missionAnalysis (Fig. 7) within the Jacobi architecture (Fig. 10a) might result from large variations of OME calculations of systemsDesign (Fig. 8) which in turn can be caused by outdated values of Fuelmax and overwritten values of OME (Fig. 6) calculated by massEstimation (Fig. 2).

3.2.4 Jacobi architecture with damping

One way to improve the convergence characteristic of the Jacobi architecture is to add a relaxation or damping method [5]. A damped, updated vector of the objective variables can be expressed as:

$$\mathbf{y}^{(i+1)} = \mathbf{y}^{(i)} + \theta^{(i)} \Delta \mathbf{y}^{(i)} \quad (9)$$

where $\mathbf{y}^{(i)}$ is the vector of objective variables of the previous iteration, θ is a damping factor and $\Delta \mathbf{y}^{(i)}$ is the difference between the objective values of the current iteration $\hat{\mathbf{y}}^{(i+1)}$ and the previous iteration:

$$\Delta \mathbf{y}^{(i)} = \hat{\mathbf{y}}^{(i+1)} - \mathbf{y}^{(i)} \quad (10)$$

As a damping factor, we set

$$\theta^{(i)} = \left(1 + \frac{i}{2}\right)^{-1} \quad (11)$$

Note that the damping factor implies the following behavior for increasing iteration number i :

$$\lim_{i \rightarrow \infty} \mathbf{y}^{(i+1)} \rightarrow \mathbf{y}^{(i)} \quad (12)$$

With this damping factor, we impose convergence for an increasing number of iterations. As depicted in Fig. 10b, the aircraft design of the CSMR with Jacobi architecture with damping factor converges after 15 iterations.

4 Results

In this section, we present the comparison between the application of Gauss-Seidel and Jacobi architecture. For comparison, we use the CSMR aircraft. As a permitted residual, we set $\epsilon \leq 2.5e-3$. The objective variables are not modified and listed in Sect. 2.2.

As stated in 2.1, we expect that in general for convergence of the Jacobi architecture more iterations are necessary, compared to the Gauss-Seidel architecture. There exists a trade-off between the iteration number and the iteration time per iteration. Note that the computation times heavily depend on the performance of the used computer, the output settings of each discipline (written files) and finally the aircraft and its transport task for which it is designed. Therefore, the times stated below are only typical values. To decide which architecture is the preferred one, let us assume, that one iteration with the Gauss-Seidel architecture for the design loop of UNICADO takes $t_s = 30\text{s}$, where the index s denotes a sequential execution. Let us further assume that the parallel execution of the design loop of UNICADO needs exactly the same time as the discipline with the highest computational cost, neglecting the time needed for creation of local copies of the aircraft exchange file and the merge process after parallel execution. The discipline with the highest computational cost is missionAnalysis and takes 10s. Then, the time needed for a parallel execution equals $t_p = 10\text{s}$. The Jacobi architecture is the preferred architecture if the following condition is fulfilled:

$$t_s n_s > t_p n_p \Leftrightarrow \frac{30\text{sec}}{10\text{sec}} n_s > n_p \Leftrightarrow 3 \cdot n_s > n_p \quad (13)$$

where n_s and n_p denote the number of iterations for a sequential and a parallel execution, respectively. In this example, the Jacobi method is preferred, if the iteration number is smaller than 3 times the iteration number of the Gauss-Seidel method.

As depicted in Fig. 10b, the aircraft design of the CSMR with Jacobi architecture with damping factor converges after 15 iterations. Compared to the design of the CSMR using the Gauss-Seidel architecture, the Jacobi architecture with damping converges slower. The design with Gauss-Seidel architecture needs 6 iterations to converge below a permitted residual of $\epsilon \leq 2.5\text{e-}3$. The iteration time of the Jacobi architecture takes 17s, whereas the Gauss-Seidel architecture needs 20s per iteration. The difference between both iteration times is small due to the fast computation time of each discipline in general and for the Jacobi architecture especially due to the extra amount of time required for the creation of the local copies and the merging operation at the end of each iteration, which is already included in the iteration time. The comparison of both architectures is presented in Table 3.

When decreasing the permitted residual from $1\text{e-}1$ to $1\text{e-}4$ the necessary number of iterations for the Jacobi architecture with damping grows faster, than the necessary number of iterations of the Gauss-Seidel architecture. E.g. for a permitted residual of $\epsilon \leq 1\text{e-}4$ the Jacobi architecture with damping needs approx. 400 iterations, whereas the Gauss-Seidel architecture needs only 11 iterations for convergence (Fig. 11).

Table 3 Comparison of Gauss-Seidel and Jacobi architecture with damping on the design of a CSMR aircraft

	GS	Jacobi w. d.
Number of iterations	6	15
Iteration time	20s	17s

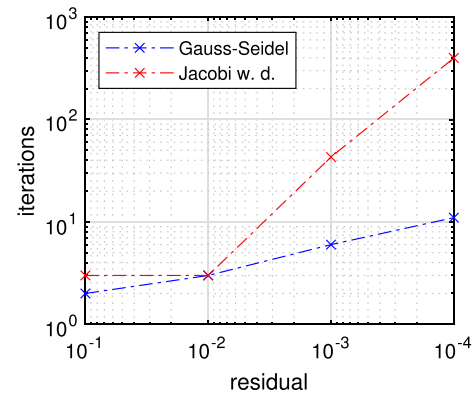


Fig. 11 Convergence of Jacobi architecture with damping for design of CSMR with permitted residual $\epsilon \leq 2.5\text{e-}3$

We conclude that for the design of the CSMR, with a permitted residual of $\epsilon \leq 2.5\text{e-}3$ and the current low computational cost of the design disciplines of UNICADO, a parallelization has no benefit in terms of time saving.

5 Conclusion

In the present paper for the UNICADO software, two MDA architectures were analyzed, namely Gauss-Seidel and Jacobi. For the Gauss-Seidel architecture different execution sequences of the aircraft design disciplines were analyzed. An algorithm for minimizing the feedback variables between design disciplines, returning an in this sense optimal execution sequence was applied. Permitted execution sequences with respect to necessary input variables were derived. The permitted execution sequences were compared to the initial execution sequence as well as to the proposed, optimized execution sequence. This analysis resulted in four execution sequences, which led to the same minimum number of iterations until convergence to a permitted residual. In fact, the proposed, optimized execution sequence was not one of the four execution sequences. The main characteristic, which these execution sequences have in common is that the missionAnalysis discipline is the last executed discipline within an iteration. The execution sequence within the Gauss-Seidel architecture can be arbitrarily chosen from the sequences

depicted in Figs. 15, 16, 17, 18, as UNICADO converges with the same number of iterations for these sequences.

As the convergence rate not only depend on the number of feedback variables, but also on the discipline sensitivities, these were analyzed as well. A main result is that we cannot expect a stable convergence behavior of the UNICADO aircraft design for residuals below $1e-4$, because the partial derivatives $\partial MTOM / \partial OME$ and $\partial m_{fuel} / \partial OME$ change in magnitude and sign of gradients for decreasing step sizes below $1e-4$ which lead to the observed oscillations in the objective variables MTOM and m_{fuel} .

Prerequisites for the implementation and application of a parallel execution of the design disciplines in the Jacobi architecture were collected. The Jacobi architecture was implemented in RCE and applied. The Jacobi architecture did not converge for a permitted residual of $\epsilon \leq 2.5e-3$. m_{fuel} oscillated with a residual of $4e-2$ within the design of a CSMR aircraft. The oscillations are mainly caused by large variations of OME calculated by systems-Design which in turn are caused by outdated values of Fuelmax and overwritten values of OME calculated by massEstimation.

A damping method was applied to the Jacobi architecture to enhance the convergence characteristics. The aircraft design of the CSMR aircraft with Jacobi architecture with damping factor converged. Compared to the Gauss-Seidel architecture the Jacobi architecture with damping factor converges much slower which results in larger overall execution times. Therefore, for the current setup of UNICADO disciplines, the parallelization of discipline execution with the Jacobi architecture has no benefit in terms of computation time and the Gauss-Seidel architecture is the preferred and applied architecture. If the iteration time of the design loop increases, e.g., when using higher fidelity methods for aircraft design, the choice of architecture must be re-evaluated.

Appendix A UNICADO architectures and Discipline relations

See Figs. 12, 13, 14, 15, 16, 17 and 18.

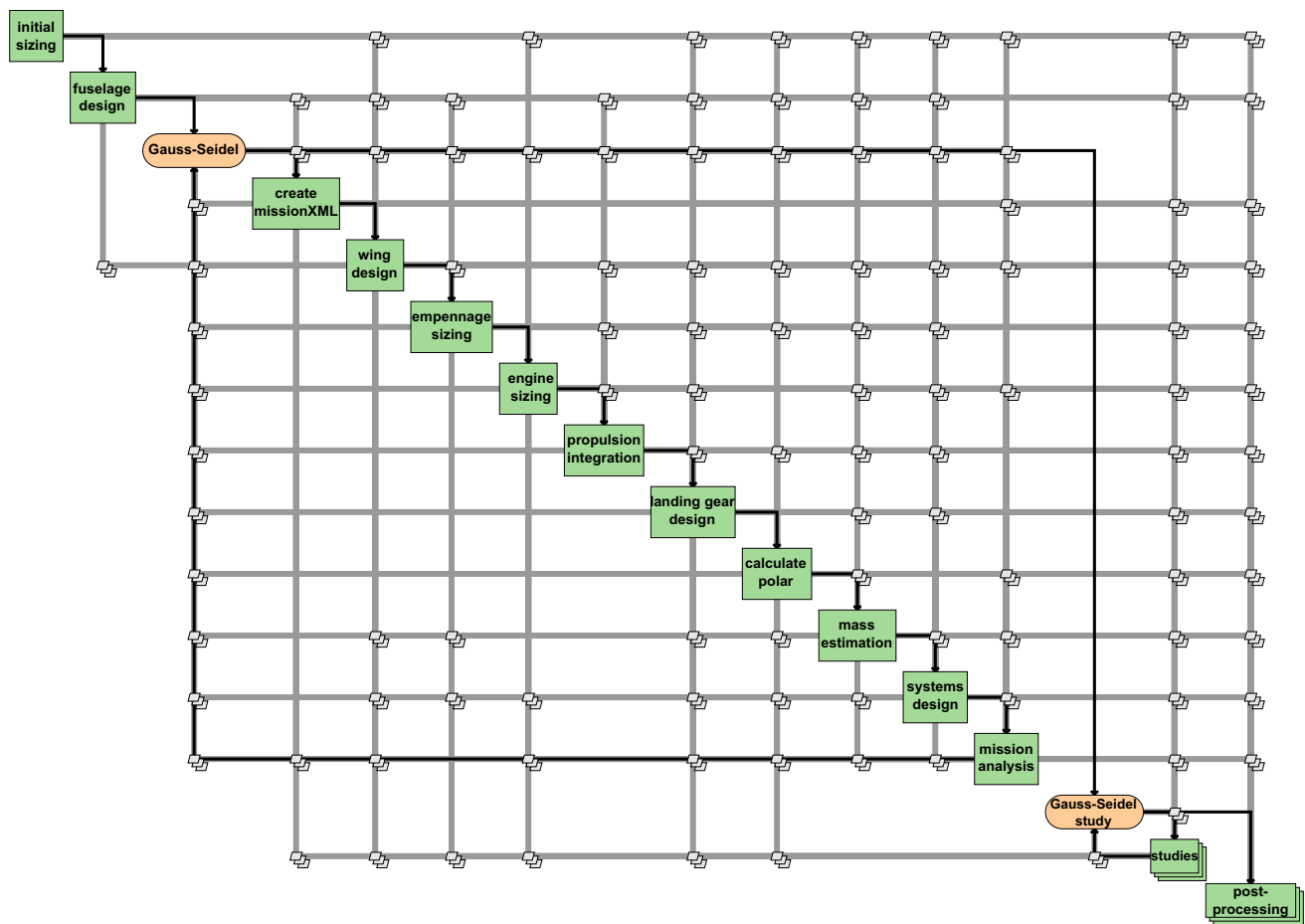


Fig. 12 Initial UNICADO architecture

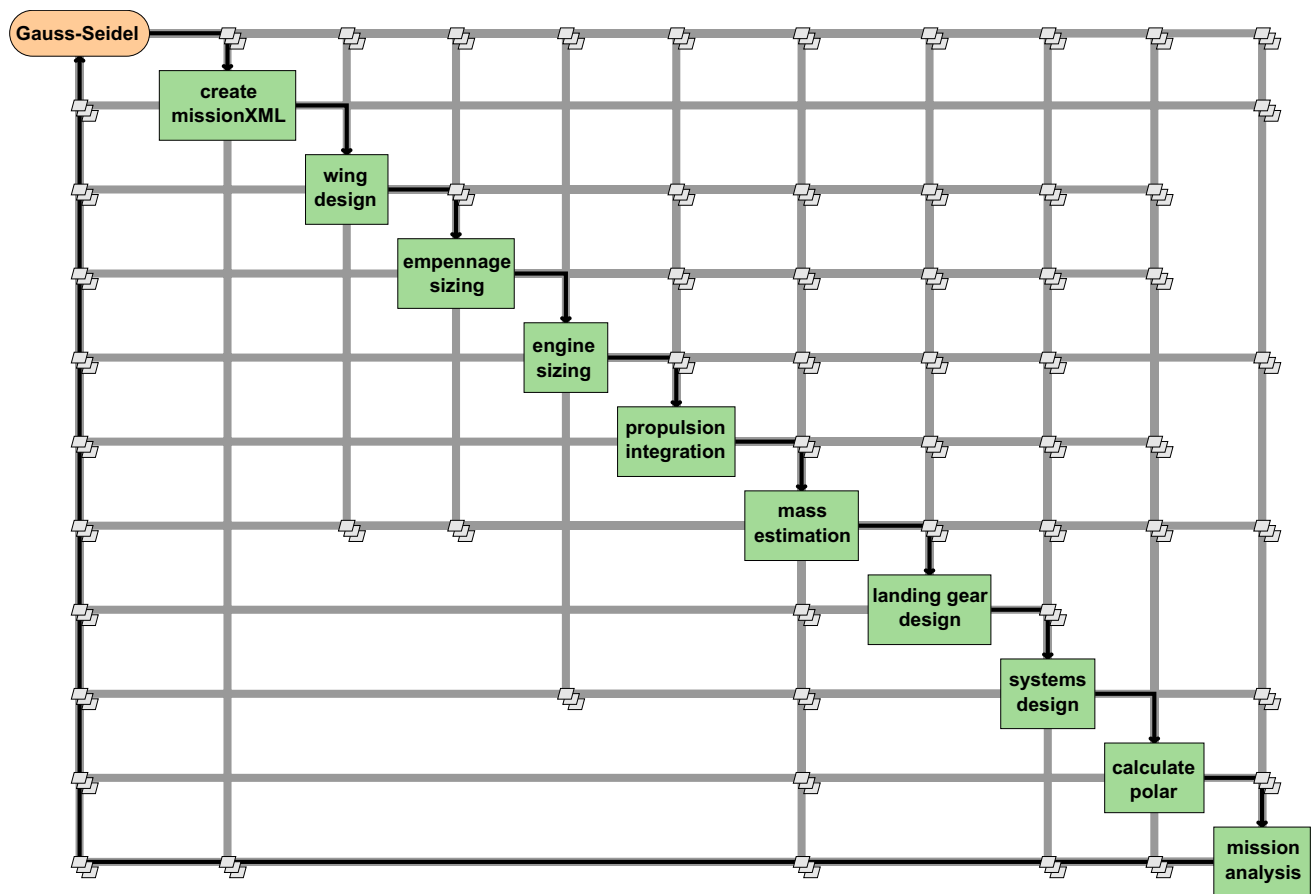


Fig. 13 UNICADO architecture with reduced backward coupling

Fig. 14 Necessary initial values for each UNICADO discipline

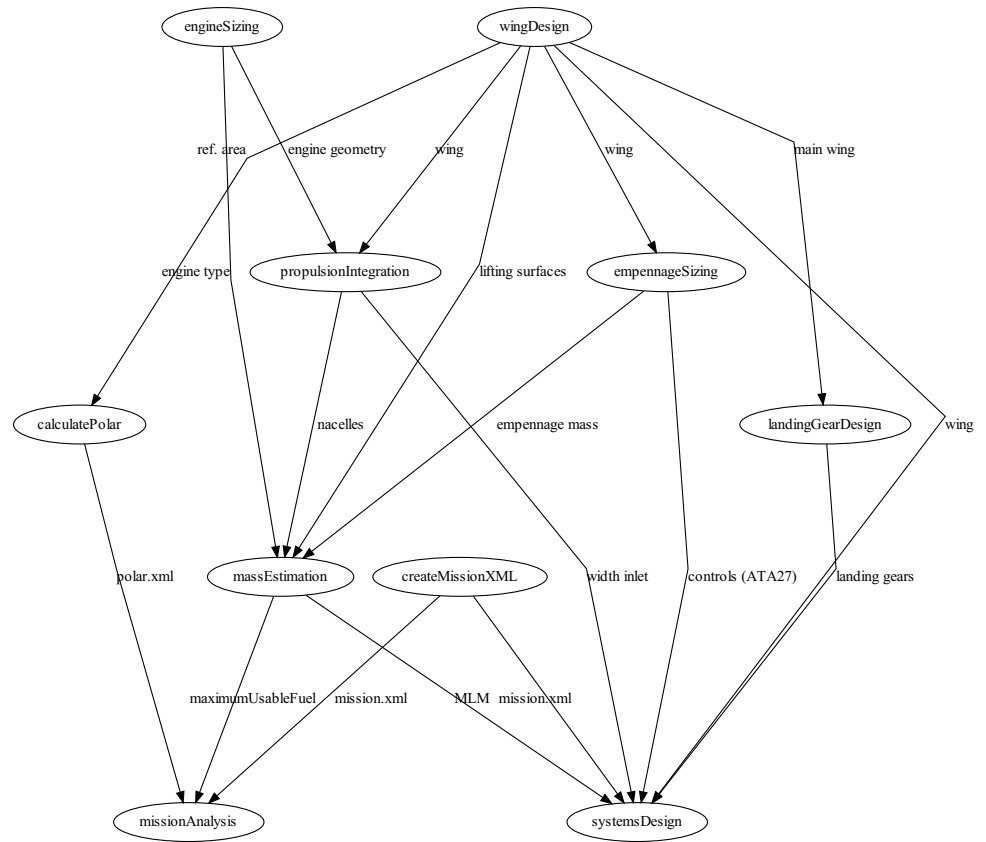


Fig. 15 Execution sequence, study 8

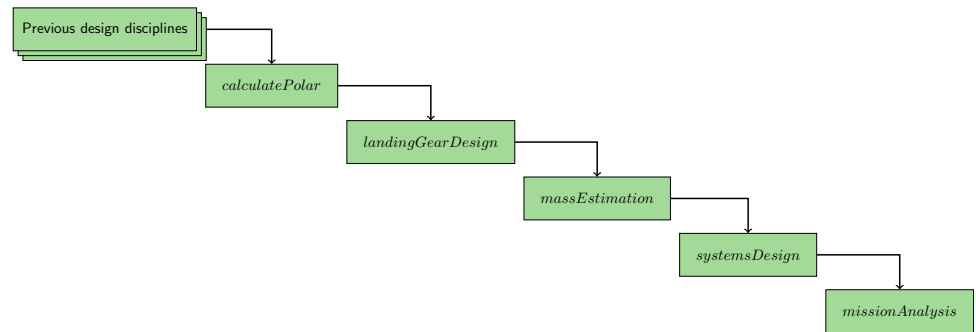


Fig. 16 Execution sequence, study 10

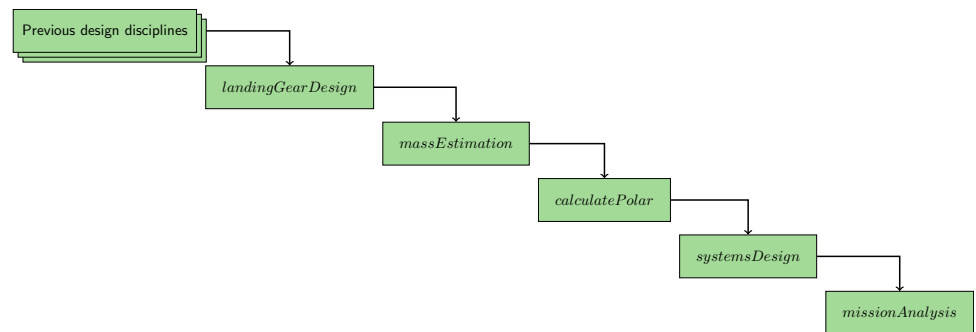
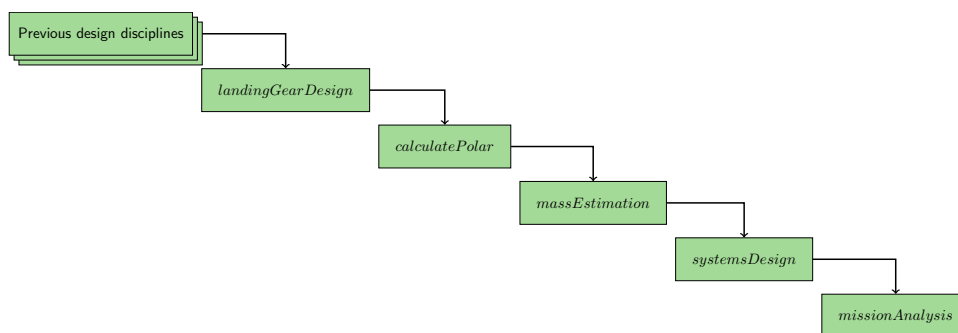
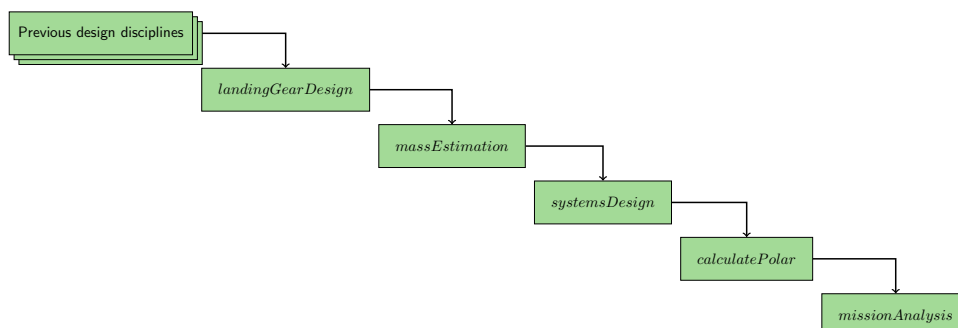


Fig. 17 Execution sequence, study 12**Fig. 18** Execution sequence, study 16

Acknowledgements The research presented in this paper has been carried out in the framework of the UNICADO (UNiversity Conceptual Aircraft Design and Optimization) research project, which has been funded by the Bundesministerium für Wirtschaft und Klimaschutz (BMWK) within the LuFo VI research and innovation program (Grant agreement no. 20E1922).

Funding Open Access funding enabled and organized by Projekt DEAL.

Declarations

Competing interest The authors have no competing interests to declare that are relevant to the content of this article.

Conflict interest The authors have no relevant financial or non-financial interests to disclose. All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript. The authors have no financial or proprietary interests in any material discussed in this article.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Gobbin, A.: UNICADO - Overall Architecture and Workflow User Guide, Technische Universität Berlin, Fachgebiet Luftfahrzeugbau und Leichtbau, Berlin, Germany
- Martins, J.R.R.A., Lambe, A.B.: Multidisciplinary design optimization: A survey of architectures. *AIAA Journal* **51**(9), 2049–2075 (2013). <https://doi.org/10.2514/1.J051895>
- Dahmen, W., Reusken, A.: Numerik Für Ingenieure und Naturwissenschaftler. Springer Berlin Heidelberg, Berlin, Heidelberg (2008). <https://doi.org/10.1007/978-3-540-76493-9>
- mdolab pyXDSM: A python library for generating PDF XDSM diagrams (2022-03-07T18:16:03.000Z). <https://github.com/mdolab/pyXDSM> Accessed 2022-03-07T18:16:03.679Z
- Schumacher, A., Vietor, T., Fiebig, S., Bletzinger, K.-U., Maute, K.: Advances in Structural and Multidisciplinary Optimization. Springer International Publishing, Cham (2018). <https://doi.org/10.1007/978-3-319-67988-4>
- Barrett, R., Berry, M., Chan, T., Demmel, J., Donato, J., Dongarra, J., Eijkhout, V., Pozo, R., Romine, C., Van der Vorst, H.: Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods. Other titles in applied mathematics, vol. 43. Society for Industrial and Applied Mathematics (SIAM 3600 Market Street Floor 6 Philadelphia PA 19104), Philadelphia, Pa (1994). <https://doi.org/10.1137/1.9781611971538>
- Schültke, F., Stumpf, E.: UNICADO : Development and Establishment of a University Conceptual Aircraft Design Environment. RWTH Aachen University. <https://doi.org/10.18154/RWTH-2021-09317>
- Schültke, F., Stumpf, E.: UNICADO: Aufbau und Etablierung einer universitären Flugzeugvorentwurfsumgebung. <https://doi.org/10.18154/RWTH-2021-04203>. <https://publications.rwth-aachen.de/record/817926>
- Risse, K., Anton, E., Lammering, T., Franz, K., Hoernschemeyer, R.: An integrated environment for preliminary aircraft design and

- optimization. In: Structures, Structural Dynamics, and Materials and Co-located Conferences. AIAA Science and Technology Forum, Honolulu, Hawaii (2012). <https://doi.org/10.2514/6.2012-1675>
10. Schültke, F., Aigner, B., Effing, T., Strathoff, P., Stumpf, E.: MICADO: Overview of Recent Developments within the Conceptual Aircraft Design and Optimization Environment. Deutsche Gesellschaft für Luft- und Raumfahrt - Lilienthal-Oberth e.V. <https://doi.org/10.25967/530093>
 11. Seider, D., Wend, J., Schreiber, A., Niemann, J.: Embedding existing heterogeneous monitoring techniques into a lightweight, distributed integration platform. In: Fabr, S.G. (ed.) Third International Conference on Advanced Engineering Computing and Applications in Sciences, 2009, ADVCOMP '09, pp. 57–62. IEEE, Piscataway, NJ (2009). <https://doi.org/10.1109/ADVCOMP.2009.16>
 12. Deutsches Zentrum für Luft- und Raumfahrt: RCE (2022-02-24T08:25:04.000Z). <https://rcenvironment.de/> Accessed 2022-03-09T11:03:24.397Z
 13. Risse, K., Schäfer, K., Schültke, F., Stumpf, E.: Central reference aircraft data system (ceras) for research community. CEAS Aeronautical Journal **7**(1), 121–133 (2016). <https://doi.org/10.1007/s13272-015-0177-9>
 14. Risse, K., Schäfer, K., Schültke, F.: CeRAS - Central Reference Aircraft data System (2022-03-09T12:15:21.000Z). <https://ceras.ilr.rwth-aachen.de/tiki/tiki-index.php?page=Welcome> Accessed 2022-03-09T12:15:21.868Z
 15. Sobieszcanski-Sobieski, J.: Sensitivity of complex, internally coupled systems. AIAA Journal **28**(1), 153–160 (1990)
 16. Aigner, B., van Gent, I., La Rocca, G., Stumpf, E., Veldhuis, L.L.M.: Graph-based algorithms and data-driven documents for formulation and visualization of large mdo systems. CEAS Aeronautical Journal **9**(4), 695–709 (2018). <https://doi.org/10.1007/s13272-018-0312-5>
 17. van Gent, I.: Agile mdao systems: A graph-based methodology to enhance collaborative multidisciplinary design (2019). <https://doi.org/10.4233/uuid:c42b30ba-2ba7-4fff-bf1c-f81f85e890af>
 18. Abdi, H., Williams, L.J.: Principal component analysis. Wiley Interdisciplinary Reviews: Computational Statistics **2**(4), 433–459 (2010). <https://doi.org/10.1002/wics.101>
 19. Modler, F., Kreh, M.: Differenzierbare abbildungen. In: Modler, F., Kreh, M. (eds.) Tutorium Analysis 2 und Lineare Algebra 2, pp. 67–96. Spektrum Akademischer Verlag, Heidelberg (2012). <https://doi.org/10.1007/978-3-8274-2896-7>