I, Matthew C Wild, hereby submit this original work as part of the requirements for the degree of Master of Architecture in Architecture.

It is entitled:
Digital Derivation: the role of algorithms and parameters in building skin design

Student's name: Matthew C Wild

This work and its defense approved by:

Committee chair: Michael McInturf, M.Arch.

Committee member: Ming Tang, M.Arch.
Digital Derivation:

the role of algorithms and parameters in building skin design

A thesis submitted to the Graduate School
of the University of Cincinnati in partial fulfillment
of the requirements for the degree of:

Master of Architecture

in the School of Architecture and Interior Design
by

Matthew C Wild

Bachelor of Arts, Architecture
Ball State University 2012

Committee Chairs:
Michael McInturf
Ming Tang
Abstract

Despite all of the technical advancements in building skin design and generative design computing, there are many questions concerning the method by which computer generated skins should be designed and the meaning behind their shapes. While the seemingly endless possibilities of digital tools have allowed for the fluid patterning and manipulation of surfaces to become an icon of parametricism, they often fail to provide any deeper meaning or correlation between the formal and functional aspects of a building. To the current extent with which the profession has utilized parametrics and algorithmic thought, there is often a lack of depth or significance behind the flashy images of assumed intricacy it produces—falling far short of its rich potential to engage with the real problems, processes, and functions of today. This thesis explores the roles of algorithmic thought and computational methods in building skin design in an effort to establish a larger framework or methodology for the implementation of digital tools.

By exemplifying how to acquire data and use it to inform design decisions, this document aims to shift complexity from the product to the process. Only then can we see the trend of computational design root itself in purpose and meaning and begin to engage with real issues. As a result, the representation of this data in and on buildings may become architecture’s new method of ornamentation—an ornamentation that stands for something beyond the mere image of the final product. Ultimately, this thesis looks to establish a meaningful methodology, guided by larger frameworks of design, that can be referenced by designers looking for help generating ideas for building skins through the use of digital tools.
Acknowledgments

A special thanks to Michael McInturf who helped guide and advise me throughout the process; to Yan Krymsky of Cannon Design’s Yazdani Studio for sharing many of his own technical and theoretical processes; to Katie Heinrich for her workshop on Grasshopper and evolutionary solvers as well as her technical assistance; to Ming Tang for being a great resource of parametric knowledge throughout; and lastly to the entire Grasshopper Community online for their frequent insight and help—I could not have done this without you.
<table>
<thead>
<tr>
<th></th>
<th>01</th>
<th>Why “Skin?” The Modern Importance of the Facade</th>
<th>pg. 14</th>
</tr>
</thead>
<tbody>
<tr>
<td>02</td>
<td>The History of Building Skins An Historical Account of the Curtain Wall</td>
<td>pg. 22</td>
<td></td>
</tr>
<tr>
<td>03</td>
<td>The Digital Age Looking for a Meaningful Methodology</td>
<td>pg. 34</td>
<td></td>
</tr>
<tr>
<td>04</td>
<td>Computational Design Method #1 Solar Radiation</td>
<td>pg. 42</td>
<td></td>
</tr>
<tr>
<td>05</td>
<td>Computational Design Method #2 Views</td>
<td>pg. 50</td>
<td></td>
</tr>
<tr>
<td>06</td>
<td>Computational Design Method #3 Daylight</td>
<td>pg. 58</td>
<td></td>
</tr>
<tr>
<td>07</td>
<td>Computational Design Method #4 Skin Patterns</td>
<td>pg. 66</td>
<td></td>
</tr>
<tr>
<td>08</td>
<td>Final Remarks Closing Thoughts on Data-Driven Design</td>
<td>pg. 78</td>
<td></td>
</tr>
<tr>
<td>09</td>
<td>Design Proposal The “Re-Skining” of the Kroger Building</td>
<td>pg. 82</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Bibliography List of Works Referenced</td>
<td>pg. 84</td>
<td></td>
</tr>
</tbody>
</table>
List of Illustrations

01 - Why “Skin?”

1.1 - http://focusonvogue.wordpress.com/2013/02/02/al-bahar-towers-abu-dhabi/
1.2 - http://ksamedia.osu.edu/sites/default/files/originals/080001179.jpg
1.3 - http://www.designdaily.us/2012/08/photo-courtesy-morphosis-weaving-as.html
1.4 - http://inhabitat.com/germanys-q1-building-boasts-a-dazzling-facade-of-400000-metal-feathers/q1-ed01/
1.5 - http://beijingwatercube.com/
1.6 - http://www.e-architect.co.uk/images/jpgs/paris/docks_de_paris_jm120109_nbhp_3.jpg

02 - History of Building Skins

2.1 - http://40.media.tumblr.com/964d1addaa60076a39a0424f8055d6c/tumblr_n00jvd99zxx1r9xcmto1_1280.jpg
2.3 - http://www.museumsinflorence.com/foto/santa%20maria%20novella/image/facciata.jpg
2.4 - http://www.csupomona.edu/~aehacker/arc362/Florence%20and%20Environments/Images/7557.jpg
2.5 - http://www.dreamstime.com/stock-image-bricks-window-masonry-wall-ancient-architecture-image19812701
2.6 - https://thegardenatoja.files.wordpress.com/2012/09/flickr-3627055350-hd1.jpg
2.8 - http://www.archdaily.com/397949/ad-classic-the-crystal-palace-joseph-paxton/51d5776db3fc4b083400230_ad-classic-the-crystal-palace-joseph-paxton_interior.jpg/
2.9 - http://www.archdaily.com/397949/ad-classic-the-crystal-palace-joseph-paxton/51d4964db3fc4b09e00001cf_ad-classic-the-crystal-palace-joseph-paxton_crystal_palace.jpg/
2.10 - http://chicagology.com/goldenage/goldenage005/

03 - Digital Age

3.1 - http://s-media-cache-ak0.pinimg.com/originals/8f/e5/1b/8fe51b2eb5c11e66cb766facf3ef5e7e.jpg
3.4 - http://www.d3space.org/competitions/gal3_img16_big.jpg

04 - Computational Design Method #1

4.2 - author
4.3 - author
4.4 - author
4.5 - author
4.6 - author
4.7 - author
4.8 - author
4.9 - author
4.10 - author
4.11 - author
4.12 - author
4.13 - author
List of Illustrations (continued)

05 - Computational Design Method #2
5.2 - author
5.3 - author
5.4 - author
5.5 - author
5.6 - author
5.7 - author
5.8 - author
5.9 - author
5.10 - author
5.11 - author

06 - Computational Design Method #3
6.2 - author
6.3 - author
6.4 - author
6.5 - author
6.6 - author
6.7 - author
6.8 - author
6.9 - author
6.10 - author
6.11 - author
6.12 - author
6.13 - author

07 - Computational Design Methods #4
7.2 - author
7.3 - author
7.4 - author
7.5 - author
7.6 - author
7.7 - author
7.8 - author
7.9 - author
7.10 - author
7.11 - author
7.12 - author
7.13 - author
7.14 - author
7.15 - author
7.16 - author
7.17 - author
7.18 - author
7.19 - author
7.20 - author
7.21 - author
7.22 - author
7.23 - author

08 - Final Remarks
8.1 - http://media-cache-ak1.pinimg.com/originals/1c/4b/34/1c4b347bda3cb454145b7c7f20656597.jpg
List of Illustrations (continued)

09 - Design Proposal

9.1 - author
9.2 - Image provided by Kroger
9.3 - http://c1.staticflickr.com/9/8249/8457327711_9adf08f06c_h.jpg
9.4 - author
9.5 - author
9.6 - http://ming3d.com/wordpress/?p=784
9.7 - author
9.8 - author
9.9 - author
9.10 - author
9.11 - author
9.12 - author
9.13 - author
9.14 - author
9.15 - author
9.17 - author
9.18 - author
9.19 - author
9.20 - author
9.21 - author
9.22 - author
9.23 - author
9.24 - author
9.25 - author
9.26 - author

10 - Bibliography

10.1 - author
WHY “SKIN?”

THE MODERN IMPORTANCE OF THE FACADE

Figure 1.1: Al Bahar Towers
Skin [noun]^1
1. an outer layer or covering
2. any integumentary covering, casing, outer coating, or surface layer
3. a sheathing or casing forming the outside surface of a structure

The importance of the building skin in architecture is greater now than it ever has been. As experimentation and exploration of facade systems and technologies rapidly increase, we are seeing the role and capabilities of building skins become increasingly comprehensive and complex. Yet I believe it is imperative to explain the most basic functions of a building skin. First off, the building skin provides protection from external climate conditions—it seeks to provide shelter. It is also responsible for creating privacy as well as transparency—controlling how much and where one can see in or see out. Additionally, the mere existence of a building skin demarcating private property. Lastly, building skins reflect the ideology of their times and culture—giving them incredible social and aesthetical importance.

Terminology
Yet before we go any further, I think it is important that we define a few terms. When I say building skin, I am simply referring to the outermost layer or covering of a building—regardless of its structural, weatherproofing, or protective responsibilities. This differentiates the building skin from the building envelope, which is defined as a physical separator between conditioned and unconditioned environments. While it is possible that a building skin could function as a building envelope by performing as a weather, air, and thermal barrier, the term skin does not necessarily prescribe itself to this specific role (Figures 1.2, 1.3). Rather it is one of the many potential roles the building skin (and the many layers it includes) may address. Thus, my use of the word skin is very similar to (and often interchangeable with) the word facade—which refers directly to the exterior of a building. However, a facade most commonly alludes to a specific side or face of a building (from latin facies/face^5), whereas the term building skin is not limited in this regard and often refers to multiple faces, planes, and surfaces which could potentially encompass an entire building.

Visual Appeal
That being said, I think it is fair to assess that building skins draw more attention than any other building component. And that is what makes them so appealing

---

^6 Ibid, 9.
and important to designers. The skin is essentially the “calling card” of a building and its architect. At a larger scale, skins can even characterize the face of an entire city (think of the influx of sterile curtain walls in the 1970’s that still define our urban environments today). Still, the building skin is the primary contributor to the public’s perception of a building and thus is often the most valued element of construction for clients to spend money on. In short, the skin is often the single greatest visual element of a building and is often emblematic of the building’s ideals or purpose (Figure 1.4).

Beyond its mere visual power, building skin technology is undergoing an extremely fast-paced evolution. According to Christian Schittich, author of *In Detail: Building Skins*, “Everywhere there is an upsurge in experimentation, boundaries are being tested and visual conventions called into questions. More than ever attention is focusing on new materials and concepts.” Because modern construction advancements have allowed for the skin to be independent of the building structure (unlike a Romanesque cathedral, for instance, in which the exterior and structure were one in the same), there is a vast amount of flexibility in both the aesthetic appearances and functional roles of the building exterior. This evolution—coupled with recent technical and digital advancements—is allowing for unprecedented exploration in the potentials for building skins. Unfortunately, many of these advancements strictly

---

8 Ibid.
Why “Skin?”

Figure 1.4: JSWD Architetken’s Q1 building in Essen, Germany uses nearly 400,000 metal “feathers” to protect the building from the sun. These feathers embody the environmentally conscious ideals of the construction and serve to express them publicly.

Figure 1.5: Beijing National Aquatics Center
emphasize the visual aesthetic of the building skin and often fall short of any greater significance (Figure 1.5). Building exteriors are constantly at risk of being relegated to nothing more than superficial, “attention-seeking packaging” by designers. However, the skin is beginning to take on a more complex role.

Environmental Performance

With today’s increased awareness of carbon dioxide emissions and the shortage of fossil fuels, architects are focusing more and more on reducing the energy consumption of buildings—both for environmental and economic reasons. And what is the single greatest element in conserving energy for buildings? The skin! Sense the advent of air conditioning systems, it has become common for architects to view the exterior skin as something simply separating the building interior from the climate outside. Now architects are viewing it as a complex multi-layered system that interacts with a multitude of energy flows (such as sun, glare, light deflection, heat, cold, rain, wind, etc) in an effort to minimize costs and energy consumption while simultaneously achieving indoor comfort. In fact, “All comfort-related parameters – with the exception of relative indoor humidity – can be directly controlled and regulated through the design of the facade and the roof and this is the principal guiding factor in the conception of the building skin.” By allowing the environmental components to drive the aesthetic appearance, architects can fully engage with contemporary facade design without falling prone to “superficial ornamentation.”

Re-Skin

With so much emphasis being placed on improving the energy efficiency of our built environment, it’s no surprise there is also a strong desire to update existing structures in today’s economy. Instead of building an entire structure from scratch, building owners are saving money, time, resources, and the environment by hiring architects to “re-skin” outdated or deteriorating buildings. Despite having a completely functional structure, many older buildings lack weather barriers such as a thermal break to prevent heat from leaking in during the summer and out during the winter—resulting in extremely inefficient and expensive energy consumption. By taking existing structures and re-skinning them, architects give new life to dilapidated (and even abandoned) buildings, cut down on energy consumption, and make cities more beautiful by giving their outdated and rundown facades a face-lift (Figures 1.6 and 1.7). And all of this can be achieved with one building component—the skin.

11 Ibid.
12 Ibid.
14 Ibid.
Architects Jakob + MacFarlane gave second life to this 1907 warehouse built in Paris by upgrading the exterior with a new, energy efficient skin.
Conclusion

In light of all this, it is no doubt that building skins are of utmost importance in the field of architecture today. Whether it is visual, explorative, conceptual, performative, environmental, or economical, building skins are capable of doing more now than ever before. As Schittich states in his book: “Given all these possibilities, the topic of ‘building skins’ is as fascinating today as it has rarely been in the history of architecture. Wherever we look, we encounter unbounded joy in experimentation: testing boundaries, querying traditional perceptions, searching for new materials and concepts.”\(^\text{15}\)

And that is exactly what this thesis aims to do.

THE HISTORY OF BUILDING SKINS

AN HISTORICAL ACCOUNT OF THE CURTAIN WALL'S DEVELOPMENT
History of Building Skins

The discussion of building skins today must start with a brief history of their origination. To do so, we must go all the way back to the earliest forms of architecture, as famously covered by German architect Gottfried Semper in his mid 19th century book, “Style in the Technical and Tectonic Arts.” Semper believed that an animal pen composed of branches and twigs was most likely the beginning of the wall and, consequently, the designation of architectural space. In studying the origination of the built environment, Semper divides architecture into two major elements: load-bearing structure and cladding. Interestingly, this theory on the distinction and separation of structure and cladding is still incredibly pertinent to the conversation of building skins today.

The Desire for Ornamentation

In its earliest forms, the exterior wall of a building was always a part of the building’s load-bearing structure. Exterior building skins were almost exclusively driven by this function as well as a few others (protection, privacy, enclosure, etc). However, much like clothing, it wasn’t long before building skins also became subject to style and ornamentation. Intricate levels of external decoration are evident in ancient monuments found all across the world (Figure 2.2). Soon an importance in how buildings presented their exterior faces to urban spaces became a key component of architecture. This was brought to a whole new level in the Renaissance through the conception of decorative facades that were placed in front of existing buildings for one purpose—attractive packaging (Figures 2.3, 2.4). I believe this desire for external beauty and ornamentation has led to a focus on facade design that has dominated the field of architecture for hundreds of years.

The Liberation of the Building Skin

One of the principle components in facade design is the relationship between the window (open surface) and the wall (closed surface). Going back to the age of antiquity, the size of windows and openings were typically very small. This was a result of the immense difficulty in cutting openings out of thick stone or clay walls—especially with the limited construction methods available—as well as the structural necessity for these elements to be load-bearing. Smaller openings would also limit the ease of unwanted environmental elements (sun, wind, rain, snow, etc) and intruders (both humans and animals) from entering a shelter. This was especially necessary before glass became a widespread commodity.

2 Ibid.
3 Ibid.
4 Ibid.
5 Ibid.
6 Ibid.
Figure 2.2: The Parthenon in Athens, Greece is one of the earliest examples of exterior decoration with its ornate frescos.

Figures 2.3 and 2.4: Facade addition (1458-70) by Leon Battista Alberti to the existing medieval church, Santa Maria Novella (1246-1360) in Florence, Italy.

Figure 2.5: Ancient brick window
Figure 2.6: Notre Dame Cathedral (1163-1345) in Paris, France utilizes external flying buttresses and internal ribbed vaulting to allow for large openings on the exterior.

Figure 2.7: Transverse section of a typical gothic cathedral with flying buttresses
With the advancement of building techniques and glass manufacturing in the Middle Ages, larger openings on building facades become possible. For example, Gothic cathedrals revolutionized the way buildings were structured in order to create larger windows that brought more light into the interior. By creating structural, stone skeletons composed of ribbed vaulting, pillars, and flying buttresses, Gothic architecture helped minimize the forces acting on load-bearing walls—allowing them to be thinner with much larger voids (Figure 2.6, 2.7). The result was an appreciation and manipulation of light in interior spaces that was not previously possible. Although not yet completely free, this was one of the first major steps in liberating building skins from the constraints of the load-bearing wall. Unfortunately, the cost of these construction techniques, as well as the glass itself prevented this type of architecture from becoming commonplace until the industrial revolution.

The Industrial Revolution of the 19th century brought about new materials and production methods that completely changed the built environment. According to Schittich: “iron and glass conquered architecture.” This is most evident in Joseph Paxton’s Crystal Palace, built in 1851 for the World’s Fair in London (Figures 2.8, 2.9). Paxton, a gardener with experience building greenhouses, ignored the formal traditions of architects and let the building be dictated entirely be the necessary fabrication and assembly techniques. The result was a beautifully light and transparent building composed of over 300,000 sheets of glass. The Crystal Palace awed its visitors and soon all of Europe was constructing these iron and glass structures. Exhibition buildings, terminals, and shopping arcades all began to emerge with this style. Eventually architects gave up on their decorated facades and began to partake in this revolution.

---

9 Ibid.
10 Ibid.
Figure 2.10: Marshall Field Store (1885-87) in Chicago hides its steel structure behind a classical masonry facade.

Figure 2.11: Carson, Pirie, Scott Building (1899) by Louis Sullivan offers a clear distinction between structural and non-structural elements on the facade.
By the mid 19th century, iron was consistently being used to produce skeleton structures that designated much less load-bearing responsibility to external walls—allowing for more openings. Unlike the Gothic cathedrals, this was a much lighter, simpler, and economical process that permeated into other urban typologies. These metal frames were also desirable because they produced open and flexible interior spaces, void of any internal structural walls. But the innovation didn’t stop there. Economic growth in the United States during the 1870s unleashed an unprecedented boom in both construction and construction advancements. The advent of elevators and structural steel, coupled with rising real-estate values in cities led to the very first high rise buildings. Although these metal framed buildings were an entirely new system of construction, many architects had trouble abandoning older methods of facade design and ornamentation—favoring classical forms over new functions (Figure 2.10). Louis Sullivan’s Carson, Pirie, Scott Building in Chicago was one of the first to formally express the true structural function of a high rise building by emphasizing the load-bearing structure on the facade (Figure 2.11). Contrastingly, the space in-between these structural elements are filled with large expanses of glass. At this point in history, building skins have gained a drastic amount of independence from the load-bearing structure, but are not yet free.

Ultimately, it was the necessity of function that completely separated the two elements. Hoping to allow as much light in as possible for workers, industrial buildings in the early 1900s were the first buildings to liberate the external skin from the structure in order to maximize the amount of glazing. For example, Walter Gropius and Adolf Meyer were able to achieve this in the Fagus Factory in Germany where they suspended a three-story curtain wall in front of a brick facade (Figure 2.12). The glass facade even turns the corner without support, resulting in a beautiful aesthetic that emphasizes the principle of the curtain wall.

Free at Last
Now that the building skin had finally been liberated from its structural responsibilities, architects began to experiment with the glass curtain wall. One of the most famous of which was Ludwig Mies van der Rohe, who designed the pinnacle of modernist high-rise architecture, the Seagram Building in New York City in 1958. At this point in time, high-rise buildings were all being built out of a steel frame from which glass curtain
walls were hung to fully enclose the building. Being a proponent of the International Style, Mies argued that the making the building’s structural elements visible would be the highest form of ornamentation. Unfortunately, American building codes prevented Mies’ structural steel from being exposed for fire-proofing reasons, requiring it to be covered in a fire-resistant material like concrete. Despite all the progress that architecture made throughout history in separating the skin from the structure, Mies wanted to make sure the structure of the building was still articulated externally. Therefore, he attached non-structural, bronze-toned I-beams onto the building, running them vertically across the glass curtain wall (Figure 2.13). Although these members may suggest structural elements, they are in fact doing the opposite—hanging 1,500 tons of extra weight on the building’s steel frame. Nonetheless, this method of construction proved that external building skins are no longer restricted to just lightweight curtain walls and they are capable of being incredibly expressive, ornately decorated, and conceptually rich.

Figure 2.13: The Seagram Building in New York City hangs a building skin composed of a glass curtain wall and 1,500 tons of bronze I-beams.
Figure 2.14: Structural sealant glazing began to enclose all types of geometric forms with unswerving regularity

Figure 2.15: G.I. School by Sanjay Puri Architects in Mumbai is cloaked in a morphing, porous membrane
Following the principles of the International Style—which responds to neither context nor internal function—Mies and his followers constructed identical glass curtain wall buildings such as these all around the world until the 1970s. At this point the invention of load-bearing silicon (structural sealant glazing) allowed for architects to clad an entire building in one continuous skin. No longer were horizontal divisions necessary to attach glass panels. This led to incredibly sterile and monotonous building surfaces of constant regularity (Figure 2.14). The validity of this new style was eventually questioned by clients and investors wanting their buildings to be more distinctive and iconic. Furthermore, these sealed glass containers, relying solely on artificial air-conditioning, were an energy nightmare and brought into question after the oil crisis of the 1970s. According to Schittich, this marked the end of the curtain walls development as a building skin:

“The curtain wall in its original sense had inevitably reached its limitations. A variety of architectural styles followed the International Style. Each reacted in a different way: Post-Modernism looked back to historic examples; Constructivism questioned traditional orders; and the proponents of High-Tech Design responded with structural components. But all share one common goal; to once again give the building skin a face.”

The Digital Age

Today, in the digital age of architecture, we are still trying to give the building a face. Often the results is cloaking it in nothing more than a fashionable packaging, or decoration (Figure 2.15). We have a desire for ornamentation and a surplus of parametric tools at our disposal, but we do not fully understand how to use or apply them in meaningful ways beyond the mere image of the final product. If we as architects are to maintain our identity as makers of space and avoid becoming mere “packaging artists,” we must examine how these tools can be used to address real architectural issues. That is the challenge of today.

23 Ibid.
24 Ibid.
THE DIGITAL AGE

LOOKING FOR A MEANINGFUL METHODOLOGY

Figure 3.1: Parametric Figuration Project
The Digital Age

As the digital age of architecture continues to grow and evolve, the proper role and implementation of computational design becomes increasingly unclear. Recent advancements in the availability of and resources for design computing have brought about an abundance of new shapes and styles evident in both the academic and professional spheres. Yet to the current extent with which the profession has utilized parametrics and algorithmic thought, there is often a lack of depth or significance behind the flashy images of assumed intricacy it produces—falling far short of its rich potential to engage with the real problems, processes, and functions of today.\(^1\) Giving designers an aid to help them move out of this realm of “superficial complexity” is exactly what this next portion of my thesis seeks to achieve.

Definitions

Before I continue, I think it is important to take a step back and familiarize readers with the concepts and vocabulary pertaining to this area of design. Starting with the broadest overarching designation, *computational design* refers to the wide-ranging discipline of creating and applying computational approaches to solve design related problems. More specifically, this type of design is often executed through the customization and reconfiguration of existing design software packages to solve individual users’ specific problems—a capability called *scripting*.\(^2\) Scripting is ultimately carried out through the creation and manipulation of individual *algorithms*.

---


\(^{2}\) Ibid, 8.
An algorithm is merely a procedure for addressing a problem in a finite number of
simpler steps, each of which can be independently understood and modified. In
algorithmic design, the step-by-step thought process used to solve a particular class
of problem is methodically dictated by a human through a series of algorithms which
are then carried out by a computer.\(^3\) By doing so, users have solved not just one
specific problem, but created a system by which to solve all problems of that type.
As the parameters (input data) of the algorithm change, so does the solution. The
creation of three-dimensional shapes from this system is called \textit{algorithmic modeling}.
Similarly, \textit{parametric modeling} refers to the creation of three-dimensional forms
conceived through algorithms that specifically depend on changing parameters,
which affect the outcome of the three-dimensional form.\(^4\)

\textbf{Lack of a Guiding Framework}

This process of \textit{algorithmic architecture}, shifts the role of the architect from being
merely a "user" to being a "toolmaker." The result is a more intimate engagement
between the computer and the user that yields new ideas for design speculation as
well as time-saving production methods through the automation of repetitive actions.\(^5\)
Yet despite the advantages of scripting, there are still many questions concerning
the method by which algorithmically generated forms are designed and the meaning
or purpose behind their shapes. As suggested at the beginning of this chapter, the
majority of algorithmic methodology within the field of architecture seems to lack any
larger framework of guiding forces—resulting in those intangible products of complex
superficiality.\(^6\) Despite beliefs that increases in the affordability and availability of
computing power and software would result in the utilization of the computer to help
solve real social, environmental and technical problems, we have instead witnessed a
continual trend of designers escaping the challenges of signification in favor of fancy
forms of endless repetition and variation.\(^7\)

This thesis seeks to reverse that development. We need to shift the complexity from
the \textit{product} to the \textit{process}—only then can we see the discipline of computational
design root itself in purpose and meaning. If it seeks to maintain relevancy, the
intricate process of algorithmic thought must stand for something beyond the
mere image of the final product, and that is why this document was conceived.
The remaining chapters of this thesis will break away from the traditional format of
a research document in order to establish a meaningful methodology—guided by
larger frameworks of design—that can be referenced by designers looking for help

\(^4\) Ibid, 16.
generating ideas for building envelopes through the use of digital tools. Frameworks such as performance, program, context, structure, economy, efficiency, and fabrication will begin to govern the guiding principles of algorithmic thought in an attempt to add meaning to both the product and the process.

**Understanding the Process**

I think it is important to note that the remaining chapters will not simply be a programming or scripting “tutorial.” Although they will reference several illustrated images documenting the scripting process, its purpose is not to provide technical training. Rather, it seeks to provide a meaningful framework for the potential capabilities and applications of scripting—focusing on the logic behind digital design while avoiding the onerous task of simultaneously teaching entire software platforms.

---

**Figure 3.3: The computational design process, as originally diagramed by the Yazdani Studio of Cannon Design**

1. Series of concepts are developed based on sources of inspiration typical to the design process i.e. function, environment, art, nature, culture, science.

2. Concepts are modeled using logic based techniques. Parameters are strategically implemented to allow the systems to respond to architectural drivers.

3. Logic based geometry is “flexed” in response to simulation and data analysis.

4. Where possible, systems are rationalized and output for manufacturing.
(all of which have these resources readily available online for free). By doing so, I am allowing the relevancy of this document to survive beyond the relevancy of the individual programs or plug-ins I may reference, while also encouraging readers to avoid becoming mere clones of “ubiquitous techniques” provided by step-by-step instructional manuals.⁸ This promotes a deeper understanding of and engagement with the design process—a necessary aim if we are to maintain control of the result and contribute to the field’s continual pursuit for deeper inquiry and discovery.⁹

The “Toolset”

That being said, I will still reference several of the programs and plug-ins I employ for the purpose of guiding users towards the incredible amount of software at their disposal. The primary modeling platform I will use is Rhino—a 3D modeling tool becoming increasingly preferred among designers, architects, and students. The process of scripting will occur via the free graphical algorithm editor closely integrated with Rhino called Grasshopper. Because it utilizes a visual programming language, Grasshopper requires no knowledge of even basic programming or coding—making it very intuitive to use. Grasshopper also offers a variety of additional open source plug-ins and resources to assist users in accomplishing specific tasks or simulations. These will be referenced in the following chapters. If you would like to follow along with this document, visit the following web page to download Grasshopper for free: http://www.grasshopper3d.com/page/download-1. Although a variety of other alternative software platforms could be employed to achieve similar results, these specific tools were selected based on the their relevancy, ease of use, availability, and my personal proficiency with them.

After installing both Rhino and Grasshopper, visit this web page to find the example files covered in this thesis: https://drive.google.com/folderview?id=0BxQ2YLiC1I4hUFZZVHgxUDBmcmM&usp=sharing. Each grasshopper definition is named after the corresponding chapter in which it is covered. One of the first things we will look at is using algorithms to generate form. These form-finding techniques are setup to be a series of fast and easy analytical tools that can inform your work from the earliest stages of design. The minute you find out you have a project at a given location, within a determined site, of a specific square footage, you can begin this process. It is simply a matter of plugging in fixed, concrete information to run analyses that help you make informed decisions. After all, you are first and foremost the designer—not a script or Grasshopper definition. These are merely tools meant to empower and inform you, the designer.

---

⁹ Ibid.
Controlling the Tools

And remember, as algorithmic designers we are not solving one specific problem, but a class of problem.\textsuperscript{10} The result is an extremely iterative design process that continually adapts and adjusts along with the input data it is comprised of. For instance, let’s say you have established a script containing a series of algorithms that together create a specific pattern across the surface of a building. If suddenly the shape of the building’s surface must change, you are not at a loss. One of the primary advantages of the parametric process is that your existing script will simply adapt to the changing input values of the surface it is patterning. Therefore, if anything changes, the results are immediately accounted for—creating a dynamic workflow that supersedes the linear restrictions of previous design methods by allowing designers to switch back and forth between multiple scales and functions throughout the entire design process.

In conclusion, I will end with a quote from scripter Robert Aish:

“The designer who wants to be completely in control of the results must be in control of the process. To be in control of the process, the designer must be in control of the tools. The tools are computation, therefore a designer who wants to be in control must also be a scripter (or suffer the consequence of the unseen influence of using other people’s tools).”\textsuperscript{11}"

Let this be a challenge to you, the designer, as you venture into the realm of algorithmic thought.

Figure 3.4: Generative design
COMPUTATIONAL DESIGN METHOD #1

SOLAR RADIATION

Figure 4.1: Al-Hamra-Tower
What You Will Need
DIVA: http://www.solemma.net/DIVA-for-Rhino/DIVA-for-Rhino.html
Snagit: https://www.techsmith.com/download/snagit

One of the easiest formal generators to explore at the beginning of a design project is solar radiation. Depending on where the project is located, solar radiation can have a huge impact on the environmental performance of a building. For instance, the cooling loads of a building in the Middle East could be greatly reduced if the building was shaped or oriented in a way that minimized solar radiation on its exterior surfaces. In this chapter, I will explain how we can utilize the capabilities of algorithmic tools to establish a generative process for analyzing solar radiation at a given location in order to algorithmically optimize a building form to reduce solar radiation on the surfaces of the output geometry.

Galapagos: An Evolutionary Solver
The process we are about to undergo requires the use of the Galapagos component within Grasshopper. Galapagos is essentially an algorithm that applies evolutionary logic to solve a specific problem. We will be using it to optimize a flexible shape to best achieve a user-defined goal. This goal is called the fitness value and in this scenario it is the lowest possible value of total solar radiation measured across the shape’s surfaces. In order to generate different shapes and test their effectiveness at lowering this fitness value, Galapagos needs a series of options, called genomes, to experiment with. These genomes are the numbers that define the shape of the geometry being tested (i.e. building rotation, height, length, width, etc). Depending on the quantity of possible combinations, this calculation could take hours or even days to compute. This is where the “evolutionary” side of Galapagos is very helpful. Fortunately Galapagos does not need to churn through every possible combination to determine an optimal solution because its solutions are based on identifying and matching positive characteristics within the genomes. After each combination, Galapagos learns from the results and takes the best characteristics of each genome to move progressively closer to an optimal answer.¹

Need for Customization
The grasshopper definition covered in this example—originally created by Yazdani Studio—is as close as I came to a “one-size-fits-all” generative model. Although computational design allows for endless generation and variation, creating a script that offers parametric control over every possible feature of every potentially generated geometry is seemingly impossible. And if not, the complexity of creating a script like this would not be worth my time to undertake, nor yours to decipher.

Rather, as I stated earlier, this document encourages readers to avoid becoming mere followers of the techniques I am demonstrating in order to better understand and engage with them—creating easy opportunities for customization.

As you customize and define the limitations of your output, always be knowledgeable about the outcome you are allowing the definition to create. Every definition has limitations on what potential geometries it can produce. For instance, the definition shown in this example consists of a loft between two planar curves representing the ground floor plate and the roof floor plate. As a result, the top of my resulting geometry will always be planar—no matter what. Therefore, when I examine and review the generated outcome, I must keep in mind that the most optimal roof shape is not necessarily flat just because the resultant geometry is. After all, the definition I setup never allowed for a different result. But once I generate my first study, it is very easy to setup a second that looks deeper into the optimization of the roof. However, if I was unaware of this limitation from the beginning, I would not know to do this.

![Figure 4.2: Fixed input data](image)

![Figure 4.3: Variable input data](image)

**Plugging in the Data**

The first step of the process is plugging in the *fixed input data* that will drive the script and always remain the same. Although these may be viewed as the restrictions or limitations of a project, the more of these you have the more refined your outcome will be. In this specific example, the fixed input data is the gross square footage of the bottom floor plate and the top floor plate. The next step is establishing the potential minimum and maximum values of the *variable input data*. The variable input data is essentially a range of numbers that control specific elements of the building’s potential configuration (i.e., allowable degree of rotation) and are controlled by *number sliders* within Grasshopper. These sliders are the genomes plugged into Galapagos, which will use these numbers to analyze thousands of different combinations to generate an optimal solution.

Always be looking for ways to customize the script to fit your needs. If your project is flexible with gross square footage, consider creating a range of potential value and moving it into the “variable input data” category.
In this example we are starting with a rectangular extrusion that is 50’x100’x100’. I have adjusted all of the genomes that comprise the variable input data to accommodate a desired range of solutions. As mentioned earlier, the building geometry is created by lofting two separate curves. Each curve is controlled by four points that can each move along both the x and y axis. As a result, this example only allows for a handful of geometric outcomes. If I were interested in a more complex geometry, I could easily duplicate one of the existing floor plate setups and add it to the loft for more control. However, because this slows down the time required for the solver to run, I will only be optimizing a two curve loft.

**Division of Massing**

The resulting shape is then divided into separate glazing and spandrel surfaces based on the user defined floor to floor and ceiling heights. Because spandrel panels are typically better insulated than glazing, only the surfaces with glazing will be evaluated. Now it is time to start using the plug-in **DIVA-for-Rhino**. DIVA is a day lighting and energy modeling plug-in for Grasshopper that allows users to carry out environmental performance evaluations.
There are other plug-ins that can do the same function (i.e. Geco/Ecotect, Ladybug, Heliotrope), but as far as I know DIVA is the only one that has been validated in the field for producing accurate results.² DIVA also allows for the classification of different materials to better represent their physical properties. Because surrounding buildings have a huge impact on shadows as well as reflections, make sure you input this information into the definition as well. At the very least you will need a ground plane, because a lot of incidental solar radiation is actually reflected off the ground. Next, it is very important that you set the simulation to the appropriate weather location. After this, you must specify the run period of the simulation. This is also important because in most locations, you do not actually want to minimize solar radiation year-round. Therefore, find out the cooling days for your given site and input them into DIVA.

Now that you have all of this information setup, it is time to let Galapagos do the rest. Connect all of your variable input data into the Genome input of Galapagos and connect the total solar irradiation value into the Fitness input. Make sure that the fitness is set to “minimize” and then start the solver. Although it will slow down the computation, I would suggest you click “Display all genomes in the Rhino viewport.” This will allow you to see every single genetic combination that Galapagos will analyze. This is where Snagit comes in handy. I recommend using a screen capturing program like Snagit to record the evolutionary process that Galapagos undergoes when running the solver. By doing so, you can see trends in the results and let these trends inform your future design decisions. In fact, this information is probably more valuable than the actual final result.

The Value of Metrics
You may be wondering if this process actually accomplishes anything more than a simple common sense solution based on an informed designer’s general knowledge of solar radiation. This is a fair question to raise. But one of the most important things to note when using evolutionary solvers is that *dumb questions get dumb answers*. The value of this definition lies entirely on the data you input and the limitations and restrictions you set. The more variables you make Galapagos account for, the more interesting and valuable your answer will be—especially if these variables are not easy to visualize on your own. For example, my results could be improved and made more useful if I would have included some nearby buildings in the simulation. One final note, this generative method will give you validated information for each configuration, allowing you to numerically compare the pros and cons of different forms. This is a level of sophistication that general rules of thumb cannot produce.
The Result

To better see and understand why the generated forms are shaped the way they are, I used the Gradient component in Grasshopper to depict which areas were experiencing the highest (red) and lowest (blue) levels of solar radiation. For the location of my simulation I used Cincinnati, Ohio and then limited the duration period to only the summer months—as would be expected in this climate.

I was not expecting much of a difference between my initial form and the generated “optimized” form as I thought I already had a pretty good setup to reduce solar radiation by making the building as narrow as possible within the constraints to minimize its southern face. Nonetheless, the results surprised me and I noticed a trend which lengthened and rotated the uppermost portions of the south facade in order to better shade that of the west. The value and effectiveness of these moves can then by analyzed and evaluated with the numerical output that accompanies them. All the solutions shown reduced solar radiation by approximately 8%-10%. This tells the designer that the exact configuration of the final solution isn’t nearly as important as the basic form these potential solutions share. And although 10% might not seem like a significant reduction, keep in mind that this was an additional 10% reduced from the common sense solution I started with.

Figure 4.13: Resulting formal generation

Figure 4.14: Galapagos Solver: graph at the top shows the optimization of the fitness value as it progresses through multiple generations of configurations. Charts below show the convergence of all of the tested genome configurations.
COMPUTATIONAL DESIGN METHOD #2

VIEWS

Figure 5.1: Detox Towers
What You Will Need
Snagit: https://www.techsmith.com/download/snagit

Depending on your site and its surrounding context, views may be something you will want to emphasize early on in the design process. In fact, unobstructed views to something such as a waterfront can be a valuable asset for many building owners as it can raise the value of their property and the asking price of their rentable space. Furthermore, there is a growing body of research in workplace design that suggests offering workers external views is extremely beneficial. Studies show that access to views has been shown to reduce boredom, anxiety, and stress while simultaneously improving productivity, health, and job satisfaction. In this chapter, I will explain how we can utilize many of the same algorithmic tools from the last chapter to create a generative process that optimizes a building form in order to maximize the number of unobstructed sightlines from a building to specific points of interest.

Plugging in the Data
Just like we did in the last chapter, this process begins by plugging in all the fixed input data such as the gross square footage of the top and bottom floors and the floor to floor heights. Because the Grasshopper definition used in this example is

---


---

Figure 5.2: Site plan of urban context
almost identical to the one we used for solar radiation, I have kept all of my fixed and variable data the same and will not go into detail again on this portion of the definition. However, everything pertaining to calculating solar radiation has been replaced by a series of algorithms that instead calculate sightlines. This portion of the definition is where we will begin.

The main difference you will find in this example is the addition of a context model. After all, you cannot have unobstructed sightlines without something to view and something to obstruct. Once you create a context model, the next step is establishing user-defined points of interest. Using your knowledge of the site, determine the location of specific points you may want to maximize views towards. In my example I have created a hypothetical urban environment and established three main points of interest that represent the following: a large waterfront, a nearby park, and a neighboring historic building. To create a point of interest, simply draw a curve across the length of the object or area. Once you load this curve into the definition, Grasshopper will break it down into several points (the density of which you can control). If you prefer just to make one individual point, that will also work. Remember, that these curves (or points) can be anywhere and do not need to be planar or at a specific height. That is why you see a building facade populated with points of interest in this example.

Now that we have desired points to see, we must set up vantage points to see them.
from. These are the points on the building facade that will connect to the points of interest to make a sightline. Much like the location of the glazing in the solar radiation tests, these points do not need to occupy the entire building facade as there is realistically only a small portion from which most occupants will be looking out from. Because of this, I placed the vantage points along the facade at the location of each pre-defined floor plate and then moved them up five and a half feet—the approximate height of an average occupant’s eye level. Depending on the conditions of your project, this may need to be adjusted. For example, you may have a mechanical room, stairwell, storage space, or restroom along the facade that you don’t care about generating views from. The results of your Galapagos solution will be more helpful if you address these variables early on in the process. As I mentioned before, the more variables you make Galapagos account for, the more interesting and valuable your answer will be.

**How it Works**

The way this script works is that it tries to connect every vantage point on the building to every surrounding point of interest with a straight line. This line represents an *unobstructed sightline*. If anything gets in the way or intersects with one of the sightlines connecting two points (such as the building itself or one nearby), that line is erased. Thus the total number of sightlines created becomes the new method for evaluating the effectiveness of a given form. This number becomes the fitness value that plugs into Galapagos. However, unlike solar radiation, we will trying to maximize this value instead of minimizing it. Therefore, Galapagos will now be trying to generate an optimal building geometry that yields the highest possible amount of unobstructed sightlines.

**Visualizing the Data**

Once again, make sure you select, “Display all genomes in the Rhino viewport” before running the simulation so that you can use Snagit to record the results. This allows you to see trends beyond just the final iteration that can help inform your future design decisions. It is also important because once you close out of Galapagos you cannot revisit this information again. That is why this example utilizes *Data Recorders* to save and store the results of each configuration for later review. Unfortunately, you still won’t be able to see the corresponding fitness graph or genome chart of the configuration, but it will of course still allow you to extract the numerical data of each generated geometry. Make sure to clear any previously recorded information by pressing the “X” on the component before running Galapagos. The definition is
Figure 5.8: Gradient component

currently setup to record the fitness value of each iteration, as well as the two curves that were lofted to produce that number. Therefore, once you re-loft these curves, you can see and review all of the forms and their resultant values in tandem. It may be easier and simpler just to record the form produced and not the two curves. However, the curves were recorded instead of the actual form they comprise because three dimensional surfaces take up considerably more space than do curves and can greatly slow down your computer if it is running a long calculation.

To better see and understand why the generated forms are shaped the way they are, it is important to setup a Vector Display component to show all of the unobstructed sightlines. Similarly, it is also useful to use the Gradient component to visualize densities of vantage points along the building facade (much like we did with solar radiation). In this example, the Gradient component is creating a black and white pattern on the facade that depicts which vantage points offer the most sightlines (black) and which offer the least (white). Besides helping us understand the generated form, this pattern can also be very useful in coming up with further ideas for articulating the building envelope. For example, the pattern created by these sightlines could be developed into a system for placing window openings and determining their respective sizes. This topic of patterning and skin articulation will be covered in more detail in a later chapter.

The Results
The generated building forms from Galapagos can be found on the following pages. It is evident that the optimized shape was much different than the one we started with. Unsurprisingly, the first noticeable trend was an increase in the building height, which created more vantage points and provided sightlines above nearby buildings. As a
result, I made a common sense decision to maximize the height of my original building shape and re-run Galapagos so that I could more accurately compare it to the optimized results. After running the solver through a few generations, the building footprints became skewed at an angle that responded to the curve of the waterfront, where most of the points of interest were located. This increased the surface area on the north side of the building and made the building longer in the east-west direction instead of the north-south as it was originally shaped. This seemed to be the most effective formal trend.

Just like with solar radiation, we can measure the true effectiveness of these moves with quantifiable data. All of the solutions shown here provided a 24-39% increase in the number of sightlines as compared to the initial form. Now it is up to the designer to choose how these results influence his or her work. Although the final building design may never adopt any of the exact forms generated by this study, it could still be informed by and respond to the information they convey.
Figure 5.10: Aerial perspective of unobstructed sightlines from optimized form

Figure 5.11: Site plan with unobstructed sightlines from optimized form
COMPUTATIONAL DESIGN METHOD #3

DAYLIGHT
Although often under-evaluated during the design process, daylight is actually one of the most prominent factors in reducing the energy costs of buildings. In fact, most building owners spend more on lighting spaces than on cooling them. Simply by allowing more daylight into their buildings, owners could vastly reduce their need for electric lighting—resulting in more energy savings than most external shading devices can provide in reducing solar radiation.\(^1\) Furthermore, daylight has huge psychological benefits, especially in workplaces. Research suggests that greater doses of daylight in the workplace can reduce stress and improve productivity, physical health, and mood.\(^2\) However, there are dozens of methods for measuring daylight performance in buildings and it is crucial that designers know and understand the best method for their desired area of exploration.

### Useful Daylight Illuminance

What we will be examining in this chapter is how daylight can reduce the need for electric lighting and help cut down on energy costs. The best metric for measuring the effectiveness of daylight to achieve this goal is Useful Daylight Illuminance (UDI).\(^3\) UDI gives the percentage of time a series of test points are receiving useful levels of daylight illumination. These test points are typically located at the height of the horizontal work surface (two and half feet off the floor) where most of the tasks that require adequate lighting take place. Although there is much debate on what range of values are considered “useful,” the standard is anything between 100-2000 lux. When daylight illumination levels reach 100 lux, electric lights can be dimmed within a workspace. At 300 lux, the lights can be turned off all together. Anything over 2000 lux is not considered useful due to glare and overheating.\(^4\) This is a common mistake designers make as they often assume all daylight is useful and desirable. However, if a work surfaces is being overwhelmed with daylight, it can be very difficult to see a computer screen or even read a document against a white page. Although rules of thumb can help with illumination calculations, it is hard for designers to intrinsically know or visualize these levels—let alone have them inform their work. In this chapter, I will be walking you through a script that generates a floor plate shape with the highest possible percentage of UDI across its respective work surface test points.

---

The Tools
The first thing you need to do is download Ladybug and Honeybee by following the link at the beginning of the chapter. Ladybug and Honeybee are two of the most advanced and comprehensive environmental plugins available for Grasshopper and work in tandem with one another. Ladybug focuses on importing and analyzing standard weather data to produce customized analytical diagrams—such as a sun-path, wind-rose, radiation-rose, shadow studies, etc. Our previous exploration of solar radiation using DIVA could also have been achieved using Ladybug radiation analysis (DIVA was chosen instead because of its simplicity and it gave me the opportunity to introduce you to multiple plugins). Honeybee is the more advanced plugin and allows users to connect Grasshopper to several simulation engines such as EnergyPlus, Radiance, Dayism, and OpenStudio. As a result, you should also install Radiance, Dayism, and EnergyPlus. Honeybee and Radiance will be your primary tools for calculating UDI, but you might as well install everything else in the package at the same time for potential future use.

Plugging in the Data
The next thing you do is let the Honeybee component “fly.” To do this you just need to drag it onto the grasshopper canvas. I have already done this for you so it should work just fine. If not, try dragging it into a new definition and then reload the example file. Because daylight analysis requires so much more computing power than the previous examples did, this file is setup a little differently. For the most part we will be using all of the same basic parameters and variables. However, this time we will only be looking at an individual floor. Plug all of your fixed input data into the definition (i.e. gross square footage, floor to floor height, ceiling height, etc) and customize as necessary. This also includes downloading an EnergyPlus Weather file (.epw) from your site’s location (or at least one nearby) and loading it into the definition. This weather data will insures that you will be running the simulations under conditions representative of your environment. In this example, I chose to make the ceiling height “fixed” in order to reduce computing time. However, this alone could be the sole variable in its own separate study. I also set the gross square footage to 10,000 to make it more challenging to get adequate light into the center of the floor plate. Next, I setup the appropriate ranges for the variable input data. This simulation takes a long...
time so I suggest limiting the ranges even further than before. Unlike the past two examples (which featured two lofted curves), this definition consists of a single extruded triangle or parallelogram with floor-to-ceiling glazing. I realize there may be many design situations in which you would not want this much glazing; therefore, I created another Grasshopper definition that informs the placement and sizing of windows that you can use if interested. But for now, we will continue as if we were studying a single floor in a glass curtain wall office building.

Setting Up the System
As you remember from the last two examples, we used a custom component to break up our lofted form into several different material components (floor, ceiling, spandrel, glazing, roof) to be analyzed independently by DIVA. We will do the same thing with Honeybee, but this separation is even more critical now as it determines the most important factors in a daylight study—which objects are letting light in, which objects are reflecting light, and how much of both. While it is possible to further customize each of these components to match specific material properties, I am assuming you don’t know this information yet and therefore have left all of them in their default states. Also, if you have a site or context model, you would input that information here. Since this is merely a proof of concept study, I have not input any contextual information.

Next we must setup the test points for our analysis. As mentioned earlier, the most critical place to measure illuminance levels is at the horizontal work surface where most tasks occur. This is typically two-and-a-half feet above the floor at desk height. The density of the test point grid is also important. The higher the density, the more accurate the results will be but the slower the computation.
will run. Similarly, fewer points result in faster computations but less accurate and informative results (sometimes even worthless). Try to find a balance where you are getting values for several different conditions but the computational time required is realistic.

Running the Simulation
It is now time to calculate the UDI percentages. A Boolean Toggle Switch (located at the beginning of this definition) is plugged into the Honeybee_Run Daylight Simulation component and controls the entire simulation. As soon as this turns on, the simulation will start running. You will also find a separate slider plugged into the Honeybee_Run Daylight Simulation component. This slider controls how many Central Processing Units (CPU’s) within your computer will be dedicated to running the simulation. It is a great feature for speeding up a calculation and one that Honeybee offers and DIVA does not. The higher the number, the more of your computing power will be dedicated to running the calculations and the faster the simulation will go. Unfortunately there are only so many CPU’s in your computer so the number you should enter depends solely on what you have available in your machine (i.e. I had 16, so that is what I used). Keep in mind that the number you enter also corresponds to how many command prompt windows open on your computer in order to run the calculation. Once the calculation is over, Honeybee offers a variety of daylight analyses tools to review the results. In fact, if you prefer to study a different daylight metric in place of UDI, it is very easy to change and customize your output. Even within UDI you can change the analysis days, the analysis hours, the range of useful illuminance exported, and even add shading sensors and electric lighting within the building.

Galapagos
Once again we’re going back to Galapagos to generate an optimal solution. Plug all of the variable input data you established at the beginning of the process into the Genome input of Galapagos. Next, connect the total sum of all the test points’ UDI percentages into the Fitness input. As mentioned in the previous chapter, we will be using Data Recorders to keep track of all of the results. Because this is just looking for an optimal shape for an extruded floor plate, we only need to record the boundary curve of the surface being tested and the corresponding sum of the percentage values. Before running Galapagos, make sure any previously recorded information is erased from the data recorders and that the fitness value is set to maximum. Then make sure to select “Display all genomes in the rhino viewport” and capture the entire evolutionary process with Snagit. This setup allows us to solve for a shape that will generate the highest possible percentage of Useful Daylight Illumination within a given square footage.
The Results

On the following pages you will find the results of my experimentation with this process. Once again I used Cincinnati, Ohio as the location and limited the study to only calculate the annual UDI percentages of the test points during normal business hours—Monday through Friday, 8am to 5pm. The results were interesting and I definitely noticed a trend in the optimized solutions.

Starting with a 100’x100’ square floor plate and a 10’ ceiling, the results were pretty much as expected. The worst UDI percentages were recorded along the perimeter adjacent to the glazing as this would obviously be an area with too much light—resulting in glare or discomfort. As you move inward the percentages go up until you get to the middle, where it becomes evident that the light is having difficulty penetrating that far into the building.

The best results came from rotating the square about 45 degrees and then skewing it to make the longer in the east-west direction, which increased its exposure to the sun. A long, singular point protrudes southward, preventing the creation of a true south facade and helps keep some of the direct sunlight from penetrating the floor plate. Meanwhile, the point on the northern end was considerably shorter in almost all the solutions, making for longer north facades. Ultimately, it is the north end of the floor plate that sees the best result, which makes sense given that it does not have to deal with direct sun exposure against the glazing like the south does. Starting with a simple square, this simulation ended up improving the total UDI percentage of the floor plate by almost 20%! Like the testing with solar radiation, once Galapagos got to a certain point, it was easy to see trends in the data, and the best solutions were all within 2% of each other.

Figure 6.10: 8-story extrusion of optimized footprint for improving UDI
Figure 6.11: Resulting formal generation of UDI optimization

Figure 6.12: Perspective of optimized floor and surrounding walls

Figure 6.13: Galapagos Solver
COMPUTATIONAL DESIGN METHOD #4

SKIN PATTERNING

Figure 7.1: Sinosteel Tower
What You Will Need
LunchBox: http://www.food4rhino.com/project/lunchbox
LadyBug: http://www.food4rhino.com/project/ladybug-honeybee

This portion of the document begins to leave the more generative, “plug and chug” methods of the last few chapters. No longer will we be covering a “one-size-fits-all” grasshopper definition that can easily be applied to any project. This is mostly because the preceding portion of this document dealt with generating a building mass and now we have reached a point where we must begin to articulate and define its composition and aesthetics. Unlike developing an abstract massing or form, this aspect of the design process is a lot more specific and involves far too many variables to reasonably include in a single script. Nonetheless, there is plenty of opportunity for algorithmic thought in building skin design. In fact, building skins might have the single greatest potential for algorithmic application.

Data Driven Design
In this chapter, I will be guiding you through a methodology that I believe could be applied to any project. Despite the fact that every facade on a building usually has different environmental, contextual, and programmatic functions, they very often share the same exact design articulation. This chapter looks at fine-tuning a panelized building so that each panel can responding to a specific condition experienced at its exact location. If done well, the result is a beautifully patterned facade with a rich aesthetic, hinting at the data from which it was derived.

The building mass we will be analyzing in the following examples is a simple square extrusion. The footprint is 200’ x 200’ and it is 300’ tall. We will be looking at 4 different conditions a designer could respond to when designing a building. Each elevation will not only exhibit a response to a unique condition, but a different system of paneling as well. Ultimately, the condition you choose to address and the system within which it is realized is up to you as the designer. Therefore, try to look at this chapter as an over-arching framework from which you could begin to manipulate and customize certain aspects to assist you in the needs of your specific project.
Image Patterning

Let’s start with the first Grasshopper definition for this chapter, labeled *Part 1*. Essentially, this definition is dividing an entire surface representing the building facade into an organizational system of panels. The script then takes the center point of each panel and creates an opening (or window) of the same shape within the existing panel—like an offset. This information is then connected to an *Image Sampler* component in Grasshopper, which scales the opening of each panel according to a user selected black and white image. The size of each opening is determined by the “brightness” of its corresponding point on the image. Typically, the darker the point, the bigger the opening. The lighter the point, the smaller the opening.

Creating the Image

You probably think this sounds like an interesting system for composing window openings, but lacks any substance or value. This is why the image you choose to sample is so important! For example, you could create an image showing the location of external views from the building and the size of your window openings would corresponded to these locations. Not only could this improve the function of the windows, but also produce a beautiful pattern or aesthetic on the building which reflects the data that derived it. On the following pages we will be using the second Grasshopper definition, labeled *Part 2*, to create useful images corresponding to four different potential conditions (solar radiation, wind pressure, views, and program). Once we generate these images, we will plug them back into the first definition to produce a pattern.

**Elevation 1: Solar Radiation**

Fortunately we already have the necessary tools to calculate solar radiation. Because Ladybug has a few advantages in conducting studies like this one, I have chosen to use it instead of DIVA. Unlike the very first example, we want to find the solar radiation values across...
an entire elevation—not just at predefined areas where the glazing is located.

The first thing you would normally need to do is model your proposed building and plug it into the _geometry_ input of the _Ladybug_Radiation Analysis_ component. I have already done this for you. Before I run the analysis, I must first load my EnergyPlus Weather file for Cincinnati and set my analysis period to take place only during the summer (June 21—September 22), when solar radiation is an issue. To get a more interesting pattern, I also created a few masses to represent context buildings that cast shadows on my elevation and plugged them into the _context_ input of the _Ladybug_Radiation Analysis_ component. Unlike the previous studies, I made sure to override the colors to be a black and white gradient so Grasshopper can easily compare brightness values on a single scale. Black corresponds to low solar radiation, and white represents high solar radiation.

After I have this information, I use a Rhino Viewport that corresponds to the face of my elevation and export or screen capture the results. I then open this up in a photo editing software such as Photoshop and crop the image to just the elevation. It is important to note that the Image Sampler component in Grasshopper will automatically distort your image to make it match the dimensions of the surface it is being applied to. Since a distorted image would completely undermine the purpose of this data driven process, it is imperative that the dimensions of the final image being used are proportionate to the facade it is being
applied to. For example, the building facade covered in this example is 200' wide by 300' tall. The image doesn’t need to be this exact size, but it does needs to have the same 2:3 ratio in order to keep from being altered.

Once this is done, go back to the first Grasshopper definition and load the image by double-clicking the Image Sampler component and resetting the file path accordingly. Next, change the channel to brightness. A plug-in called LunchBox will be dividing the entire surface of our facade into a panelized system. In this first example, I will be using the Diamond Panels for my system of patterning. I then chose to match the number of horizontal and vertical panels to the overall proportions of the facade so that the diamonds would be symmetrical—but you can set them however you see fit. After this, it is time to adjust your results. This is done by manipulating the Domain Start and Domain End sliders—each of which can have a value between 0 and 1. All of the panels will be resized according to this new domain range, so it is important to set it appropriately. For instance, if the start is set lower than the end, then all the results will be inversed—meaning a darker color in the image sampler will yield smaller window openings. In this study, it makes most sense to increase the window size where there are lower solar radiation values and decrease the window sizes where there are higher solar radiation values—creating an environmental response that lowers heat gain.
Elevation 2: Wind Pressure

Wind pressure can be a huge factor in building design, especially at great heights. In fact, for skyscraper design, the wind loading on the exterior is often the principle structural consideration as its lateral force is often larger than the dead or live loads. Therefore, this next example will look at using wind pressure values on a facade as a basis for sizing a structural frame and the windows it holds. To get these values, I will be using Autodesk Flow Design.

The first step is to select the model of your building mass in Rhino, and export it as an SAT file. Make sure to use “Inventor” in the export settings. Next open up Autodesk Flow Design and import your file. Makes sure the light bulb icon at the top of the screen is turned off so that the colors are not being distorted. Under Simulation, make sure to set the Model Dimensions to match your Rhino units. Next, click on Wind Tunnel and change the wind speed to match the conditions of your site. Because you cannot alter the direction of the wind, rotate the orientation of your building instead by changing the Z Angle in the Orientation tab. Next, make sure you are running a 3D analysis of Wind Velocity and that Surface Pressures and IsoSurface are turned on. Under the IsoSurface Settings, make sure the opacity is set to 1. Lastly, set your legend colors to Grey and then find your elevation view. Wait for the status to say Stabilized, as this means the solution is done calculating, and then save an image of the results.

Open the image in a photo editing program and crop it down to just the facade elevation, making sure the proportions of the final image match the facade dimensions. Once this is done, go back to the first Grasshopper definition and load the image. In this example I will be using LunchBox’s Hexagon Cells for my panel.
system. Once again I specify the number of horizontal and vertical panels to match the proportions of my elevation. Finally, it is time to adjust the domain. Keep in mind that the goal of this study is to make the structure thicker where the wind pressure is higher, and the windows bigger where the wind pressure is lower.

Elevation 3: Views

In an earlier chapter I mentioned the potential of using view density patterns to inform a facade. Here is where we apply that information. Instead of generating a form that creates more views, we will be analyzing the conditions of an existing form and responding to those views in the sizing of window openings. Once again, go back to the second Grasshopper definition for this portion of the example. Here we have a setup for identify sightlines that is very similar to the one we had before. I set the spacing of the new mass’ floor-to-floor height at 12’ and then moved the vantage points up five and half feet above those so it could be at eye level. Hypothetical points of interest were established and the same context buildings from the solar radiation example were put to use once more. Now we must examine the pattern of view densities with a black and white Gradient component that shows which vantage points offer the most sightlines (shown in black) and which offer the least (shown in white). Setup a Rhino Viewport for your next elevation and save a copy of the new pattern.
Crop the image appropriately then go back to the first Grasshopper definition to load the image. In this example I will be using LunchBox’s *Quad Panels* for my facade’s organizational system. Once again, I specify the number of horizontal and vertical panels to match the proportions of my elevation and begin adjusting the domain. As was the goal, I make the window openings larger where there are more views, and smaller where there are few views.
Elevation 4: Program
This final example looks at using a building’s internal program to drive external facade design. First I began by creating a hypothetical program for an office building and laying out all the spaces (private offices, open offices, conference rooms, circulation space, storage rooms, etc). Next, I organized these spaces according to their respective levels of privacy. White represented the most private, and black represented the most public. I then turned this into a cropped image of the programmatic components along the facade and plugged it into the remaining Image Sampler in our Grasshopper definition. In this example we will use LunchBox’s Triangle Panels as the system for panelizing the facade. After adjusting the number of horizontal and vertical panels, it is time to adjust the domain and finalize our results. Because this triangular system is a much denser looking panel, this one takes a little bit more time to perfect. In the end, I have created a facade in which the public spaces within have bigger, more open windows while the private spaces have much smaller windows.
Conclusion

Regardless of how you realize these patterns, I think the concept of *data driven patterning* has a lot of validity and potential. It is a great method for not only improving the function of internal spaces, but it also adds an aesthetical element to the external facade that is rooted in something definable and meaningful. The goal of this chapter was to introduce you to some of these ideas and give you the necessary tools to better explore them.
Figure 7.25: Solar radiation
Figure 7.26: Wind pressure
Figure 7.27: Views
Figure 7.28: Program
FINAL REMARKS

CLOSING THOUGHTS ON DATA-DRIVEN DESIGN
Final Remarks

I stated near the beginning of this document that my primary goal was to provide you, the reader, with a basic framework for implementing algorithmic thought in architecture. I think it is fair to say that we have covered a variety of easy-to-employ methods for several different tasks, but I want to touch back on the overarching framework once more.

As you probably have discovered by now, this thesis is ultimately about acquiring data in order to inform design decisions. Data-driven design (a design process informed by quantifiable, numeric information) can be a very powerful design methodology because it grounds itself in facts. However, the validity of these “facts” is ultimately dependent on the methods you employ to acquire them. In other words, how you define your algorithmic systems ultimately determines if your data is credible. For instance, an incorrect analysis period while calculating for solar radiation would yield incorrect data. Just because the ensuing design would be based off of “data” obviously does not mean it is credible. Keep this in mind while partaking in this process and always ask yourself if the results you are seeing make sense. If your data is accurate and your process logical, the application of this information is extremely powerful.

We are the so-called “big data” generation and I believe that the representation of this data in and on our buildings is becoming architecture’s new method of ornamentation. As more and more architects begin moving towards parametric tools to decorate and package their buildings, we should be seeking to connect these patterns with quantifiable data that addresses real social, environmental and technical problems. Thus, I encourage you to keep exploring new ways to acquire useful data and let it inform your design work.
DESIGN PROPOSAL

THE “RE-SKINNING” OF THE KROGER BUILDING
The Kroger Building

In order to more fully exhibit some of my proposed thoughts and methods on algorithmic architecture and building skin design, I chose to focus my design proposal on “re-skinning” an existing building. I selected the Kroger Building in downtown Cincinnati to serve as a base for these ideas. The Kroger Building is located at the intersection of Vine and Central Parkway. It is the tallest building north of the financial district and is arguably the most noticeable structure when headed downtown from the north side. The building has 25 stories and is 320 feet tall.\(^1\) The building is the corporate headquarters of the Kroger Company, which was founded in Cincinnati in 1883. Kroger is the largest supermarket chain and the second-largest retailer in the country.\(^2\) With over 2,640 stores in 31 states, their size and growth is astonishing.\(^3\) Despite such great financial success, Kroger has chosen to stay headquartered in Cincinnati and continues to work out of the same building since it was constructed in 1959. While many other companies their size have opted for iconic, state-of-the art, corporate campuses, Kroger has stayed dedicated to maintaining their historic roots in their nostalgic location. However, despite the historical significance of this building, the company is now realizing that their dull and outdated workplace is in serious need of a facelift. Since relocating their corporate headquarters is not an option, this project proposes that Kroger should re-clad their entire building—a process they underwent once before in 1980 when GBBN Architects updated the building with a new aluminum skin.\(^4\)

This would be a great option for Kroger to pursue once more as there is strong relationship between a corporation’s physical workplace and the image it portrays. Despite the many unobservable aspects of a given company, their built environment is the one thing that can always be seen. In fact, the appearance of the workplace has a large impact on how the company is perceived by the general public, stockholders, customers, and the corporation’s own employees. These appearances influence consumers’ perceptions of price level, quality of product, decisions of employment, and even investors’ decisions of investment. Therefore, the image being portrayed by Kroger’s corporate headquarters is of upmost importance—particularly with its iconic size and location within the city.

No longer can dull and outdated be the image communicated by this office building. By enhancing the appearance and functionality of its headquarters, Kroger can not only improve its own identity, but simultaneously give back to its surrounding city. For instance, a new exterior skin could change the building’s lackluster presence in the

city's skyline, its weak relation to the street, and its poor interior work spaces. Talks with Kroger employees resulted in the establishment of 7 different project goals: to create an iconic landmark, to nurture and inspire Kroger's culture, to create multi-use breakout spaces, to provide access to fresh air, views, and daylight, to reduce energy consumption, and to incorporate passive and active systems (Figure 9.5).

Parametric Customization
Throughout the last century, mass production has been the guiding rule of modernism. The premise has been that through repetition, we can cut building costs. Unfortunately, the result is often sterile building surfaces of constant regularity across every facade—regardless of extreme differences in both internal and external conditions. These monotonous buildings are not only displeasing to the eye, they are completely unresponsive to their own needs. However, recent technological advancements are changing everything. With the aid of digital fabrication, complexity no longer equates to higher costs. Architects and designers alike are using parametric tools to organize the surplus of data available to us today and use it to efficiently manage complex forms and tasks. This allows for designers to fine-tune every portion and panel of a building in response to its exact conditions.5

We must look no further than the scales of a reptile to see this same principle in nature. Much like our skin, these rough external plates protect the body of snakes, lizards, crocodiles, turtles and more (Figure 9.6). Interestingly, the parameters (size, shape, flexibility, etc) of each scale change depending on its location on the body. For example, scales near the eyes of snakes actually protrude out to shade and protect the eye while their scales that touch the ground are flatter and smoother. I believe this system of geometric divergence to improve functionality is a perfect metaphor for how a building skins can be fine-tuned to meet the specific needs of their exact locations.6

---

Outer Skin

Using this metaphor as a guiding logic, I am developing a building skin that responds to varying internal and external conditions at the location of each individual panel. Much like a reptile, the building envelope will be systemically organized into a diagrid of diamond shaped “scales” or panels with a window offset inside of each one. The articulation of these windows and their outer frames will cater to multiple data-sets simultaneously. For example, the internal program of the building will be utilized as a data-set driving the size of each of the windows within the frames. Additionally, the varying angle, location, and intensity of the sun’s rays (quantified by solar radiation) will be used as a data-set driving the formation of the frames that surround these windows. Therefore, the parameters of each panel will vary according to its location on the facade. For instance, panel frames on the south facade will have larger extrusions at the top, while panel frames on the east and west facades will have larger extrusions on the left and right sides (Figure 9.7). The result is a coherent and unified system that can be applied across differing facades in an extremely responsive way. In this system, functional and formal variation go together. The movement of the sun translates into a gradient transformation reflecting the panels evolving formations. Thus, the performance-driven shape of the parametrically generated panels becomes heightened into an artistic concept.7

To achieve this goal, I had to find the total radiation of the windows using Ladybug and then run Galapagos to find the optimal frame extrusion to best shade the windows within. In order to reduce the number of different frame types, I divided each facade into 4 different regions and placed a panel in the middle of each one (Figure 9.8). Next, I ran Galapagos to optimally form it to equally minimize solar radiation in the summer and maximize solar radiation in the winter. As a result, each facade has four potential frame types. I then used the same objectives in Galapagos again to optimally populate the facade with these frame types. The result is essentially a gradient of the 4 different frame types across each facade, for a total of 16 different frame types (Figure 9.9). Each of these frames is then populated with one of 4 different in-fill panels. They range from completely transparent to completely opaque. The specific designation for each panel is based off of the internal program of the building. I assigned different levels of “porosity” (openness) to each programmatic space based on its use, need for daylight, and desired level of privacy. These values were then plugged into an Image Sampler in Grasshopper, which ultimately produced the resulting pattern of panels.
16 FRAME TYPES:

NORTH

SOUTH

EAST

WEST

4 PANEL TYPES:

Figure 9.9: All 16 different frames arranged by facade and the 4 different panels that fit within them
Figure 9.10: Components of outer skin

- Diagrid steel structure
- Optimized frames inset into diagrid
- Panels inset into frames
- Windows inset into panels
Figure 9.11: Elevations: solar radiation analysis of original massing

Figure 9.12: Elevations: programmatic break down of original massing

Figure 9.13: Elevations: porosity levels of new massing

Figure 9.14: Elevations: final skin outcome
Inner Skin

Because Kroger’s goals established at the beginning of the design process included reducing energy consumption, access to fresh air, and the incorporation of both passive and active systems, this diagrid skin is simply the outer layer of a double-skin facade. Underneath the parametric exterior is a conventional curtain wall—separated by a 2’-6” air space—that serves to enhance the environmental performance of the building and allow for natural ventilation. To achieve this, the outer skin features automatic sensors that allow the glass panes to “open-up” for air when the weather is conducive—indicating that the tower is “breathing”. Meanwhile, the conventional curtain wall within features user controlled openings that ultimately admit this fresh air into the offices. This twin-face system is a much more appropriate response for high-rise towers than a single operable skin because the outer layer helps protect from sound, wind, and rain—allowing access to fresh air without the associated noise, turbulence, or precipitation. (student paper)

“The reduction of wind pressure by the addition of the extra pane of glass means that the windows can be opened even in the uppermost floors of a high-rise building. Natural ventilation of offices by fresh air is much more acceptable to the building’s users and it has the additional benefits of reducing investment in air handling systems and also reducing energy consumption.”

Most importantly, this system promotes user engagement. It gives people within the building the ability to open windows and control their own environment and comfort. This is a huge step forward for the future of workspaces. Yet beyond the added benefits of natural ventilation and user engagement, the interstitial air cavity also creates a buffer zone that helps absorb some of the incoming solar radiation. This added layer of insulation helps reduce solar heat gain in the summer and reduce heat loss in the winter—which of course helps reduce the energy consumption of the building year-round.

A double-skin façade also reduces heat losses because the reduced speed of the air flow and the increased temperature of the air in the cavity lowers the rate of heat transfer on the surface of the glass. This has the effect of maintaining higher surface temperatures on the inside of the glass, which in turn means that the space close to the window can be better utilized as a result of increased thermal comfort conditions.”

9 Compagno, Andrea. Intelligent Glass Facades; Material, Practice, Design. 1995, 94.
Unfortunately, simply “opening-up” a building facade and letting air in is not enough to allow it to “breathe.” Proper natural ventilation and airflow only occur if an effective system is in place to draw the air across the occupiable spaces. This is achieved here through the use of a solar chimney which draws cool air into the tower and exhausts hot air out at the top (Figure 9.18). In order to achieve this system of ventilation, the core was bumped out in each direction to create a continuous shaft throughout the tower (Figure 9.20). The stack effect created by this shaft is enhanced through the use of a giant solar collector which lies on top of the chimney. The angle and orientation of this collector, which accounts for the tower’s iconic form, was determined by using Galapagos and Ladybug to find the optimal shape for maximizing the collector’s
annual solar radiation (Figure 9.17). The result is a roof that collects 30% more total radiation than if it were simply flat.

This system allows for openings on all sides of the building for year-round access—not just on the side of the prevailing winds in certain months. However, it also takes up a significant portion of the available square footage of each floor plate (900 sq ft less per floor). This was accounted for by adding additional floors to the tower and changing the office layout to an open plan. Furthermore, extra space was created on every floor through the addition of the large, south-facing atriums.
Sky Gardens

In order to achieve Kroger’s final goals of creating a culture and multi-use breakout spaces, the south side of the building was built-out to create large active community gathering spaces with great views towards downtown Cincinnati. These 3-story “sky gardens” meet a variety of different programmatic needs and help promote a sense of community and connection amongst the occupants of various floors. Filling the interstitial space between the outer and inner skins, these sky gardens feel like controlled outdoor environments—full of vegetation, daylight, and fresh air. Because these environments will always reflect the weather outside, these spaces may require occupants to wear a coat or even sunglasses! On the floors above, conference rooms, breakout spaces, and corridors extend out into the atrium, creating a vibrant social setting for interaction and collaboration. The design promotes chance encounters and encourages occupants on all floors to interact and engage with others circulating throughout.

Figure 9.21: Section showing the vibrant activity of one of the community sky gardens
Figure 9.22: Rendering of light entering the sky garden

Figure 9.23: Rendering of sky garden’s second level
Figure 9.24: View of west facade

Figure 9.25: View of north and west facades from the street
Figure 9.26: North/south section cut of building
Bibliography


