AIAA SciTech Forum
8-12 January 2024, Orlando, FL
AIAA SCITECH 2024 Forum

10.2514/6.2024-1802

A Methodology for Modular MDO Model Assembly via a Solver Independent Field Representation

Michael A. P. Warner *,
Andrew H. Fletcher †,
Sebastiaan P. C. van Schie ‡,
John T. Hwang§
University of California San Diego, La Jolla, CA, 92093

With Multidisciplinary Design Optimization (MDO) gaining popularity in a variety of aerospace applications, MDO problems are increasing in complexity. As solvers from new disciplines with multiple fidelity levels are added to an MDO problem, there arises an interface complexity problem - each new solver from a given domain must interface with every other solver of a coupled domain, resulting in a quadratically growing number of interfaces. Additionally, each new discipline adds new design variables and constraints, further adding to the complexity. Simultaneously, there is a desire to simplify the required implementation work to formulate MDO problems, to encourage widespread adoption of MDO. To achieve the goal of simplifying MDO problem formulation given the highlighted complexity, this work introduces the Solver-Independent Field Representation (SIFR) methodology, which unifies the representation of all spatially-distributed quantities, be them design variables, states, or constraints. Simultaneously, this unification of field representations cuts the interface complexity from quadratic to linear by coupling each solver with the solver-independent representation at a framework level, rather than directly to another solver. This paper details the SIFR methodology, and gives a demonstration in the form of a structural optimization problem of an aircraft wing.

I. Introduction

As gradient-based Multidisciplinary Design Optimization (MDO) has gained popularity over time, there has be an increasing interest in developing more detailed MDO problems. Often, this comes in the form of adding additional disciplines and fidelity levels, resulting in a large number of design variables, constraints, and solver-solver interactions. As a consequence, these MDO problems are becoming increasingly more complex, resulting in a desire for methods that simplify the construction of these models. Responding to this, many methods have been developed to tackle various sources of complexity. To reduce effort in computing gradients, automatic-differentiation software such as CSDL [1] and JAX [2] have been developed. Similarly, platforms such as NASA's OpenMDAO [3] and CADDEE [4, 5] give a common interface where solvers can be coupled together for gradient-based optimization. Additionally, a multitude of frameworks have been introduced to tackle solver coupling in MDO, especially aero-structural coupling [6–8]. In this work, complexity arising from solver interfacing is addressed by unifying the representation of all spatially-distributed quantities, allowing solvers to couple to a common framework.

Traditionally, multidisciplinary coupling would require a custom interface between each solver. This limits the ability to add disciplines and swap between solvers, as the number of interfaces grows as the square of the number of solvers. *Mphys* addresses this problem by providing a standardized API to interface solvers with OpenMDAO [3, 9]. This allows solvers within the same domain to be swapped out without the need to define a new coupling scheme. This capability is achieved by interfacing with a set of surface coordinates, on which the states for the coupled problems are transferred. While this approach enables a level of solver modularity, it forces all solvers in a coupling to interface with states at the same predefined points. Additionally, the approach does not guarantee work-conservative coupling when needed. To address these shortcomings, a novel state-transfer scheme is proposed where each solver can define its own state transfer points, and modularity is enabled by a central, solver-independent field representation. The Solver-Independent Field representation (SIFR) methodology proposes both a way of representation field quantities at a

*Ph.D. Student, Department of Mechanical and Aerospace Engineering, AIAA student member
†Ph.D Student, Department of Mechanical and Aerospace Engineering, AIAA student member
‡Ph.D. Student, Department of Mechanical and Aerospace Engineering, AIAA student member
§Assistant Professor, Department of Mechanical and Aerospace Engineering, AIAA senior member

framework level, as well as a scheme for transferring data between solvers and the framework representation. Further, work-conserving coupling can be guaranteed by fitting the continuous state representation under the constraint of equal virtual work, as demonstrated by van Schie et al [10].

Because solvers are able to define their own interface points, the restrictions on the solver interface are minimal. Thus, even one-way coupling, or solver outputs for visualization and other purposes, can use this same interface. The continuous state representation leverages an existing B-spline description of the geometry. This technique uses the mapping from parametric to physical space generated by the B-spline geometry representation of the system to represent states in physical space without duplicating geometry information. Functions describing the spatial distribution of properties and states are defined in parametric space, then transformed into physical space using the geometry mapping.

While motivated by solver-independent state transfer, the same scheme can be used to transfer other spatially distributed data, such as material properties. The use of B-splines for material optimization has been explored, with studies finding reductions in structural weight and increases in stiffness when optimizing tow-steered composites [11, 12]. However, these works used the same B-spline to represent both the geometry and ply angle, meaning knot vectors and order of the geometry and ply angle representation were forced to be the same. By decoupling the geometry and property representations, it is expected that better convergence or lower computational cost will be possible, or both in cases where uneven distributions of control points can be utilized.

In the first part of this work, the solver-independent field representation is described, giving a central representation of all spatially-distributed quantities. Additionally, a method of coupling solvers with the solver-independent representation is detailed, introducing a three-state data transfer technique. The second part of this work applies the SIFR methodology to a structural optimization problem. Specifically, the material thickness of an aircraft wing is optimized, using the SIFR methodology to handle design variables, constraints, and data transfer between an aerodynamic solver and a structural solver.

II. The SIFR methodology

The Solver-Independent Field Representation (SIFR) methodology seeks to introduce a high level of modularity between different solvers in an MDO problem, by coupling each solver with a common, framework-level representation of all field quantities, rather than coupling with other solvers directly. Specifically, it unifies the representation of spatially-distributed quantities such as states, design variables, and other quantities of interest. In the SIFR methodology, field quantities are treated in a similar manner to the geometry. Just as each sub-surface in the geometry acts as a map from its parametric space to physical space, each field quantity is represented by a map from parametric space to the relevant state space. This creates a unified, framework-level field representation, which passes data to solvers via an evaluation at a mesh of points in space. In this section, the details of the solver-independent field representation will be explained, as well as the method to transfer field data between the solver-independent representation and different solvers.

A. The Solver-Independent Field Representation

Under a geometry-centric framework, the geometry is continuously represented by a map from a parametric space to a physical space. Each individual sub-surface in this set can be considered a *geometry function*, selected from an arbitrary function space. The SIFR methodology uses this same technique to represent a given field at the framework level. The solver-independent field representation consists of *field functions*, a set of maps from parametric space to the relevant *field space*. Crucially, the parametric space used for the geometry representation is the same as the parametric space used for the field representation, effectively embedding the field representation within the geometry. Additionally, while they share a parametric space, no other restrictions are placed on the geometry function and the state function.

To demonstrate this technique, consider a geometry representation consisting of a set of B-spline surfaces of the form

$$g(u,v) = \sum_{i=0}^{m} \sum_{j=0}^{n} B_{i,p}(u) B_{j,q}(v) c_{ij} \tag{1}$$

where (u,v) is the input parametric coordinate, $g(u,v)$ is the output spatial coordinate, $B_{i,p}(u)$ and $B_{j,q}(v)$ are B-spline basis functions of degree p and q, and c are the B-spline control points. Additionally, n and m give the number of control points in the u and v direction, respectively Vectorizing this equation, we get

$$g(u,v) = B(u,v)C \tag{2}$$

where B is a matrix mapping from the control point matrix C of shape $(n \cdot m) \times 3$ to to the output spatial points $g(u, v)$. The matrix B represents the function space, in this case a B-spline space with shape $n \times m$ and order $p \times q$. The control point matrix then picks a particular function within the function space to represent the geometry.

Because the geometry will generally be made of a union of such B-spline surfaces, we introduce an *indexed parametric space*, $I \times u \times v$, where $I = (1, 2, ..., l)$ and l is the number of B-splines in the geometry. Then, we can define a function that represents the entire geometry,

$$G(i, u, v) = g_i(u, v) \tag{3}$$

where $g_i(u, v)$ is the geometry function of the form given in equation (2) corresponding to index $i \in I$.

Similarly, while the field representation may be represented using a function from an arbitrary function space, consider a similar B-spline representation,

$$f(u, v) = \bar{B}(u, v)\bar{C} \tag{4}$$

where $\bar{B}$ is a matrix mapping from the control point matrix $\bar{C}$ to to the output field values $f(u, v)$. In this case, $\bar{C}$ is of shape $(\bar{n} \cdot \bar{m}) \times k$, where $\bar{n}$ and $\bar{m}$ are the number of control points in the u and v direction, and k is the dimension of the field quantity. Note that $\bar{n}$ and $\bar{m}$, as well as the order of the field B-spline, may be chosen independently of the parameters selected for the geometry B-spline. This gives a representation of the field over a particular member of the set of B-splines representing the entire geometry.

A field may be defined over an arbitrary subset of the geometry, dubbed the *field domain*. The corresponding *parametric field domain* is an indexed parametric space of the form $\bar{I} \times u \times v$, where $\bar{I} \subseteq I$. As with the geometry, we can define a function to represent the field over the full field domain,

$$F(i, u, v) = f_i(u, v) \tag{5}$$

where $f_i(u, v)$ is the field function of the form given in equation (4) corresponding to index $i \in \bar{I}$. Together, the geometry function and the field function give a map between physical space and field space, as shown in Fig. 1. To evaluate the field at a physical point x, the geometry function is used to find the corresponding index i and parametric point u. Then, the value of that field is given by $f_i(u)$.

This method of field representation has two properties that are crucial to the SIFR methodology. First, this is a continuous, functional representation. This means a field can be evaluated at any set of points in its domain, allowing for maximum flexibility when transferring data two and from solvers. Second, the function used to represent the state is not tied to that used to represent the geometry, and may be selected arbitrarily. This ensures a given field can be represented by a function that gives sufficient accuracy, without effecting the accuracy of the geometry representation. Additionally, different types of function spaces can be selected for different fields as may be needed to best represent the field.

B. Three-Column Data Transfer

Once the solver-independent field representation is established, it must then be evaluated to pass data to solvers and fit to outputs from solvers. This is done by breaking the data transfer process into three distinct stages, with the data representation changing between each stage. This process is shown in Fig. 2. From right to left, the three stages are the framework representation (F), the discrete nodal representation (D), and the representation internal to the solver model (M).

Let z^i be a particular input field to a solver model, and y^j a particular output field. Then, $z^i_{f'}$ and $y^j_{f'}$ are the solver-independent field representations of these two quantities, as detailed above. Similarly, $z^i_{m'}$ and $y^j_{m'}$ are the representations of the fields within a particular solver. This is usually continuous, but the details depend on the particular solver being used. For example, they may be represented using a particular finite element discretization. The discrete nodal representation is used to connect these two functional representations. This consists of a set of points over the field domain at which one functional representation is evaluated, and the other functional representation is fit to. For the input field and output field, z^i_m and y^j_m, respectively, are the discrete representations.

To compute z^i_m, the solver-independent field representation is evaluated at the chosen discrete locations, with $P^i_{mf'}$ being the resulting matrix that maps from the framework representation to the discrete representation. The solver must then compute the matrix $P^i_{m'm}$ which maps from the discrete representation to the solver's internal representation. This process is unique to each solver, and may utilize certain assumptions and interpolation techniques specific to the solver. Note that the location of the discrete representation points may be chosen arbitrarily within the field domain, and thus may be chosen to facilitate this step, such as the nodes of a finite element mesh.

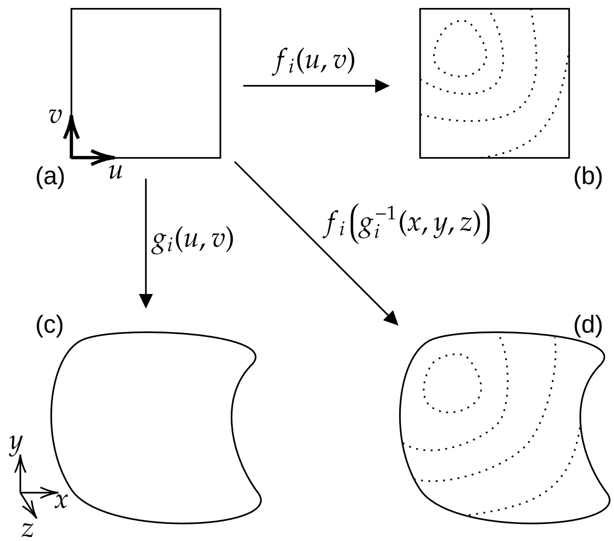


Fig. 1 Diagram of the solver independent field representation: (a) parametric space, (b) contour plot of field quantity corresponding to patch i, (c) geometry patch i in physical space, (d) contour plot of field quantity in physical space

This process is then reversed on the output side, with $Q^i_{mm'}$, computed by the solver, mapping from the solver's internal representation to the discrete representation. Note that nodes this discrete representation are independent of those used for the input z^i, and may be chosen to best facilitate the transfer of this particular field. Finally, The nodal representation is fit to the framework representation via the matrix $Q^i_{f'm}$. The computation of this matrix will depend on the choice the field function space for y^i. If using a B-spline function as given in equation 4, this would be the pseudo-inverse of $\bar{B}(u, v)$, where (u, v) are the parametric coordinates of the nodes in the discrete representation found via the geometry function. This selects a control point matrix $\bar{C}$ which represents a particular function within the chosen function space for the output field.

III. Application to Passenger Air Vehicle (PAV)

To demonstrate the SIFR methodology, a structural weight optimization problem of a wing was formulated. This optimization is built using the Comprehensive Analysis and high-Dimensional DEsign Environment (CADDEE), introduced and demonstrated by Sarojini et al. [4] and Ruh et al. [5]. Both CADDEE and the aerodynamic and structural solvers used in the optimization are built with the Computational Systems Design Language (CSDL), which uses a computational graph to automate adjoint-based sensitivity analysis [1]. The SIFR methodology is implemented in the Multi-fidelity Multi-disciplinary Modelling Language (M3L), which builds on top of CADDEE and CSDL. M3L provides function spaces for the framework representation and automates much of the data transfer between the framework and the solvers, in addition to other functionality that automates MDO model construction. This optimization is focused on the wing of the PAV, with more details about the full aircraft available in [13]. This work was done in collaboration with Aurora Flight Sciences, leveraging their expertise in eVTOLs. A key aspect of this collaboration has been the development of a test case model inspired by Aurora Flight Sciences' Passenger Air Vehicle (PAV). It is important to note that our model does not replicate the actual PAV, but rather draws inspiration from it. To ensure the protection of Aurora Flight Sciences' proprietary information, our approach has been to utilize publicly available data and images of the PAV as a foundation for creating an analogous yet distinct model. This approach allows us to explore design and optimization strategies while respecting intellectual property considerations.

A. Optimization Formulation

The internal structure of the PAV consists of a spar at 25% chord and a spar at 75% chord, with 19 evenly spaced ribs between. This structure is shown in Fig. 4b, while the outer skin of the PAV wing is shown in Fig. 4a. The structure is assigned the material properties of 6061 aluminum. The front and rear spars, and the top and bottom

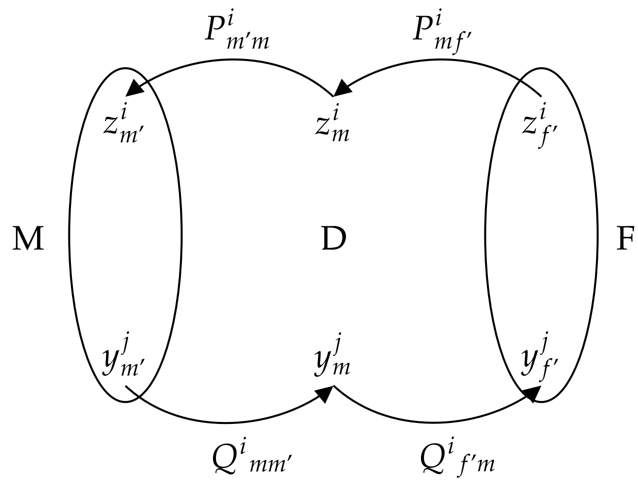


Fig. 2 Illustration of the SIFR process between framework (F) and model (M) with input z and output y

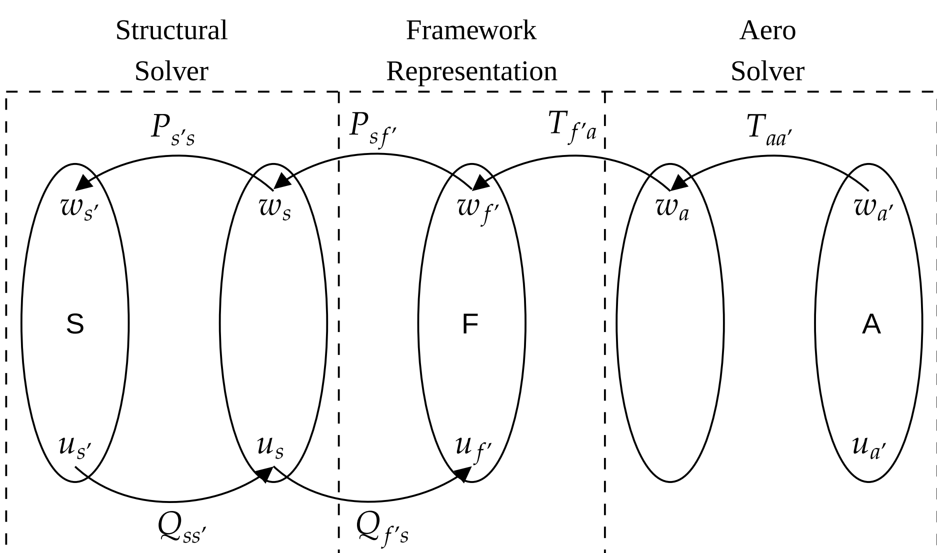
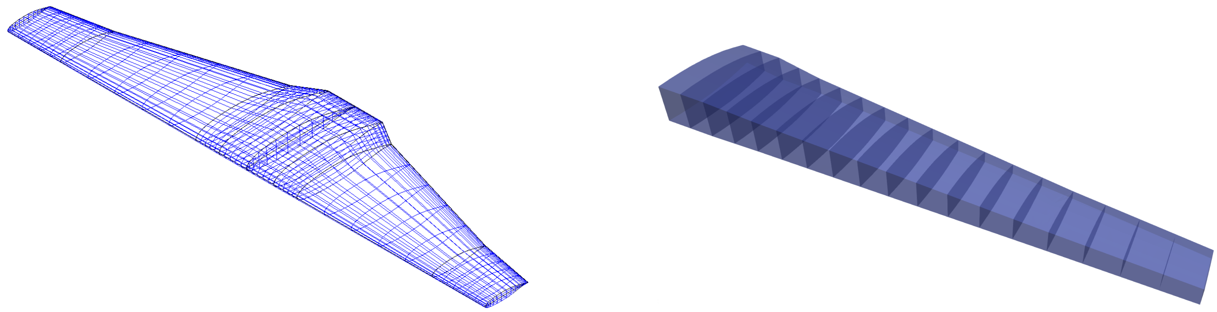


Fig. 3 Data transfer between structural and aerodynamic solver using the SIFR methodology



(a) (b)

Fig. 4 (a) Outer mold line of PAV wing (b) PAV wing-box structure

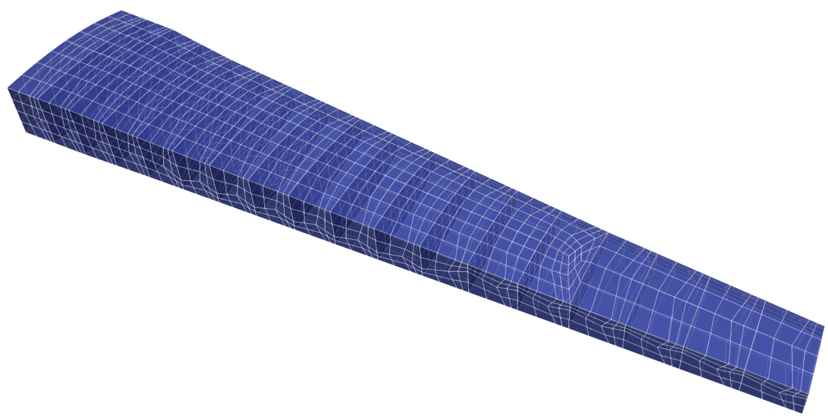


Fig. 5 Quad mesh of PAV wing box with 2472 elements

skin of the wing, are discretized into patches between each spar. This results in 91 distinct geometric patches, each represented via a B-spline surface. The thickness field is represented using SIFR, with each geometric patch receiving a corresponding map to a constant thickness, which is set as a design variable. This results in 91 thickness design variables, corresponding to a constant thickness over each geometry patch. The minimum thickness was set to 0.5 mm. The aircraft is simulated to be operating at 50 m/s at sea level, undergoing a 3g pull up maneuver. The optimization was constrained by the maximum von Mises stress, set to 216 MPa. The optimizer used was SNOPT [14], which converged in under 10 minutes for this problem.

B. Structural Solver

A Reissner-Mindlin shell based finite element model was used as the structural solver for the PAV wing box. This solver was written using the FEniCSx PDE solution platform [15], and connected to CADDEE using the Finite Elements in PDE-constrained Multidisciplinary Optimization (FEMO) toolkit [16]. This shell model is featured in Sarojini et al. [13], where it is compared to other structural solvers under a 3g loading condition of the PAV wing. The structural solver was given an all-quad mesh consisting of 2,472 elements with 2.303 nodes, shown in Fig. 5. The solver uses a linear basis to represent the force inputs, and a quadratic basis to represent the displacement field. Because the SIFR methodology allows forces and displacements to be evaluated at the nodes of the mesh, no interpolation is required during the data transfer.

C. Aerodynamics Solver

The aerodynamic forces are computed using solver based on vortex lattie method (VLM). The VLM solver was written in CSDL, and has been used in various optimization applications [4, 17]. Additionally, it is compared against other aerodynamic solutions using the PAV wing geometry in Sarojini et al. [13]. Wing geometry information is passed to the VLM solver in the form of a mesh of the wing camber surface. The VLM solution returns forces on this camber mesh, however these need to be extrapolated to the wing surface to comply with the SIFR methodology.

To appropriately distribute forces onto the wing surface, a machine-learning based airfoil model is used. This tool, described in Ruh et al. [18], can compute sectional airfoil properties given a airfoil discretized from the wing surface geometry. In this application, the sectional forces are known from the VLM solution, and the airfoil model is used in reverse mode to find a sectional angle of attack that matches the known sectional forces. The model is then run in forward mode with this angle of attack, giving a pressure distribution that is used to distribute the VLM forces onto the wing outer surface. This is then used to fit a framework representation of the forces, for transfer to the structural solver.

D. Framework Representation and Field Data Transfer

There are five field quantities that are represented in this optimization: material thickness, aerodynamic forces, aerodynamic pressures, structural displacements, and structural stress. Because material thickness is constant over each patch, the field function is a constant map, given by

$$T(i, u, v) = t_i(u, v) = t_i \tag{6}$$

where t_i is the thickness design variable for that patch. The pressure, displacement, and stress fields are all represented using a B-spline field function, as given in equations (1) and (4). The control point density and order were adjusted in each case to give an accurate fit to the solver outputs without over-fitting the data. For example, the pressure distribution changes rapidly along the chord of the wing, but gradually along the span of the wing. As a result, a high density of control points were used in the chord-wise direction, and a low density in the span-wise direction.

Finally, the framework representation of the force values required a function that conserved total force when fitted and evaluated. To achieve this, an inverse distance weighting (IDW) scheme was used. Specifically, the framework representation consisted of a grid of points in parametric space, each assigned a force value. To evaluate this function, the distance between each evaluation point and each function point is computed in parametric space. The force from each function point is then distributed to the evaluation points proportional to the inverse of these distances, normalized such that the sum of these inverse distances add to one. This gives the evaluation a similar form to that in equation (4), with the added property that the columns of the $\bar{B}$ matrix, which are the inverse distances corresponding to each function point, sum to one. To fit the force function to the output from the aerodynamic model, this same process is followed, with the forces from the fitting points being transferred to the function points proportional to the inverse distance in parametric space. This ensures that the total force is conserved when transferring to and from the framework representation, while also allowing the forces to be evaluated at any set of points on the wing surface.

The full data transfer scheme can be seen by following the data transfer diagram shown in Fig. 3. Starting with the aerodynamics solver (A), the map $T_{aa'}$ is computed by the machine learning airfoil model to transfer the forces from the VLM camber mesh to a nodal distribution on the wing surface. Next, $T_{f'a}$ is computed via the IDW method described above, to transfer these nodal forces into the framework representation. The framework representation is then evaluated at the nodes of the structural mesh on the wing surface via the IDW scheme, resulting in the map $P_{sf'}$. These nodal forces are then ingested by the structural solver via the map $Ps's$, resulting in the solver's finite element representation of the forces. The structural solver then computes the resulting displacements, which are transferred to a nodal representation by evaluating the finite element representation of the displacements at a dense grid of points, resulting in the map $Q_{ss'}$. The B-spline framework representation is then fit to these nodal values via the pseudo-inverse $Q_{f's}$. Note that this is a one-way coupling scheme between the aerodynamic solver and the structural solver. The effects of adding two-way coupling to this problem, including both work-conservative and non-conservative force-displacement transfer is explored by van Schie et al [10].

E. Optimization Results

The optimization converged in under 10 minutes, resulting in an optimized structural weight and satisfaciton of the maximum stress constraint. Because analysis of the PAV is more thoroughly explored by Sarjoni et al. [13], this section will explore the resulting optimized solver-independent field representations of the thickness, pressure, and stress distributions. Additionally, the specific solvers and fidelity levels chosen are largely independent of the SIFR

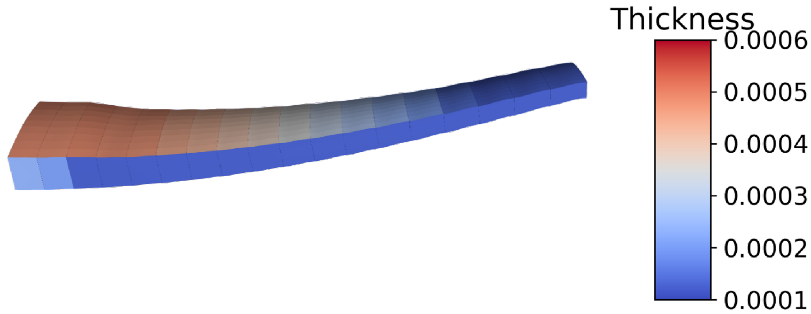


Fig. 6 Optimized thickness distribution of PAV wing (m)

methodology. Either of the structural or aerodynamic solvers could be independently replaced with another formulation, requiring only to interface with the SIFR framework, not any other solver.

1. Thickness distribution

The optimized thickness distribution is shown in Fig. 6. As expected, the surface panels near the root of the wing are the thickest, with most of the spar, the ribs, and the surface panels near the tip of the wing converging to minimum thickness. Because this optimization only considered stress failure and not any global or local buckling, this thickness distribution may not be realistic. However, this does demonstrate the SIFR methodology's use in representing spatially-distributed design variables under optimization. Additionally, because a constant function space was chosen to represent the thickness in each panel, there are discrete jumps in panel thickness across the wing. If optimization of a continuously varying panel thickness is desired, the constant thickness function can be changed for one that allows variation, such as a linear or quadratic B-spline function.

2. Pressure distribution

The pressure distribution resulting from the VLM and machine learning airfoil model is shown in Fig. 7. This distribution is exactly as expected, with pressure concentrated at the leading edge of the wing, dropping off near the wing tip due to finite wing effects. This shows that a good fit to solver outputs can be achieved when using an appropriate B-spline function, even in challenging situations such as this. The framework representation is able to capture the rapid change in pressure along the chord of the wing, without any over-fitting along the span. Additionally, no large discontinuities in pressure are seen across geometry patches, even though continuity was not enforced in this case. Continuity was not enforced because the pressure distribution is being used only for visualization purposes in this case. If continuity is desired, this can be implemented by using clamped B-splines, and setting an equality condition on the values of the control points along the edges of the patch.

F. Stress distribution

The stress distribution resulting from the shell structural analysis is shown in Fig. 8. The peak stress reaches the maximum constraint of 216 MPa, as expected when minimizing the structural weight. Similar to the pressure distribution, the B-spline functions chosen to represent the stress field were able to accurately represent the stress distribution over the structure, without any over-fitting. This demonstrates the application of the SIFR methodology to constraints within an optimization problem, as the constraint is able to be directly applied to the framework representation of the stress.

IV. Conclusions

The growing complexity in MDO problems arising from the integration of multiple disciplines and fidelity levels demands innovative methodologies to simplify model construction while addressing interface complexities. In response to these challenges, this paper introduces the Solver-Independent Field Representation methodology, aimed at unifying spatially-distributed quantities within a common framework. By providing a unified representation for design variables,

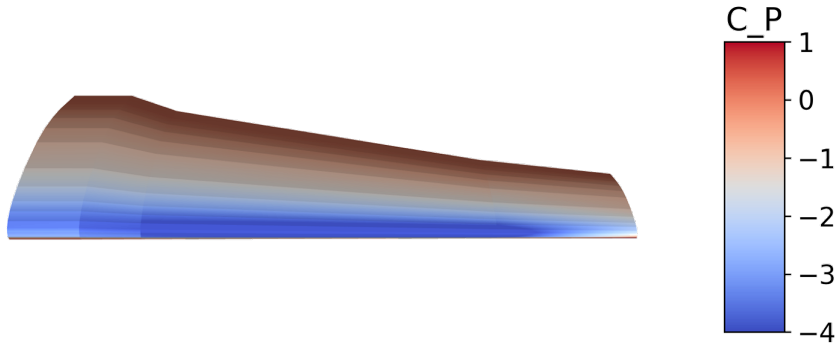


Fig. 7 SIFR representation of pressure distribution over PAV wing

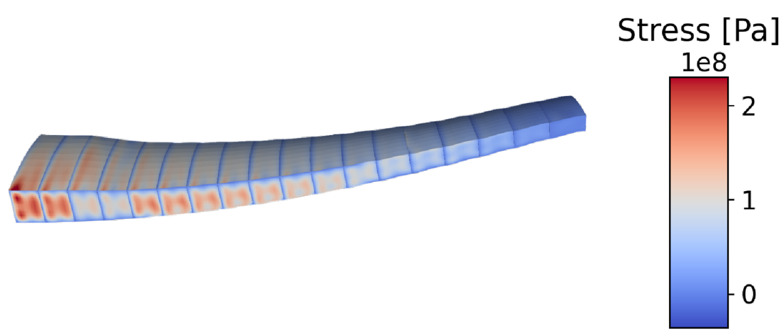


Fig. 8 SIFR representation of von Mises stress distribution over optimized PAV wing

states, and constraints, SIFR significantly reduces interface complexities from quadratic to linear, enhancing modularity and simplifying interactions among solvers. The SIFR methodology offers a novel approach by representing field quantities in a similar way to the geometry, enabling a seamless transfer of data between solvers through a three-state data transfer technique. This technique ensures a continuous, functional representation at the framework level while allowing different function spaces for different fields, maintaining accuracy without compromising the geometry representation.

The application of SIFR in a structural optimization problem of an aircraft wing demonstrates its effectiveness in handling design variables, constraints, and data transfer between aerodynamic and structural solvers. The optimization of the Arroura Passenger Air Vehicle wing showcased the practical implementation of SIFR, exemplifying its adaptability to different field types from different disciplines. By employing SIFR, the optimization achieved the desired structural weight while satisfying stress constraints within a short computational time. The optimized thickness, pressure, and stress distributions validated the methodology's capability to accurately represent spatially-distributed quantities, offering a simplification of MDO problem construction without compromising model complexity.

While this work focused on applying the SIFR methodology to a static, one-way coupling, the methodology itself is applicable to static two-way coupling, dynamic coupling and modal analysis. In future work, this method will be demonstrated in these other contexts, showing the simplicity and flexibility of the SIFR methodology. Demonstrating the representation of time-varying field quantities using this method is of particular interest. Additionally, firm guidelines for the selection of an appropriate function to represent a given field must be developed, along with a method for quantifying the overall error introduced to the optimization from the fitting step. This may take the form of benchmarking analysis and optimization using the SIFR approach against common models, such as the Pazy wing [19] and the Goland wing [20].

V. Acknowledgements

This research was funded through NASA award No. 80NSSC21M0070.

References

[1] Gandarillas, V., Joshy, A. J., Sperry, M. Z., Ivanov, A. K., and Hwang, J. T., "A graph-based methodology for constructing computational models that automates adjoint-based sensitivity analysis," *Structural and Multidisciplinary Optimization*, 2022.

[2] Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q., "JAX: composable transformations of Python+NumPy programs," , 2018. URL `http://github.com/google/jax`.

[3] Gray, J. S., Hwang, J. T., Martins, J. R. R. A., Moore, K. T., and Naylor, B. A., "OpenMDAO: An Open-Source Framework for Multidisciplinary Design, Analysis, and Optimization," *Structural and Multidisciplinary Optimization*, Vol. 59, 2019, pp. 1075–1104. https://doi.org/10.1007/s00158-019-02211-z.

[4] Ruh, M. L., Sarojini, D., Fletcher, A., Asher, I., and Hwang, J. T., "Large-scale multidisciplinary design optimization of the NASA lift-plus-cruise concept using a novel aircraft design framework," , 2023.

[5] Sarojini, D., Ruh, M. L., Joshy, A. J., Yan, J., Ivanov, A. K., Scotzniovsky, L., Fletcher, A. H., Orndorff, N. C., Sperry, M., Gandarillas, V. E., Asher, I., Chambers, J. T., Gill, H., Lee, S., Cheng, Z., Rodriguez, G., Zhao, S., Mi, C., Nascenzi, T., Cuatt, T., Winter, T. F., Guibert, A. T., Cronk, A., Kim, H. A., Meng, S., and Hwang, J. T., *Large-Scale Multidisciplinary Design Optimization of an eVTOL Aircraft using Comprehensive Analysis*, 2023. https://doi.org/10.2514/6.2023-0146, URL https://arc.aiaa.org/doi/abs/10.2514/6.2023-0146.

[6] Niessink, M., "Aeroelastic Aircraft Framework for the Design Performance Analysis of Composite Wings," , 2016.

[7] Werter, N., and De Breuker, R., "A novel dynamic aeroelastic framework for aeroelastic tailoring and structural optimisation," *Composite Structures*, Vol. 158, 2016, pp. 369–386. https://doi.org/https://doi.org/10.1016/j.compstruct.2016.09.044, URL https://www.sciencedirect.com/science/article/pii/S0263822316318669.

[8] Kiviaho, J. F., Jacobson, K., Smith, M. J., and Kennedy, G., *A Robust and Flexible Coupling Framework for Aeroelastic Analysis and Optimization*, 2017. https://doi.org/10.2514/6.2017-4144, URL https://arc.aiaa.org/doi/abs/10.2514/6.2017-4144.

[9] Yildirim, A., Jacobson, K. E., Stanford, B. K., Gray, J. S., Mader, C. A., Martins, J. R. R. A., and Kennedy, G. J., "MPhys: A Modular Multiphysics Simulation Package Using the OpenMDAO Framework," *AIAA Aviation Forum*, 2020.

[10] Van Schie, S. P. C., Warner, M. A. P., Fletcher, A. H., and Hwang, J. T., "Modular, Consistent And Conservative Aeroelastic Coupling For Large-Scale Multidisciplinary Design Optimization," *AIAA SciTech 2024 Forum*, 2024.

[11] Brooks, T. R., Martins, J. R., and Kennedy, G. J., "High-fidelity aerostructural optimization of tow-steered composite wings," *Journal of Fluids and Structures*, Vol. 88, 2019, pp. 122–147. https://doi.org/https://doi.org/10.1016/j.jfluidstructs.2019.04.005, URL https://www.sciencedirect.com/science/article/pii/S0889974618306340.

[12] Hao, P., Yuan, X., Liu, C., Wang, B., Liu, H., Li, G., and Niu, F., "An integrated framework of exact modeling, isogeometric analysis and optimization for variable-stiffness composite panels," *Computer Methods in Applied Mechanics and Engineering*, Vol. 339, 2018, pp. 205–238. https://doi.org/https://doi.org/10.1016/j.cma.2018.04.046, URL https://www.sciencedirect.com/science/article/pii/S0045782518302317.

[13] Sarojini, D., Ruh, M. L., Yan, J., Scotzniovsky, L., Orndorff, N., Xiang, R., Zhao, H., Krokowski, J. J., Warner, M., van Schie, S. P., Cronk, A., Guibert, A., Chambers, J. T., Wolfe, L., Doring, R., Despins, R., Joseph, C., Anderson, R. M., Ning, A., Gill, H., Lee, S., Cheng, Z., Cao, Z., Mi, C., Meng, S., Silva, C., Chen, J.-S., Kim, A., and Hwang, J. T., "Review of Computational Models for Large-Scale MDAO of Urban Air Mobility Concepts," *AIAA SciTech 2024 Forum*, 2024.

[14] Gill, P. E., Murray, W., and Saunders, M. A., "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization," *SIAM Review*, Vol. 47, No. 1, 2005, pp. 99–131. https://doi.org/10.1137/S0036144504446096, URL https://doi.org/10.1137/S0036144504446096.

[15] Scroggs, M. W., Dokken, J. S., Richardson, C. N., and Wells, G. N., "Construction of Arbitrary Order Finite Element Degree-of-Freedom Maps on Polygonal and Polyhedral Cell Meshes," *ACM Trans. Math. Softw.*, Vol. 48, No. 2, 2022. https://doi.org/10.1145/3524456, URL https://doi.org/10.1145/3524456.

[16] Xiang, R., van Schie, S. P., Scotzniovsky, L., Yan, J., Kamensky, and Hwang, J. T., "Automating adjoint sensitivity analysis for multidisciplinary models involving partial differential equations," , 2024. In review.

[17] Orndorff, N., Wang, B., Ruh, M., Fletcher, A., and Hwang, J., "Gradient-based sizing optimization of power-beaming-enabled aircraft," 2023. https://doi.org/10.2514/6.2023-4019.

[18] Ruh, M., Liu, X., Yu, R., and Hwang, J., "Airfoil shape parameterization using reconstruction-error-minimizing generative adversarial networks," 2023. https://doi.org/10.2514/6.2023-3722.

[19] Hilger, J., and Ritter, M. R., "Nonlinear Aeroelastic Simulations and Stability Analysis of the Pazy Wing Aeroelastic Benchmark," *Aerospace*, Vol. 8, No. 10, 2021. https://doi.org/10.3390/aerospace8100308, URL https://www.mdpi.com/2226-4310/8/10/308.

[20] Beran, P., Strganac, T., Kim, K., and Nichkawde, C., "Studies of Store-Induced Limit-Cycle Oscillations Using a Model with Full System Nonlinearities," *Nonlinear Dynamics*, Vol. 37, 2004, pp. 323–339. https://doi.org/10.1023/B:NODY.0000045544.96418.bf.