



SYNTHESISING COMPUTATIONAL DESIGN METHODS FOR A HUMAN-CENTRED DESIGN FRAMEWORK

[Authors will be inserted automatically]

[Institutions will be inserted automatically]

[Corresponding author data will be inserted automatically]

Abstract

[Abstract will be inserted automatically]

[---]

[---]

[---]

[---]

[---]

[Keywords will be inserted automatically]

1. Introduction

While recent advances in computational design methods have yielded many interesting results, particularly in the field of architectural design, there remains significant limitations in terms of generalizable methods for specific human-centred design tasks (Cagan et al., 2005). Considering the two main subgroups of computational design - optimisation and generative methods - both have distinct operational outputs and applications. Though there is significant overlap between these methods, there is a lack of coherent understanding of how they can be synergised for more focused adoption within design practice.

A major gap lies in trying to bridge the gap between abstract subjective human-factors - which have been detailed extensively by studies in human-centred design (Giacomin, 2015), design semantics and design emotion (Demirbilek & Sener, 2010) - and the fundamentals of optimisation and efficiency within the objective space of engineering mechanics. Computational methods have been used for some time with successful results within architectural design (see Caetano et al., 2020 for an overview) though there is less obvious output within industrial design possibly for reasons related to localised complexity and specific use-case requirements.

This paper will seek to explore this issue in greater depth by examining the different cultures within the field of computational design and how they can be reformulated for a general human-centred design framework. Firstly, some background around the topic will be provided to provide a foundation for the discussion and introduce the core concepts and themes relating to computational design. Secondly, we will consider how the different forms of computational design can be distinguished; what the overlaps are and what the distinctions are between the various methods along with exploring their distinct applications. Thirdly, bringing these themes together, the discussion will shift to address how the distinctions between the computational methods introduces space in which generalizable

methodological models can be developed in which computational methods can be formulated around "macro level" or "micro level" design approaches. Lastly, the differentials between these models are examined whereby the key assumptions, outputs and tools that define the approaches are made plain, with recommendations for the most effective human-centred computational design tools.

2. Distinguishing forms of computational design

2.1. Overview and history

For the purposes of this paper, "computational design" will function as an umbrella term for the kinds of design activity that involve computation of some kind. While there are many terms that could be reasonably used as a catch-all description, "computational design" is sufficiently broad to encompass the various forms we wish to explore. Here we will provide an overview of the different form of computational design taking the range of processes known as *optimization* as our starting point.

Optimization has become one of the most widely used tools for engineers undertaking civil and mechanical design work. The processes have evolved over recent decades into lots of different variations though some of the most interesting outputs within a design context have emerged in the last ten or twenty years. The advent of advanced visual programming languages such as Rhino-Grasshopper and much better accessibility to additive manufacturing technologies has beckoned a new era of computational design. "Rock star" designers such as Neri Oxman, Thomas Heatherwick and the late Zaha Hadid have all applied computational methods within their respective industrial design and architectural projects (see Oxman, 2010; 2012 for instance) accelerating and stimulating its uptake in other domains. While the processes utilised will differ on a case-by-case basis with discrete problems requiring distinct approaches, this truism does not hamper us in mapping a general understanding of optimization with a view to advancing a general understanding of computational design methods.

The influential work in optimization theory by Bendsøe and Kikuchi's (1988) describes the basic problem to be solved; how should material be laid-out within a given space envelope? Indeed, they describe three categories of structural optimization a) sizing optimization where the basic design in terms of connectivity of elements b) shape optimization, where the optimization is engineered around a fixed topology - in the 2D case this means that a set of 2D areas or domains has fixed topology but changeable boundaries within the plane and c) topological optimization where only the domain loads and displacement constraints are specified and material may be freely assigned to each location in the design domain (2D or 3D), (see Fig 1). These methods get more advanced in succession with a topological optimization generated from an understanding of design function, as Bendsøe and Kikuchi's (1988, p.2) describe: "The topology, shape, and size of the structure are not represented by standard parametric functions but by a set of distributed functions defined on a fixed design domain. These functions in turn represent a parametrization of the stiffness tensor of the continuum and it is the suitable choice of this parametrization which leads to the proper design formulation for topology optimization."

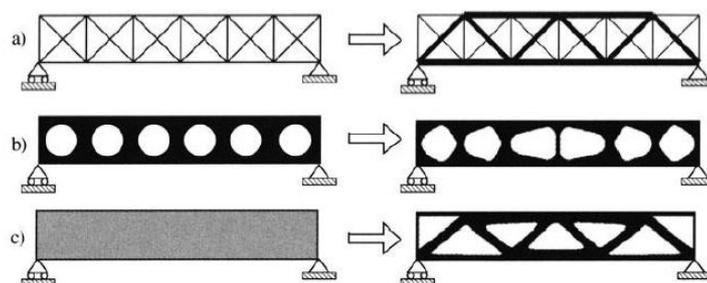


Figure 1. The initial problems are shown on the left with the optimised solutions shown on the right - a) sizing optimization b) shape optimization c) topological optimization. From Bense and Sigmund (2004)

A topological optimization of material is strictly a mathematical method that can change material layout to satisfy design goals. For example, structural compliances can be established within an envelope of geometry, combined with information regarding desired displacement goal which may relate to a desired

function, an “optimized” structure can be generated. The core benefit of these methods is the saving of material while achieving specified structural properties such as strength or rigidity in the design. For this reason, the methods are widely used within particular domains such as architectural engineering, automotive and aerospace (Yildiz, Kaya et al. 2004).

Sigmund and Maute’s (2013) summaries are instructive for defining the varieties of spatial optimization tools and approaches, these are described in Table 1 below. Each method approaches the optimization in different ways though all are built around similar mathematical frameworks. Density and evolutionary approaches use nodal-based design variables, differing from level set methods which are often used in conjunction with topological derivative methods. Hybrid methods are becoming more widely applied such as the combination of level set and shape derivatives (Yamada, Izui et al. 2010). Digital parameterization of geometry allows for the systematic analysis of the structural elements where it can be decomposed into finite elements.

Table 1. Summary of optimization methods

Density based methods: the process involves splitting a structure down into microscale voids and optimising material distribution (density) based on given design constraints. The method is highly developed and notable variants include the Simplified Isotropic Material with Penalization (SIMP) and Rational Approximation of Material Properties (RAMP) approaches (Bendsøe and Sigmund 1999)
Evolutionary approaches: finite element analysis is utilised to train optimization software in an evolutionary fashion to follow particular material distribution paths (Mattheck and Burkhardt 1990)
Topological derivative methods: applications of functional shape derivatives with respect to microscale changes in shape topology, such as adding small defects such as seeding points or infinitesimal holes (Sokolowski and Zochowski 1997)
Level set methods: the structure under optimization is implicitly represented by a moving boundary embedded in a scalar function (known as the “level set” function) of a higher dimensionality. The method is flexible in handling complex topological change (Wang et al. 2003)
Phase field methods: method developed as a way to represent the surface dynamics of phase transition phenomena such as solid-liquid transitions. By utilising the approach, perimeter control can be implemented enabling optimization (Bourdin and Chambolle 2003)

Each of these methods have various benefits and drawbacks that are discussed extensively within the research literature. For instance, the density based methods are well-developed but the algorithmic sophistication is more limited compared to level set or evolutionary methods. Topological derivative and phase field methods, while quite sophisticated are fairly limited in their usefulness and possible applications.

Chronologically, generative design grew out of the work on engineering optimisation that preceded it. Critical work such as Bendsøe and Sigmund (1999; 2004) formulated processes by which static structures could be optimised within a set of design constraints. As the reader can doubtless see, the optimization tools that were principally developed in order to explore distributions of material framed strictly within the context of engineering efficacy, doubled (somewhat unintentionally) as a more broad-brush design tool that could be used to generate elaborate geometric designs such as the bridge formations shown in Figure 1. While the purpose of this paper is not to go into a detailed review of the backstory of computational design methods, it is valuable at this juncture to explore how more recent developments have birthed interesting cultural artefacts within commercial design, art and academic research. All of which will be instructive for establishing how best a generalizable methodology can be conceived that combines the two "cultures" of human factors and engineering optimization.

2.2. Applications and culture

So far we have seen how computational design methods have been utilised to develop optimized static and solid structures, emerging work is using the methods for other creative pursuits such as flexible wearable products and art sculptures. While this is still an emerging area of practice and research, there

are a number of interesting examples to which we can refer. Prominent industrial designer and researcher Neri Oxman (2010; 2012) for instance has developed a set of tools utilising “digital morphogenesis” methods. These methods utilise a biological approach to form building, integrating biological processes of growth into algorithmic generation tools. Her prototype chaise lounge dubbed “Beast” (Fig 2a) is locally modulated for both structural support and corporeal aid through the adaption of material density, stiffness, and flexibility. It is also worth noting the overall aesthetic that is generated by generative design efforts; as well as the increased functional articulation, the aesthetic created by the fusion of organic and biological forms with modern production processes is notably striking (see Fig 2b, organic sculpture by Neri Oxman). This has been discussed directly in an article by Rain Noe, (2019) writing for Core77 and has cited the creation of computationally optimized car tyres as a notable output of the culture.

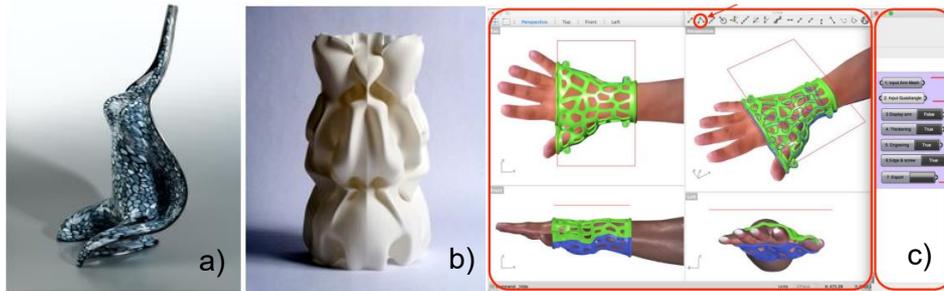


Figure 2. “Beast” chaise lounge by Neri Oxman (a), generated organic sculpture by Neri Oxman (b) images from Wikimedia Commons, Voronoi optimisation of orthotic utilising Grasshopper (from Li & Tanaka, 2018) (c)

Other prominent areas of research are the production of prosthetics whereby generative methods have been employed to explore more interesting aesthetics that allow a user to feel more comfortable while retaining full functionality of the prosthetic. In this sense, computation is being utilised for another core human-factors dimension, that of aesthetic experience, product identity and design emotion (following Desmet & Hekkert, 2007). Some researchers have used generative design methods to develop more reliable prosthetics designs for amputees (*removed for peer review*). This is still an experimental area of medicine, and it is not yet hugely widespread, but research work in this area has expanded recently. Some interesting examples have included work from Rajput et al., (2021) who’s approach aimed to create an optimised design for a lower-leg prosthetic utilising form finding optimisation algorithms. The work resulted in an alternative design for a foot and calf prosthetic which has an integrated compliant mechanism to create the required localised flexibility. Similar principles have been explored by a range of researchers. Writing in a volume of *Advanced Structured Materials*, Hermida-Ochoa et al., (2021) have presented a strategy for creating personalised prosthetic components through anatomical scanning that are subsequently fabricated using 3D printing. Additionally, Beltrán-Fernández et al., (2021) utilised mechanical analysis to optimise the design of an orthosis for clubfoot. The analysis was able to identify key areas of stress and strain in the design and hence optimize the material distribution during the 3D printing fabrication process. A strategy similar to this is employed by Li and Tanaka (2018) who use Rhino-Grasshopper programming to produce bespoke orthotics designed to immobilise fractures in hands. The paper illustrates a proof-of-concept for using a Voronoi optimization approach for the creation of bespoke structures attuned around user anatomy (see Fig 2c).

We can see that the different research efforts have utilised a different array of computational tools with many relying on more traditionally understood optimization methods as opposed to methods that incorporate other human-factors. While this is perhaps indicative of the particular methodological constraints that computational tools present, it is a clear knowledge gap in which more study and analysis could prove valuable for research and practical innovation.

The key factor missing in these discussions is how this could be translated into a generalizable model for human-factors led computational design. While the overarching goals of the various projects are different, they are all interested in the interfacing between form and human ergonomics with the caveat that advanced additive manufacturing methods facilitate the production of their complex and ornate

parts. Though they also use different methods to produce the products, there are clear areas in which an overarching approach can be defined.

2.3. Conceptual opportunities in human-centred design

So where can this discussion take us? The key question is how the panoply of computational design methods can be reined-in to focus their output on questions of human-factors. We have seen how optimization tools can be incredibly powerful in determining a broad sweep of functional design solutions usually with respect to material distribution. Additionally, by contrast, we have also explored examples that use form building algorithms to create art in the form of sculpture. The intersection of these distinct cultures has been explored in emerging areas of research that seek to focus on factors such as ergonomics or the creation of distinct aesthetics. Some initial practical explorations by the authors for instance have explored how computation can be used in a design context to generate both interesting aesthetic outputs and interesting functional outputs considering the respective examples of a trumpet horn and plastic bottle shape (Figure 3). The forms were explored using Rhino-Grasshopper solvers.

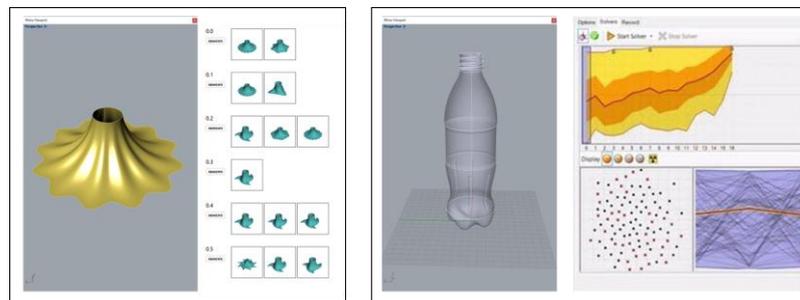


Figure 3. Generative approaches for aesthetic (left) and functional (right) design options

Here is a clear practical example of evaluating the subjective properties of form in a systematic and guided way. Conceptually, this is an evolution from the default culture of material-space optimizations and opens new space in which creative design work can operate. This sets the scene for the next section which will continue to evolve these ideas and establish a framework for human-centred computational design.

3. Towards a framework for human-centred computational design

When developing a variant framework to the computational design methods we have been examining, it is important to consider how such a process may differ to those more traditionally understood design methodologies and philosophies. Many well-known and widely used methodologies such as the British Design Council's "double diamond", Ulrich and Eppinger's methods (1994) or Pugh's earlier matrices-based methods (1990) all have a variety of benefits and draw backs for different design contexts. The double diamond for instance is high-level and generic providing the design team lots of freedoms within the approach. By contrast, Pugh's methodology is much more prescriptive and focuses the design work by applying sets of constraints around various elements such as materials and costs.

Computational design methods use the powers of algorithmic or mathematical logic to explore "design space". While this space is intentionally configured through the definition of constraints, many aspects of designer agency are removed in favour of a computational method that systematically interrogates the possible geometric formations that fit to the predefined constraints (Bensoe and Sigmund, 2004). In this respect, the process is fundamentally different to the default hylomorphic framing of form-emergence (see Ainsworth, 2016 for a summary). When trying then to define how to understand the overall processes of computational design for practical usage it is useful to subdivide the activities. By drawing an equivalence to the broad design space constraints set out during the first stages of an optimization; we can term this the "macro" level of the process where the broad architecture of the design is set out. Following this, a "micro" level of the process somewhat equivalent to design detailing whereby the geometry is optimized around a set of ad hoc constraints (e.g., ergonomic requirements) is carried out. This macro-micro bilateral can function as a useful equivalence for the "two cultures"

observation discussed earlier - where traditional optimization methods are more interested in the output at the macro-scale i.e., large structural changes that could say, save large amounts of material, where more modern methods integrating human-factors may require detailing at the micro-scale where the nuances of ergonomics and aesthetic experiences are established.

Shown below in Figure 4 is the methodological-model that has been developed in which we will explore further and use as the foundation for exploring the differentials between the macro and micro levels of computational design thinking. This is the high-level version of the model but it will be used as a starting point for establishing our synthesis between the technical outputs of computational design and the human-factors outputs which we have examined in the previous sections. Narratively, the model follows the rough structure of a traditionally understood design methodology, starting with specification and ideation and ending in a product realisation allowing the reader to grasp the critical points in which computational methods are typically applied. Shown is how computational design thinking intersects with the traditional format for design work. The critical differences are seen as the solutions are being developed and the distinction between the macro solutions and the micro solutions which include a looping structure that integrates knowledge acquired from human-factors analysis and testing that is then fed back into the computational coding definitions. As a general rule, a micro level approach will always include a macro level phase before full realisation.

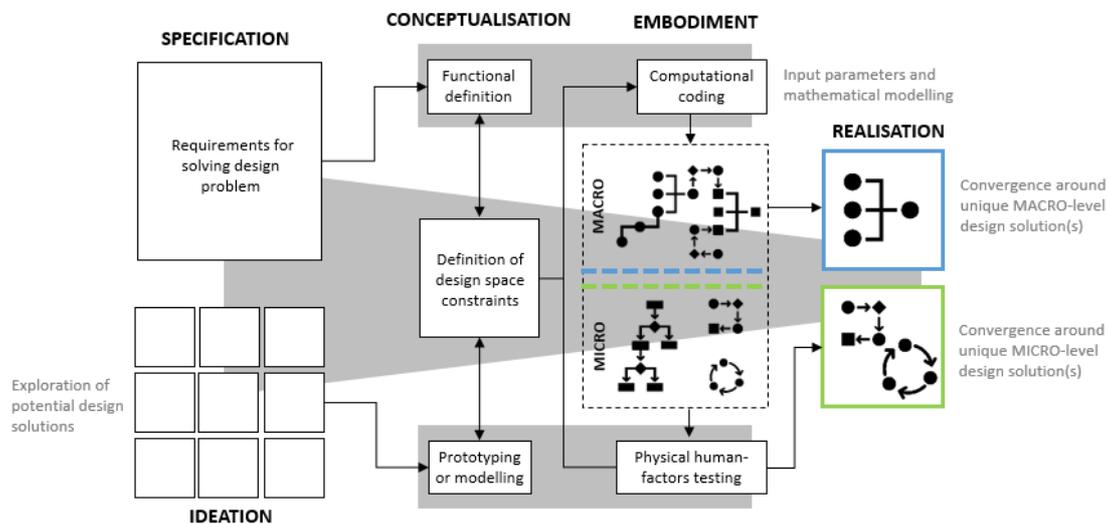


Figure 4. Generative design methodology incorporating macro and micro algorithmic stages

We can start with a more detailed overview of the methodology where the underlying concepts can be described in relation to computational design thinking:

1. **Specification/ideation stage:** *The core requirements for the product are set out within the specifications which run in conjunction to initial ideation. This is perhaps the most well-aligned element to the traditional design methodologies whereby sets of specifications and design concepts would be explored iteratively and potential design solutions are set out.*
2. **Conceptualisation stage:** *The pre-computational phase in which the functionality of the concepts is explored in the abstract and the design space constraints are more properly established. This stage may also include sketch modelling or other forms of prototyping that help to examine which design features need the most computational resources.*
3. **Embodiment stage:** *Computation stage in which the abstract understanding of form and functionality explored in the first two stages become refined and are built into an algorithmic definition. The algorithmic definition has a macro stage detailing the larger structural form features, the micro stage detailing the finer form features and in-built functionality (e.g. compliant mechanisms - see Zhu, 2020).*
4. **Realisation stage:** *Finalisation of design after computational design explorations. The design depending on the macro or micro strategy taken will be full optimised around the specified constraints and any inputted data. At the stage, the design should be ready for manufacturing.*

As shown the macro and micro computational activities can be more effectively deployed at certain design stages and may be deployed iteratively with the aim of achieving a unique design solution. The methods articulate with other practical design activities such as prototyping but the key computational activity is the coding of algorithms that facilitate the generation of the design solutions. Next, we can explore how the differentials and synergies between the macro and micro level strategies can be established in more detail.

3.1. Establishing a synthesis

While this paper has sought to argue that there are two distinct "cultures" of computational design effort, establishing a synthesis between these cultures is vital for understanding how computational approaches can be better presented as part of a more generalizable framework for human-centred design based innovation and indeed advancing efforts in traditional topology optimization practices.

To think further about the differentiations and possible synergies between a macro level approach with a more nuanced micro level approach, areas of the model shown in Figure 4 before can be explored in greater detail. To do this, some of the overarching process narrative has been removed and the critical steps leading to design solutions have been focused on. Critically, three important elements will be focused on that allow for the tracking of the utility of each approach for each context; **assumptions, outputs and tools**. These three elements facilitate both an analysis of function in terms of what the approaches are good for and what tools can be wielded to get particular outcomes, and furthermore cultural concerns around the assumptions required to initiate the computational design process itself - what the designer or engineer imagines the product creation process and the end product to be. Starting with macro level computational approaches, Figure 5 presents an analysis in terms of assumptions, outputs and tools.

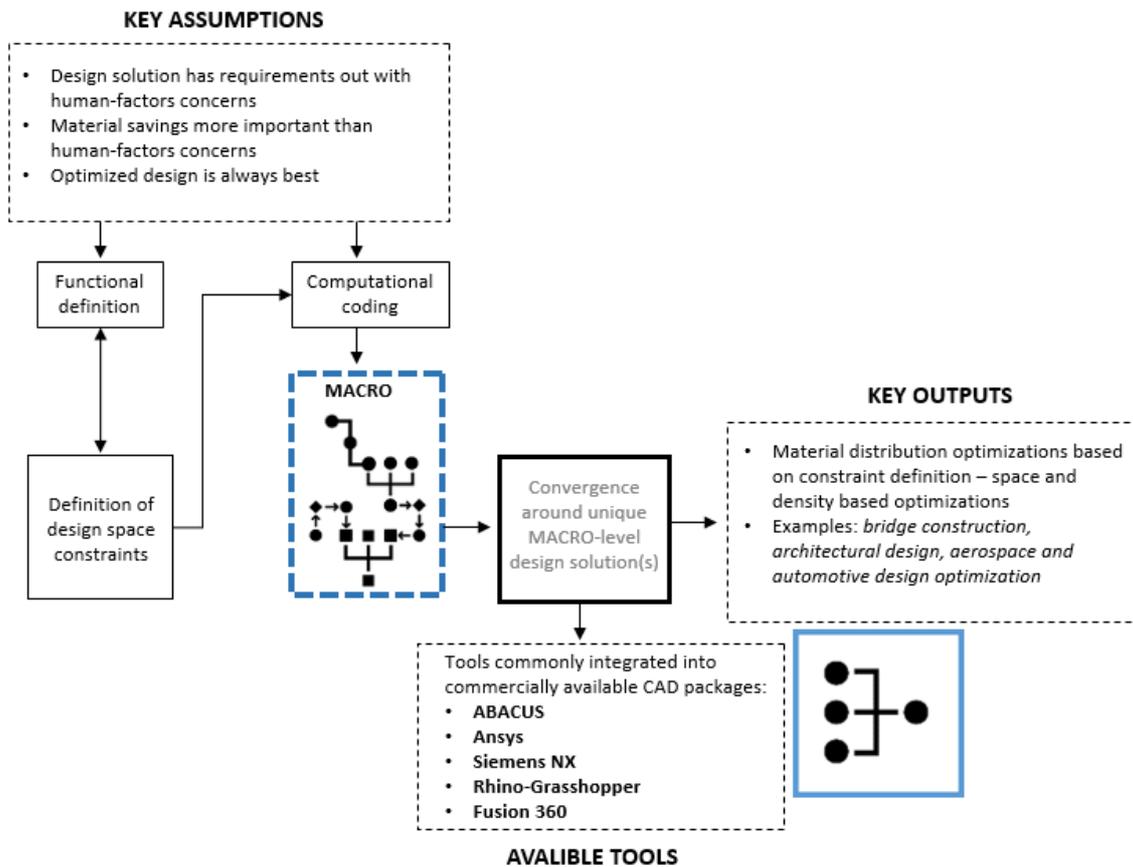


Figure 5. Analysis of Macro-level computational design methods in terms of assumptions outputs and tools

This diagram illustrates how the macro level approaches are linear in their methodological process. Once the design constraints have been adequately defined, the algorithmic optimization packages available in the commercially available software listed within the available tools section can be applied. The critical assumptions for these kinds of approaches partition the design solution from direct human interaction at a subjective psycho-social level and instead focus on material optimization (e.g. space or density distributions) as the critical output. Many of the packages cited as key tools allow the user to define forces acting within the space envelope and define material specifications allowing the structure to be optimized with respect to a definite field of forces. This can now be compared to the micro level approaches which operationalise a range of different factors and allow for the creation of explicitly human-centred products.

Figure 6 maps the methodological model for micro level computation efforts. As shown, the process is much less linear and involves looped exchanges of bespoke data that has been established from empirical human factors analysis - this could be emotional responses to certain form elements (see Bar & Neta, 2006) or ergonomics and usability data derived from physical testing or data capture such as 3D scanning or motion capturing (Paoli et al, 2020 reviewed several relevant technologies). Crucially, this block of bespoke data fundamentally changes the "culture" of the operation as the critical assumption is that subjective psycho-social factors will play a role in how the end product will interact with the human world; the design outcome is not partitioned from direct human influence. The key outputs in this regard are bespoke forms that might be tailored around a specific ergonomic features like anthropometric measurements or predictable emotive responses such as the negative associations most humans place on observed angularity (Bertamini et al., 2016). Notably, the tools cited here are more distinct, making reference to several Grasshopper plug-ins which facilitate these kinds of unique design explorations. The flexibility of Grasshopper makes it an excellent candidate tool for designers wanting to explore the integration of human-factors data into a computational design exploration.

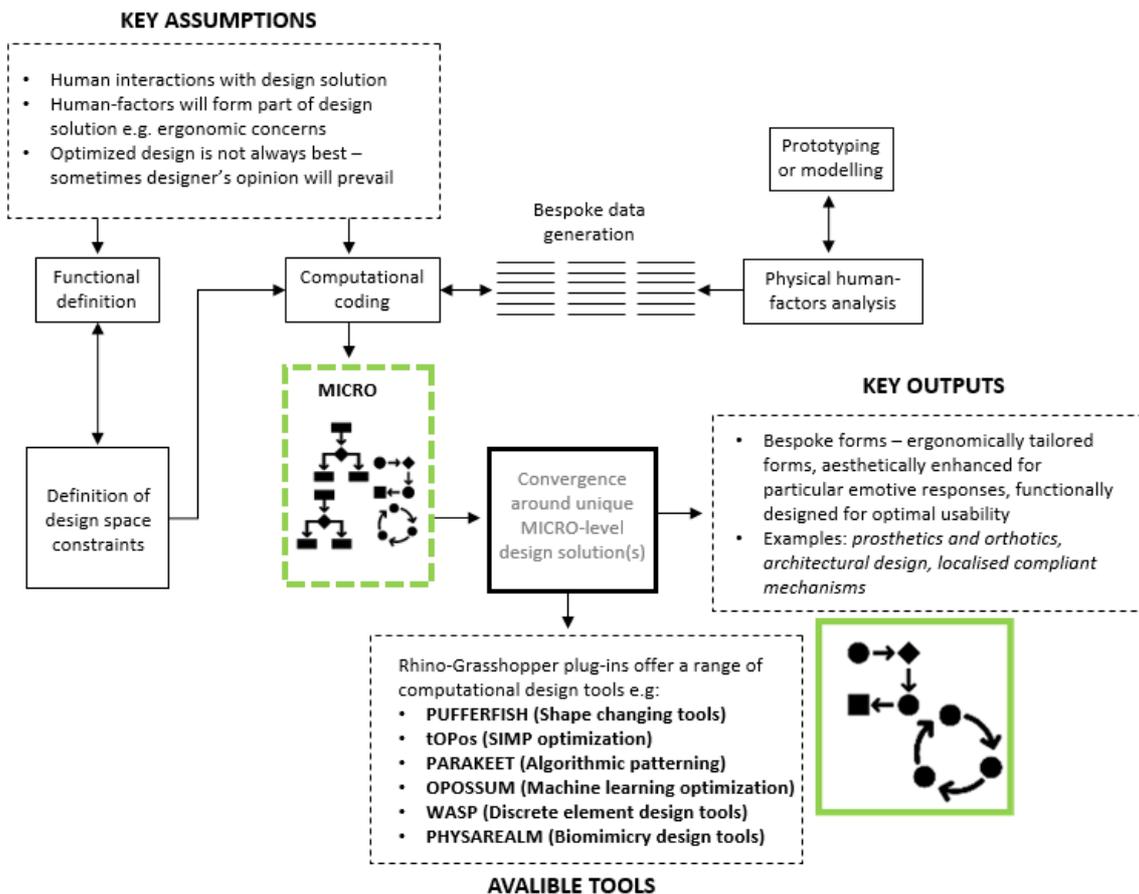


Figure 6. Analysis of Micro-level computational design methods in terms of assumptions outputs and tools

So where can this mode of thinking take us? And how can it be used by design practitioners interested in exploiting computational design methods to explore human factors? The models we have set out in this paper invites further expansion by allowing for the meaningful distillation of the synergies and differences between the two "cultures" of computational design. These can now be outlined but drawing comparisons between the models presented at Figures 5 and 6.

1) Key assumption synergies: the creation of a "unique" form for the design solution. For the macro level this is a by-product of the specific material and weight constraints, the micro level may be specifically guided by the articulations between materials, functionalities and human interaction concerns. 2) Key output synergies: material savings and functionality optimisation with a given space envelope. Both macro level and micro level approaches are engaged with the problem of material-space arrangements and how best to optimize with respect to particular constraints to achieve the desired functionality. While different assumptions are at play in the processes in general, these particular outputs align closely. 3) Rhino-Grasshopper visual programming language. The programming language facilitates a wide-array of analysis and design generation tools and allows for both macro and micro level computation efforts. In terms of human-centred design and human factors analytics, Grasshopper is an excellent tool and has been explored within many contexts to create interesting design generation workflows that integrate human-factors data (Abualigah & Diabat, 2020). Furthermore, for a human-factors and computation integration focus, Grasshopper presents a very promising architecture in which to define a generalizable framework for approaching these design problems.

4. Conclusions

This paper has been theoretical in nature and has proposed a set of arguments that differentiate two "cultures" of computational design efforts. One culture - the "macro"- focuses on technical engineering problems of material and space optimization where a definite outcome is desirable based on a set of defined constraints. By contrast, the "micro" culture focuses on the integration of complex human interaction problems and may also explore the subjective realms of psychological experience as an input to the algorithmic solvers.

These arguments were developed by reviewing a number of examples within the research literature and also by looking at prominent examples within design practice that have integrated complex human-factors data like anthropometrics for innovative generative solutions. A number of models were presented that set out this differentiation and identified the macro cultures with a more linear approach that is less iterative in algorithmic solving due to the design space being smaller. The micro level culture that is required for human-centred computational design presents different challenges and will often utilise completely bespoke data and physical testing to build in the design constraints and improve the solutions iteratively. The intrinsic complexity and subjectivity of human-factors data means that the process is somewhat less linear than the traditional material distribution problems.

By comparing these models, several synergies between the approaches were identified where alignments between key assumptions, outputs and tools were clear. Notably, Rhino-Grasshopper was identified as a critical tool for a human-centred design focus due to its flexibility in the integration of many kinds of data types. This synthesis furthermore allows for a better theoretical understanding of the different computational design cultures and may be applied by researchers and practitioners to more effectively steer their thinking on computational design problems.

5. References

- Abualigah, L.M., & Diabat, A.T. (2020). A comprehensive survey of the Grasshopper optimization algorithm: results, variants, and applications. *Neural Computing and Applications*, 1 - 24.
- Ainsworth, T. (2016). Form vs. matter. In E. N. Zalta (Ed.), *The Stanford encyclopaedia of philosophy (Spring 2016 Edition)*.
- Bar, M., & Neta, M. (2006). Humans prefer curved visual objects. *Psychological Science*, 17(8), 645–648. doi: 10.1111/j.1467-9280.2006.01759.x
- Beltrán-Fernández J.A., Andrade J.L.C., Ochoa J.C.H., Gómez L.H.H., Uribe-Cortés T.B., Garibaldi P.M. (2021). *Numerical–Experimental Study of 3D Printed Ortheses for Rehabilitation of Patients with Musculoskeletal*

- Lesions*. In: Beltran-Fernandez J.A., Öchsner A. (eds) Design and Simulation in Biomedical Mechanics. Advanced Structured Materials, vol 146. Springer, Cham. https://doi.org/10.1007/978-3-030-65983-7_8
- Bendsøe, M. P. and N. Kikuchi (1988). Generating Optimal Topologies in Structural Design Using a Homogenization Method. *Computer Methods in Applied Mechanics and Engineering*, Holland.
- Bendsøe, M. P. and O. Sigmund (1999). Material interpolation schemes in topology optimization. *Archive of Applied Mechanics*. 69, 635–654 (1999). <https://doi.org/10.1007/s004190050248>
- Bendsøe, M. P. and O. Sigmund (2004). *Topology Optimization Theory, Methods, and Applications*. NY, USA, Springer.
- Bertamini, M., Palumbo, L., Gheorghes, T. N., & Galatsidas, M. (2016). Do observers like curvature or do they dislike angularity? *British Journal of Psychology*, 107(1), 154–178. doi: 10.1111/bjop.12132
- Bourdin, B. and A. Chambolle (2003). Design-dependent loads in topology optimization. *ESAIM: Control, Optimisation and Calculus of Variations*. 9: 19-48.
- Caetano, I., Santos, L., & Leitão, A.M. (2020). Computational design in architecture: Defining parametric, generative, and algorithmic design. *Collection of Frontiers of Architectural Research*, 9, 287-300.
- Cagan, J., Campbell, M.I., Finger, S., & Tomiyama, T. (2005). A Framework for Computational Design Synthesis: Model and Applications. *J. Comput. Inf. Sci. Eng.*, 5, 171-181.
- Demirbilek, O., & Sener, B. (2003). Product design, semantics and emotional response. *Ergonomics*, 46, 1346 - 1360.
- Desmet, P., & Hekkert, P. (2007). Framework of product experience. *International journal of design*, 1(1).
- Giacomin, J. (2014). What Is Human Centred Design? *The Design Journal*, 17, 606 - 623.
- Hermida-Ochoa J.C., Beltrán-Fernández J.A., Andrade J.L.C., Hernández-Gómez L.H., Uribe-Cortés T.B., Moreno-Garibaldi P. (2021). *Tridimensional Design and Printing Techniques to Obtain Personalized Prosthetic Components for Specific Cases Involving Bone Defects*. In: Beltran-Fernandez J.A., Öchsner A. (eds) Design and Simulation in Biomedical Mechanics. Advanced Structured Materials, vol 146. Springer, Cham. https://doi.org/10.1007/978-3-030-65983-7_7
- Li, J., & Tanaka, H. (2018). Rapid customization system for 3D-printed splint using programmable modeling technique – a practical approach. *3D Printing in Medicine*, 4.
- Mattheck, C. and Burkhardt, S. (1990). A new method of structural shape optimization based on biological growth. *International Journal of Fatigue*. 12, 185-190. [https://doi.org/10.1016/0142-1123\(90\)90094-U](https://doi.org/10.1016/0142-1123(90)90094-U)
- Noe, R. (2019). *Where Does Generative Design Belong? Designers Must Decide* [online], Core 77. Available at: <https://www.core77.com/posts/89318/Where-Does-Generative-Design-Belong-Designers-Must-Decide>
- Oxman, N. (2010). *Material-based design computation*. MIT Press, US
- Oxman, N. (2012). Material computation. In B. Sheil (Ed.), *Manufacturing the bespoke: Making and prototyping architecture* (pp. 256–265). Hoboken, NJ, USA: John Wiley & Sons, Inc.
- Paoli A, Neri P, Razionale AV, Tamburrino F, Barone S. (2020). Sensor Architectures and Technologies for Upper Limb 3D Surface Reconstruction: A Review. *Sensors*. 20(22):6584. <https://doi.org/10.3390/s20226584>
- Pugh, S. (1990). *Total Design: Integrated Methods for Successful Product Engineering*. Prentice Hall, UK
removed for peer-review
- Shuib, S., Ridzwan, M.I., & Kadarman, A.H. (2007). Methodology of Compliant Mechanisms and its Current Developments in Applications: A Review. *American Journal of Applied Sciences*, 4, 160-167.
- Sigmund, O. and K. Maute (2013). Topology optimization approaches. *Structural and Multidisciplinary Optimization*. 48(6): 1031-1055.
- Sokolowski, J. and A. Zochowski (1997). *On Topological Derivative in Shape Optimization*. Lorraine, France. Technopole de Nancy-Brabois.
- Ulrich, Karl & T., Eppinger (1994). *Product Design and Development*. McGraw-Hill Inc., US
- Yildiz, A. R., et al. (2004). Optimal design of vehicle components using topology design and optimisation. *International Journal of Vehicle Design*. 34: 387-398.
- Wang, M. Y., et al. (2003). A level set method for structural topology optimization. *Computer Methods in Applied Mechanics and Engineering*. 192: 227–246.
- Zhu, B., Zhang, X., Zhang, H., Liang, J., Zang, H., Li, H., & Wang, R. (2020). Design of compliant mechanisms using continuum topology optimization: A review. *Mechanism and Machine Theory*, 143, 103622.